Robust Reinforcement Learning : Theory and Practice

Pierre Clavier

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2024IPPAX102

Thèse de doctorat

Robust Reinforcement Learning: Theory and Practice

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à l'Ecole polytechnique

École doctorale n°574 Ecole Doctorale de Mathématiques Hadamard (EDMH)
Spécialité de doctorat : Mathématiques appliquées

Thèse présentée et soutenue à Paris, le 20 novembre 2024, par

PIERRE CLAVIER

Composition du Jury :

Rémi Munos Directeur de recherche, Inria et Meta, FAIR	Président du jury
Aurélien Garivier Professeur, Ecole Normale Supérieure de Lyon	Rapporteur
Ana Bušić Chargée de recherche (HdR), Inria et ENS Ulm	Rapportrice
Eric Moulines Professeur, Ecole polytechnique	Examinateur
Michal Valko Chargé de recherche (HdR), Inria	Examinateur
Shie Mannor Professeur, Technion et Nvidia Research	Examinateur
Erwan Le Pennec Professeur, Ecole polytechnique	Directeur de thèse
Stéphanie Allassonnière Professeure, Université Paris Cité	Co-directrice de thèse
Matthieu Geist Professeur, Université de Lorraine et Cohere	Invité

[†]CMAP, CNRS, Ecole Polytechnique, Institut Polytechnique de Paris, Palaiseau, France.
[‡]Inria Paris, HeKA

Robust Reinforcement Learning: Theory and Practice

Pierre Clavier [†,‡]

Paris, France,
November 21, 2024

This work was supported by the Paris Ile-de-France Region via the DIM Math Innov program.
We also acknowledge support from Fondation Mathématiques Jaques Hadamard.

À mes grands-parents, Odette et Jean.

Contents

Remerciements

Dans un premier temps, je tenais à remercier mes directeurs de thèse Erwan Le Pennec et Stéphanie Allassonnière avec qui j'ai eu la chance de pouvoir travailler et échanger pendant les dernières années. Merci beaucoup à Stéphanie de m'avoir introduit au monde de la recherche, et de m'avoir donné les bases de statistiques computationnelles quand tu étais ma professeure au MVA. Merci pour ta bonne humeur et ton enthousiasme qui permet de persévérer dans la thèse et de se motiver quand on l'est moins. Erwan, cela était un vrai plaisir d'avoir pu collaborer avec toi pendant plus de trois ans, de pouvoir apprendre à tes cotés que cela soit en math ou en machine learning, ainsi que toutes les discussions intéressantes que j'ai pu avoir avec toi en général.

Je tenais également à remercier Matthieu, pout tout le temps que tu as pris pour m'enseigner le RL ! Merci pour tes conseils avisés, ta bonne humeur, tes conseils LaTeX que "j'essaye" de suivre autant que faire se peut, ton recul en recherche et ta capacité à comprendre et répondre à mes questions pour transformer la "mathémagie" en problème intéressant et en preuve rigoureuse. Merci particulièrement pour tout ce que tu m'as appris, pour que je devienne un jeune chercheur, en m'envoyant des papiers, en m'incluant dans la communauté du RL ou en me soutenant dans les moments plus difficiles. Si j'ai autant apprécié ma thèse c'est aussi grâce à toi !

Je remercie également les membres du jury, et en premier lieu les rapporteurs de la thèse Ana et Aurélien, pour m'avoir consacré de leur temps que je sais précieux et pour leurs remarques pertinentes. Merci également aux autres examinateurs Eric, Michal et Shie et Rémi. Thank you to the members of the thesis committee, and foremost to the referees, for their precious time and relevant remarks.

I would like to thank everyone at Caltech with whom I have had the opportunity to talk and collaborate. Many thanks to Eric and Adam for welcoming me into their great and friendly group and for introducing me to another vision of research. I would especially like to thank Laixi, with whom I've had the chance to work and collaborate this year, who taught me a lot about research and take the time to explain concepts to me in a very pedagogical way ! Then, I would also like to thank Jolene for all her work and without whom my visit would have been impossible. Finally, I thank all the people I had the chance to talk to in California, who gave me such a warm welcome, especially Théo, Solène, Elvira and Eric!

Un grand merci à toute le groupe d'Emmanuel à Supaero et tout particulièrement à Adil pour tous les projets que j'ai eu la chance de mener avec toi, toujours dans la bonne humeur même après 2 deadlines ! Merci Emmanuel pour toutes les discussions super intéressantes, scientifiques ou pas, et pour le dynamisme que tu apportes à la communauté RL !

Par la suite je tiens à remercier toutes les personnes du groupe de Stéphanie à savoir Vianney, Solange, Grand Clément, Petit Clément, Fleur, Louis, Agathe, Théo ! Merci à tous pour toutes les supers moments passés en conférence ou séminaire à Bruxelles, Bordeaux, avec vous et tous ce que vous m'avez appris. Un grand merci à Petit Clément pour tous tes conseils en début de

thèse et les conseils en info, pour me motiver à faire des applications comme faire courir des robots/saucisses sur Mujoco ! Un petit merci à Grand Clément qui a presque réussi à me faire croire que SAEM-MCMC pouvait marcher en très grande dimension, mais un grand merci pour m'avoir fait découvrir la fonction torch.einsum qui m'émerveille toujours autant.

Je tenais à remercier tous les du CMAP et pas uniquement les gens "SIMPA" ! Un grand merci à tous les permanents de SIMPA avec qui j'ai eu l'occasion de discuter et particulièrement Alain, Aymeric et Eric pour les nombreuses discussions scientifiques et les collaborations avec Alain qui m'a beaucoup appris en math ! Merci également à Rémi, Emmanuel, Marylou, Mahdi. Merci à tous les doctorants avec qui j'ai pu échanger pendant les retraites au ski ou à la mer, Louis, Aymeric, Daniil (pout toutes les discussions super intéressantes en RL), Antoine, Antoine, Jean, Michael, Orso, Clément, Margaux, Renaud, Guillaume, Alexandre. Enfin, merci Nasséra pour tout son travail et son implication dans le labo. Sans toi le CMAP ne tournerai pas et je n'aurai pas eu la chance de partir en conférence scientifique ces dernières années. Merci également à toute l'équipe gestion du CMAP en général. Merci à toute la team Lagrange : Louis, Lisa, Vincent qui me motive à faire des calculs, Valentin, Pablo, Maxence et son beurre de cacahouète au petit déjeuné en séminaire, Badr, Yazid, Achille pour tes précieux conseils en début de thèse, Mehdi, Mehdi, Thomas, Antonio et le boss de la guitare Gabriel ! Comment oublier Tom, avoir pu collaborer avec toi a toujours été super que ça aux nuits passé à Lagrange à faire des maths et à concevoir des algorithmes pour améliorer la VITS de ThomPson Sampling.

Je souhaite également remercier tous les gens de mon équipe INRIA, HeKA et particulièrement Sarah, Adrien, Moreno, et Jean Feydy pour toutes les discussions super intéressantes avec toi, Alice, Juliette, Jean-Baptiste ainsi que tous les ingénieurs de recherche. (Un merci très sincère à Antoine et Marine pour la cantine toutes ses années.) Je tiens à remercier évidement Linus pour avoir été un super compagnon de thèse toutes ses années et pour le dynamisme et l'énergie que tu mets ans la communauté !

Un grand merci à toute l'équipe de Cohere et spécialement l'équipe RL qui m'a beaucoup appris pendant mon stage à savoir Omar, Yannis, Nathan et Florian.

Merci à toute l'équipe du MAP5 qui m'ont donné envie de faire de recherche quand j'étais en stage de master en particulier mon bureau composé de Pierre, Alexandre, Claire, Vincent, Pierre Louis, Anton et enfin Remi le maitre ultime de python qui m'a tout appris ainsi les gens du LPSM, Adeline, Alexandra et Anna. Je tenais également à remercier Warith pour toutes tes discussions scientifiques et de m'avoir donné envie de faire de la recherche, enfin à mes anciens encadrant de stage Oliver, Grégory qui m'ont introduit à la recherche.

Merci à mes professeurs de sciences qui m'ont donné l'envie de faire des sciences, en particulier M. Briche à Fénélon et M. Aubert à Poinca qui m'ont donné le gôut des maths.

Merci à Julia d'avoir été présente tout au long de ma thèse et en master, d'avoir partagé la quasi-totalité des projets avec moi au MVA. Ton dynamisme et ta rigueur scientifique me pousse à donner le meilleur de moi et j'apprécie tous les moments passé avec toi scientifiques ou non ! Enfin merci à Jos pour ta bonne humeur et ton courage de commencer une thèse de RL que j'en suis certain sera géniale ! Enfin merci infiniment à Raphael, ou plutôt Shakespeare d'avoir relu mon introduction avec attention.

Je remercie celles et ceux qui ont croisé ma route à de nombreuses reprises et avec qui les discussions furent toujours enrichissantes.

Je tiens à remercier mes ami · e · s, en particulier celles et ceux que je n'ai pas déjà mentionné · e · s ou que je ne mentionnerai pas ci-dessus/dessous !

Dans mes amis qui ne font pas/plus des maths (ou pas encore !) je tenais à remercier :

Mes amis d'enfance Théophile, Billy, Hubert, Constantin, Jean. Remerciement spécial à Edmond pour sa contribution à cette thèse à avoir le super template, les conseils en info, le télétravail le mardi et nos repas équilibrés achetés chez Rachid.

Jag ville tacka mina vänner jag fick i Sverige, Carla, Clara, Grison, Claire, Christelle, Marie, Sylvain, speciellt de två bästa rumskamraterna, Nerf-bossen och bowling-bossen (jag låter dig bestämma vem som är vem) Paulo och Mimi. Massor av pussar och kannellbular!

Mes (anciens) colloc, Manon, Laura, Corentin, Weisrock, Louis avec qui je partage ou j'ai partagé des supers moments.

Mes amis de toute part et personnes que j'apprécie beaucoup depuis le collège jusqu'à maintenant: Ezechiel, David, Mathieu, Ugo, Clément M., Sophia, Elodie, Raphael, Philippine, Bélen, Florian, Clément B, François, Marina, Yoann, Antoine, Thomas, Louis M., Gaspard, William, Antonin, Marie, Hugo, Arthur, Serge, Mehdi, Aurian, Mathieu, Morgane, Fred, Martin, Hortence, Niel, Paul, Louis D, Nicolas, Erwan, Edouard, Ariala, Lucie, Audrey, Manon, Eva, Alexandre, Laure, Baptiste, Youenne, Marwan, Pierre, Anaid, Natacha, Maud et Anastasia.

Merci à toute ma famille qui m'a soutenue pendant toutes ses années, je pense notamment à mes grands-parents Gilbert, Jean, Odette, Colette.

Enfin un grand merci à mes parents Evelyne et Jean-Yves, ma sœur Lucile qui m'ont toujours encouragé à faire ce que je voulais, ont éveillé ma curiosité et m'ont soutenu toutes ces années à chaque instant.

A Paris, November 21, 2024
Pierre Clavier

Résumé court

L'apprentissage par renforcement (RL) est un paradigme d'apprentissage automatique qui aborde la question de la prise de décision séquentielle. Dans ce paradigme, l'algorithme, désigné comme un agent, réagit à des interactions avec un environnement. À chaque interaction, l'agent effectue une action dans l'environnement, observe un nouvel état de l'environnement et reçoit une récompense en conséquence. L'objectif de l'agent est d'optimiser une récompense cumulative, qui est définie par l'utilisateur pour s'aligner sur la tâche spécifique à accomplir dans l'environnement. La théorie du processus décisionnel de Markov (MDP) est utilisée pour formaliser ce concept. Cependant, en cas de mauvaise spécification du modèle ou d'erreur dans la fonction de transition de l'environnent ou de la récompense, les performances du RL peuvent diminuer rapidement. Pour résoudre ce problème, le concept de MDP robustes a émergé, l'objectif étant d'identifier la politique optimale sous l'hypothèse que le noyau de transition appartient à un ensemble d'incertitude. Cette thèse présente une étude théorique de la complexité d'échantillonnage des MDP robustes, ou de la quantité de données nécessaires pour atteindre une erreur arbitrairement petite. Ces résultats démontrent que dans certains cas, cette complexité peut être inférieure à celle des MDP classiques, ce qui constitue une voie prometteuse pour concevoir de nouveaux algorithmes efficaces sur le plan de l'échantillonnage. La thèse se poursuit par des propositions de nouveaux algorithmes RL robustes pour renforcer les performances de RL ayant des ensembles d'action continus. Notre méthode est basée sur les MDP averses aux risques et les jeux à somme nulle, dans lesquels l'adversaire peut être considéré comme un agent qui change l'environnement dans le temps. En conclusion, la dernière section présentera des nouvelles tâches pour l'évaluation des algorithmes RL robustes, qui manquent de références pour l'évaluation des performances.

Mots clés : processus décisionnel de Markov, appentissage par renforcement robuste, robustesse

Abstract

Reinforcement learning (RL) is a machine learning paradigm that addresses the issue of sequential decision-making. In this paradigm, the algorithm, designated as an agent, responds to interactions with an environment. At each interaction, the agent performs an action within the environment, observes a new state of the environment, and receives a reward in consequence. The objective of the agent is to optimise an cumulative reward, which is defined by the user to align with the specific task at hand within the environment. The Markov Decision Process (MDP) theory is used in order to formalise these concepts. However, in the event of mispecifications or errors in the transition or reward function, the performance of RL may decline rapidly. To address this issue, the concept of robust MDPs has emerged, whereby the objective is to identify the optimal policy under the assumption that the transition kernel belongs to a bounded uncertainty set. This thesis presents a theoretical study of the sample complexity of robust MDPs, or the amount of data required to achieve an arbitrary small convergence error. It demonstrates that in certain cases, the sample complexity of robust MDPs can be lower than for classical MDPs, which is a promising avenue for the derivation of sample-efficient algorithms. The thesis then goes on to derive new robust RL algorithms to strengthen the performance of RL in continuous control. Our method is based on risk-averse MDPs and zero-sum games, in which the adversary can be seen as an agent that changes the environment in the time. In conclusion, the final section present a benchmark for the evaluation of robust RL algorithms, which currently lack a reproducible benchmark for performance assessment.

Keywords : Robust Markov Decision Process, Robust Reinforcement Learning, Sample Complexity

Notations

Mathematical Notations

- $\mathbb{N}$ set of intergers
- $\mathbb{R}$ set or real numbers
- $M^\top$ transpose of a matrix M
- $\mathcal{N}$ normal distribution
- $\mathbb{E}$ expectation under a probabilistic model
- $\mathbb{V}$ variance
- $\Delta(\mathcal{S})$ the space of probability distributions over $\mathcal{S}$ (i.e., the probability simplex)
- 2^S set of subsets of a set E
- $\|.\|$ an arbitrary norm and $\|.\|_p$ the classical L_p norm.
- θ parameter to learn in a statistical model
- $\arg\max$ set of all maximizers
- $\mathcal{U}$ uniform distribution
- $\mathbf{1}$ for unitary vector and $\mathbf{1}_s$ for unitary of dimension S

Markov Decisions Processes Notations

- $\mathcal{M}$ a MDP
- $\mathcal{S}$ state space of context space of dimention $S \leq \infty$ in Chapter 2 and 3
- $\mathcal{A}$ action space of dimension A with $A \leq \infty$ in Chapter 2 and 3
- γ the discount factor, $\gamma \in [0, 1)$
- r reward function reward function of the agent $r : s, a \to r(s, a)$
- P transition kernel $s' \sim P(s'|s, a)$
- τ the trajectory or rollout following kernel P and policy π.
- $\mathbb{P}$ the probability distribution over the trajectories or rollout $\tau \sim \mathbb{P} = (\pi, P)$
- R the return of one trajectory $R(\tau) = \sum_{t \geq 0} \gamma^t r_t$

- ρ initial state distribution
- σ radius of the uncertainty set in Robust MDPs
- P^0 nominal kernel in Robust MDPs
- $\mathcal{T}$ the Bellman Operator
- V state value function where $*$ stands for optimal value and π for policy value
- Q state-action value function, where $*$ stands for optimal value, π for policy value
- H the horizon factor in infinite discounted MDP equal to $H = 1/(1-\gamma)$
- π the policy learn and Π the set of all policies from $\mathcal{S}$ to $\mathcal{A}$
- $\mathcal{D}$ a dataset
- $\mathcal{B}$ a batch of the dataset $\mathcal{D}$
- t the index of the time in the MDP
- k iteration index of an algorithm, usually used as a subscript

Chapter 1

Introduction

Contents

Essayer d'imiter un esprit humain adulte nous oblige à beaucoup réfléchir au processus qui l'a conduit à cet état. Nous pouvons en relever trois composantes.

- *(a) l'état initial de l'esprit, à la naissance ;*
- *(b) l'éducation à laquelle il a été soumis ;*
- *(c) un autre type d'expérience, que nous ne rangeons pas sous le terme "éducation" à laquelle il a été confronté.*

Au lieu d'essayer de produire un programme qui simule l'esprit adulte, pourquoi ne pas plutôt essayer d'en produire un qui simule celui de l'enfant ? S'il était soumis à une éducation appropriée , on aboutirait au cerveau adulte. Il est probable que le cerveau de l'enfant est une sorte de calepin comme on peut en trouver dans les papeteries : un mécanisme plutôt petit et avec beaucoup de feuilles blanches. ("Mécanisme" et "écriture" sont pour nous pratiquement synonymes.) Notre espoir est qu'il y ait un si petit mécanisme dans le cerveau de l'enfant qu'il soit aisément programmable. En première approximation, nous pouvons supposer que la quantité de travail nécessaire à cette éducation serait pratiquement identique à celle qui est destinée à un enfant humain.

Alan Turing, Machine à calculer et intelligence (1950) (traduit par Gromov)

1.1 Résumé et introduction en français

On évalue souvent la pertinence d'une décision après une certaine période de temps. Dans les jeux ou dans la vie en général, les décisions peuvent avoir des impacts qui s'étendent bien au-delà du moment initial du choix, et agir en prenant en compte les implications futures est un aspect primordial de l'intelligence. Bien que les récents progrès en apprentissage automatique aient démontré des capacités impressionnantes dans les prédictions à une étape ou de manière non séquentielle, telles que la transcription de la parole en texte, la prédiction de la forme des protéines ou la reconnaissance du contenu des images, la création d'algorithmes capables de modifier leurs actions pour tenir compte des résultats futurs reste l'un des défis les plus significatifs de la recherche contemporaine en intelligence artificielle. La capacité de planifier et de prédire une séquence d'actions pour résoudre ce problème est généralement désignée sous le terme de *prise de décision séquentielle.*

Dans la nature, les humains et les animaux sont capables de prendre des décisions séquentielles. Par exemple, les neurotransmetteurs tels que la dopamine, qui est synthétisée dans le cerveau et les reins des humains et animaux, sont impliqués dans la modulation des comportements motivés par une récompense (Berridge 2007). Notamment, la libération de dopamine en anticipation d'un stimulus gratifiant ou en réponse à une récompense qui dépasse les attentes (Montague et al. 1996) montre la capacité des mécanismes neurochimiques à adapter leur comportement en réponse aux stimuli environnementaux et à optimiser leurs actions pour les résultats souhaités.

D'un point de vue plus informatique ou mathématiques, l'un des pionniers de la prise de décision séquentielle est Bellman. Dans son célèbre ouvrage intitulé "Dynamic Programming", Bellman (1966) a été l'un des premiers à établir les fondements de l'apprentissage par renforcement. Bien que le travail de Bellman soit principalement théorique et méthodologique, la compréhension ultérieure des phénomènes biologiques a fortement influencé ses travaux.

Enfin, le terme ”apprentissage par renforcement”, tel que défini formellement par Sutton and Barto (2018), est un paradigme mathématique qui permet aux agents d'interagir avec leur environnement et d'apprendre des comportements qui maximisent leur récompense cumulative au fil du temps. Au cours de ce processus, les agents apprennent à éviter les mauvaises actions qui peuvent avoir des conséquences négatives à l'avenir et à agir de manière à améliorer leur résultat final dans un environnement donné.

Un type particulier de prise de décision séquentielle est appelé problèmes des bandits. Les algorithmes de bandits représentent une classe d'approches conçues principalement pour résoudre le problème du bandit manchot (Auer et al. 2002, Lattimore and Szepesvári 2020). Dans sa forme classique, le problème du bandit manchot consiste à sélectionner une stratégie pour maximiser le profit, étant donné n machines à bandit manchot à un seul bras avec des gains qui suivent des distributions de probabilité distinctes et inconnues. Une caractéristique unique de ce problème est que les décisions passées n'ont pas d'impact sur les résultats des décisions ultérieures, de la même manière qu'un nouveau tirage de roulette dans un casino est indépendant des tirages précédents.

L'apprentissage par renforcement (RL), contrairement au problème des bandits, est un problème d'apprenti-ssage séquentiel où l'influence des décisions passées pèse sur les décisions futures. Le RL a démontré des performances impressionnantes dans une grande variété de domaines, notamment les jeux (Silver et al. 2017), l'alignement de grands modèles linguistiques (Ziegler et al. 2019, Achiam et al. 2023), la robotique et le contrôle (Kober et al. 2013) ou encore les soins de santé (Liu et al. 2019, Fatemi et al. 2021). Ces remarquables accomplissements peuvent être attribués à la quantité importante de données utilisées dans le processus d'apprentissage de la politique ou statégie de choix des actions.

Cependant, dans certaines situations, les données disponibles peuvent ne pas être suffisantes pour apprendre une politique efficace, ce qui entraîne des politiques qui généralisent mal et qui mènent des performances sous-optimales lorsqu'elles sont déployées dans des applications réelles. Les approches basées sur les données deviennent de plus en plus cruciales pour améliorer divers aspects de la vie humaine. Par conséquent, lors du développement d'algorithmes d'apprentissage par renforcement, quels facteurs devraient être pris en compte ?

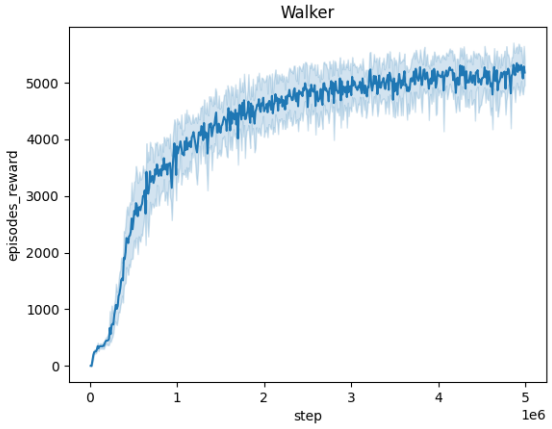
- **La notion de robustesse**. Dans le contexte de l'apprentissage par renforcement, la robustesse face à des perturbations est une caractéristique primordiale. En RL, les performances d'une politique apprise dans l'environnement d'entraînement peuvent se dégrader considérablement une fois déployée en phase de test dans un environnement en en raison de l'incertitude et de la variabilité, qui peuvent être causées par des perturbations aléatoires et des événements rares ou même des attaques malveillantes (Mahmood et al. 2018). Par conséquent, il est crucial de développer des algorithmes RL capables de gérer efficacement de telles incertitudes et de garantir que les politiques apprises peuvent se généraliser de manière adéquate à de nouveaux environnements.

- **L'efficacité d'échantillonage** est également un aspect crucial de l'apprentissage par renforcement (RL) moderne. La complexité des problèmes RL contemporains a augmenté de manière significative, avec des environnements plus grands et des modèles pour apprendre politique plus complexes (Silver et al. 2017, Achiam et al. 2023). Par conséquent, les algorithmes de RL ont souvent besoin de vastes quantités de données pour apprendre des politiques efficaces. Ce défi est d'autant plus difficile de par la nature séquentielle des problèmes RL, où la complexité de l'environnement croît de manière exponentielle avec la longueur de l'horizon. Par conséquent, l'amélioration de l'efficacité d'échantillonage est une direction de recherche essentielle pour permettre aux agents RL d'apprendre des politiques efficaces avec des données et des ressources de calcul limitées. Du point de vue théorique,

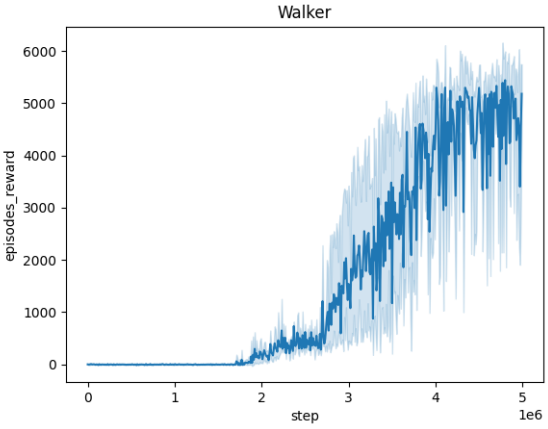
les recherches récentes se sont concentrées sur le développement d'un cadre théorique à échantillon fini (Kakade 2003) pour évaluer et comparer l'efficacité d'échantillonage des algorithmes d'apprentissage par renforcement dans des contextes de grande dimension. Cependant, la compréhension statistique du RL actuelle reste incomplète, en particulier en raison des difficultés techniques rencontrées d'un point de vue théorique. Par conséquent, des recherches supplémentaires sont nécessaires pour améliorer l'efficacité d'échantillonage des algorithmes RL dans des contextes de grande dimension.

- **La reproductibilité des performances** des algorithmes d'apprentissage par renforcement et la façon de s'adapter à des espaces de grande dimension sont également d'une importance capitale. Dans les applications pratiques, la dimensionnalité des environnements rencontrés est souvent élevée, ce qui rend la mise à l'échelle des algorithmes RL une considération critique, en particulier dans les situations où les ressources en mémoire et calcul sont limitées. De plus, le RL est fréquemment critiqué pour son manque de robustesse et ses performances difficiles à reproduire. Par conséquent, il est essentiel de développer des algorithmes RL qui peuvent performer efficacement dans des environnements de haute dimension et concevoir des benchmarks pour tester la robustesse des algorithmes et obtenir des performances reproductibles.

Le problème de complexité ou **efficacité d'échantionnage** est représenté dans la Figure 1.1b. Dans cette figure l'algorithme demande beaucoup d'échantillon à l'environnement pour converger ou pour obtenir des bonnes performances alors que dans Figure 1.1a, la convergence est plus rapide pour le même environnement Walker-v3.



(a) Un algorithme avec une bonne complexité d'échantionnage



(b) Un algorithme avec moins bonne complexité d'échantionnage

Le problème de robustesse face à des perturbations est présenté dans la figure 1.2. Dans cette figure, on enraîne un agent qui interagit dans un environnement nommé CartPole, dont le but est de contrôler une barre fixée à un chariot par une articulation non actionnée. Cet environnement possède initialement une grandeur physique (la longueur de la barre) relative de 1, mais en phase de test, on évalue l'agent en modifiant la cette grandeur physique. Le but de cette thèse est de concevoir des algorithmes robuste à ces perturbations. Le paramètre α dans le graphe contrôle la robustesse induite dans l'algorithme de RL. Plus de détails sont disponible au chapitre 4 . La question centrale à laquelle tenterons de répondre est:

Pouvons nous concevoir des algorithmes de RL qui aient à la fois une bonne complexité d'échantillonnage, soient robustes, passent à l'échelle en terme de dimension tout en ayant des performances reproductibles ?

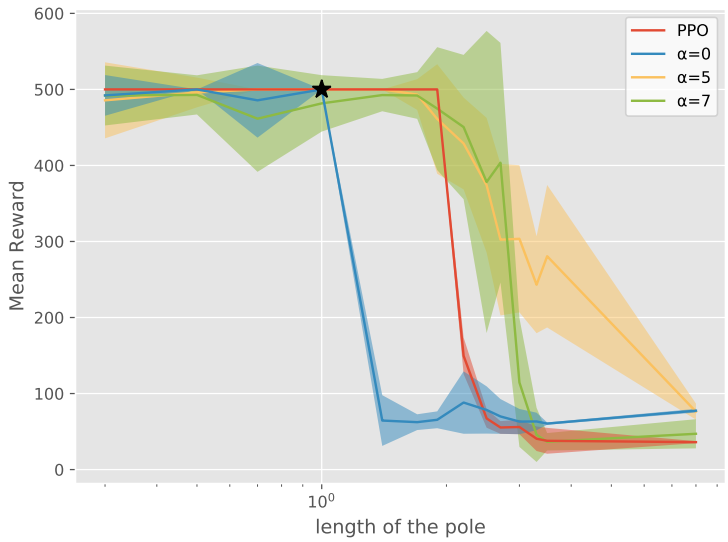


Figure 1.2: Performance d'un algorithme entrainé avec une masse relative de 1 sur l'environment CartPole-v1, variant la masse du CartPole en phase de test

Nous tenterons de répondre à cette question en se focalisant sur le RL robuste en montrant que ce dernier est un paradigme qui peut répondre à ces différents critères.

Comme mentionné précédemment, l'apprentissage par renforcement (RL) a connu des réussites significatives ces dernières années ; cependant, il fait souvent face à des défis en termes de robustesse et de généralisation. Ces défis surviennent principalement parce que les agents sont trop ajustés à l'environnement d'entraînement spécifique, ce qui peut entraîner de piètres performances lors du déploiement. Les agents RL sont généralement entraînés en simulation en raison du coût élevé de l'interaction avec les systèmes physiques. Toutefois, les simulations peuvent contenir des erreurs de modélisation et des paramètres imprécis, ce qui entraîne une divergence entre la simulation et la réalité où la politique entraînée peut avoir du mal à gérer pendant la transition de la simulation au réel. Même les politiques entraînées directement sur le système réel peuvent résister à des incertitudes ou des perturbations préalablement non rencontrées, de légères déviations dans les paramètres de l'environnement, tels que la masse ou la friction, peuvent avoir un impact significatif sur les performances d'une politique, ce qui peut faire la différence entre la réussite et l'échec dans les scénarios de test (Morimoto and Doya 2005, Pinto et al. 2017).

Pour résoudre ce problème, les processus de décision Markov robustes (RMDP) ont été introduits dans Iyengar (2005), Nilim and El Ghaoui (2005). Ce cadre est naturel et polyvalent car il exploite les informations issues de l'optimisation robuste distributionnelle et de l'apprentissage supervisé (Bertsimas et al. 2018, Blanchet and Murthy 2019, Duchi and Namkoong 2021). Contrairement aux MDP conventionnels, les RMDPs fournissent un cadre de modélisation plus étendu, permettant la spécification de la forme et de la magnitude de l'ensemble d'incertitude. Fréquemment, l'ensemble d'incertitude est choisi comme étant une petite boule centrée autour du noyau nominal ayant un rayon σ, ayant une forme définie par une métrique qui mesure la distance entre les distributions de probabilité. Pour faciliter la faisabilité de la résolution des RMDP, l'ensemble d'incertitude est généralement supposé posséder certaines propriétés structurelles. Par exemple, des travaux antérieurs (Iyengar 2005, Wiesemann et al. 2013) ont proposé que l'ensemble d'incertitude puisse être décomposé en sous-ensembles indépendants pour chaque état ou paire état-action, appelés respectivement s- et (s, a)-rectangularité. Dans cette thèse, nous adopterons l'hypothèse de (s, a)-rectangularité pour l'ensemble d'incertitude. D'un point

de vue théorique, cette hypothèse sera utile dans les chapitres 2 et 3 tandis que nous essaierons de l'éviter d'un point de vue pratique car il s'agit d'une hypothèse restrictive dans le chapitre 6.

Les contributions de cette thèse sont les suivantes. Après un bref aperçu résumant les notions utiles dans le chapitre 1, le manuscrit est organisé en deux parties : la première se concentre sur la théorie des MDP robustes et en particulier sur la question de la complexité de l'échantillon, la seconde étudie l'apprentissage par renforcement robuste d'un point de vue pratique. Ces deux directions ne sont pas orthogonales : au contraire, l'étude théorique des MDP peut donner des idées sur la façon de concevoir des algorithmes d'apprentissage par renforcement robustes efficaces et l'utilisation d'algorithmes d'apprentissage par renforcement robustes donne une intuition sur la complexité de l'échantillon des MDP robustes.

Tout d'abord, nous aborderons la question de la *complexité d'échantionnage.*. Supposons que l'on ait accès à des échantillons de données générés par un MDP avec un noyau de transition nominal, obtenus par certains mécanismes d'échantillonnage. L'objectif principal de l'apprentissage par renforcement traditionnel est d'apprendre la politique optimale spécifiquement adaptée au noyau nominal, pour lequel la limite de complexité d'échantillon minimax a été bien établie (Azar et al. 2013a). En revanche, l'objectif de l'apprentissage par renforcement robuste distributionnel est d'apprendre une politique plus résiliente en utilisant le même ensemble d'échantillons de données, en optimisant les performances dans le pire des cas lorsque le noyau de transition est choisi arbitrairement à partir d'un ensemble d'incertitude prédéfini autour du noyau nominal. La complexité de l'échantillon pour les RMDP a été étudiée dans (Yang et al. 2022, Panaganti and Kalathil 2022a, Shi et al. 2024). D'un point de vue de la complexité de l'échantillon, nous démontrerons que les RMDP ne sont pas plus difficiles à apprendre que les MDP classiques pour un petit rayon d'incertitude σ, et peuvent même être plus simples à apprendre lorsque le rayon est plus grand. Cette constatation fournit une motivation pour l'utilisation de RMDP afin de développer un algorithme efficace en termes d'échantillons dans le chapitre 2 et 3.

Première contribution sur la complexité d'échantionnage des RMDPs

Dans la première partie, nous nous concentrons sur la compréhension de la complexité de l'échantillon des MDPs robustes. Plus précisément, dans le chapitre 2, nous étudions la complexité de l'échantillon pour obtenir une politique ϵ-optimale dans les processus de décision Markoviens robustes à horizon infini escompté ou actualisé (RMDPs), en n'ayant accès qu'à un modèle génératif du noyau nominal. Ce problème est largement étudié dans le cas non robuste, et il est connu qu'une approche de planification appliquée à un MDP empirique estimé avec $\tilde{\mathcal{O}}(\frac{H^3SA}{\epsilon^2})$ échantillons fournit une politique ϵ-optimale, ce qui est optimal au sens minimax. Les résultats dans le cas robuste sont beaucoup plus rares. Pour les ensembles d'incertitude *sa-* (resp *s-*) rectangulaires, jusqu'à récemment, la meilleure complexité d'échantillon connue était $\tilde{\mathcal{O}}(\frac{H^4S^2A}{\epsilon^2})$ (resp. $\tilde{\mathcal{O}}(\frac{H^4S^2A^2}{\epsilon^2})$), pour des algorithmes spécifiques et lorsque l'ensemble d'incertitude est basé sur la divergence de la variation totale (TV), la divergence KL ou la divergence du Chi-square. Dans cet article, nous considérons des ensembles d'incertitude définis avec une L_p-boule (retrouvant le cas TV), et nous étudions la complexité de l'échantillonage de n'importe quel algorithme de planification (avec une garantie de haute précision sur la solution) appliqué à un RMDP empirique estimé à l'aide du modèle génératif. Dans le cas général, nous dérivons une complexité d'échantillon de $\tilde{\mathcal{O}}(\frac{H^4SA}{\epsilon^2})$ pour les cas *sa-* et *s-*rectangulaires (améliorations de S et SA respectivement). Lorsque la taille de l'incertitude est suffisamment petite, nous améliorons la complexité de l'échantillon à $\tilde{\mathcal{O}}(\frac{H^3SA}{\epsilon^2})$, retrouvant la borne inférieure pour le cas non robuste pour la première fois et une borne inférieure robuste. Enfin, nous introduisons également des algorithmes simples et efficaces pour résoudre les MDPs robustes L_p étudiés.

Deuxième contributions sur la complexité d'échantionnage des RMDPs

Dans le chapitre 3, nous affinons le résultat du chapitre 2 en supposant l'accès à un modèle génératif qui échantillonne à partir du MDP nominal. Nous examinons la complexité de l'échantillon des RMDPs en utilisant une classe de normes L_p généralisées comme fonction de "distance" pour l'ensemble d'incertitude, sous deux conditions *sa*-rectangulaires et *s*-rectangulaires couramment adoptées. Nos résultats impliquent que les RMDPs peuvent être plus efficaces en termes d'échantillons à résoudre que les MDPs standard en utilisant des normes L_p généralisées dans les cas *sa-* et *s-*rectangulaires, ce qui pourrait inspirer davantage de recherches empiriques. Nous fournissons une borne supérieure quasi optimale et une borne inférieure minimax correspondante pour les scénarios *sa*-rectangulaires. Pour les cas *s*-rectangulaires, nous améliorons la borne supérieure de l'état de l'art et dérivons également une borne inférieure en utilisant la norme L_∞ qui vérifie l'exactitude. Par rapport au chapitre 2, nous améliorons la complexité de l'échantillon, montrant qu'il est possible d'obtenir une complexité d'échantillon inférieure à celle des MDPs classiques. Cette partie ouvre une voie prometteuse pour dériver des algorithmes qui peuvent atteindre une complexité d'échantillon plus faible tout en étant plus robustes aux perturbations.

Dans la deuxième partie de cette thèse, nous nous concentrons sur la dérivation d'algorithmes d'apprentissage par renforcement robustes (Robuste RL) à partir d'un point de vue pratique. Nous montrons que les idées issues des MDPs robustes peuvent être utilisées pour concevoir des algorithmes de Robuste RL en utilisant une formulation basée sur MDPs risque-averses. Plus précisément, l'idée de cette classe d'algorithmes est d'approcher l'opérateur minimum interne présent dans l'opérateur de Bellman robuste (1.37). Les travaux précédents ont généralement employé une approche duale pour le problème minimum, où la probabilité de transition est

contrainte à rester dans une boule spécifiée autour du noyau de transition nominal. Dans des travaux précédents, (Kumar et al. 2022) a dérivé un algorithme approché pour les RMPDS avec des boules L_p, (Liu et al. 2022) a utilisé une boule définie avec une divergence KL et nous essayons d'approcher pour les RMDPs avec une divergence χ^2 dans le chapitre 4. En pratique, la robustesse est équivalente à la régularisation (Derman et al. 2021) et, par exemple, l'algorithme SAC (Haarnoja et al. 2018a) possède des charactéristiques robuste en raison de la régularisation entropique (Eysenbach and Levine 2021). Enfin, Wang et al. (2023) propose une nouvelle approche en ligne pour résoudre les RMDP. Contrairement aux travaux précédents qui régularisent avec la politique ou la fonction de valeur, Wang et al. (2023) crée de la robustesse en simulant les pires scénarios de noyau pour l'agent tout en utilisant n'importe quel algorithme d'apprentissage par renforcement classique dans le processus d'apprentissage.

L'idée que la régularisation et la robustesse sont étroitement liées sera également centrale dans cette thèse dans les chapitres 4 et 5. L'idée centrale dans le chapitre 5 est que nous évitons l'estimation d'une pénalisation ou d'une régularisation, et estimons plutôt l'expectile de la fonction de valeur, ce qui crée une robustesse implicite. Ces types d'algorithmes sont mathématiquement bien fondés, mais utilisent uniquement que des échantillons provenant du noyau de transition nominal. L'idée d'utiliser des échantillons pas seulement du noyau nominal est présente dans le concept de randomisation de domaine (DR) Tobin et al. (2017) qui apprend une fonction de valeur qui maximise le rendement attendu en moyenne sur une distribution fixe (généralement uniforme) sur l'ensemble d'incertitude. Cette méthode utilise des échantillons de toute l'incertitude et sera combinée avec une formulation d'aversion au risque dans le chapitre 5. Pour relever les défis mentionnés précédemment d'utiliser uniquement des échantillons à partir du nominal et d'éviter les hypothèses de rectangularité, une approche utilisant le concept de jeux à deux joueurs à somme nulle ou min-max est proposée dans le chapitre 6. Notre algorithme est basé, comme de nombreux algorithmes d'apprentissage en profondeur robustes existants tels que M2TD3 Tanabe et al. (2022a), M3DDPG (Li et al. 2019a), ou RARL (Pinto et al. 2017), sur le jeu à deux joueurs à somme nulle présenté dans la section 6.2.

Première contribution algorithmique en RL robuste

L'apprentissage par renforcement robuste essaie de rendre les prédictions plus robustes aux changements dans la dynamique ou les récompenses du système. Ce problème est particulièrement important lorsque la dynamique et les récompenses de l'environnement sont apprises et estimées à partir des données. Dans le chapitre 4, nous essayons d'approcher l'apprentissage par renforcement robuste contraint avec une divergence de χ^2 en utilisant une formulation de RL averse au risque approchée. Nous montrons que la formulation classique de l'apprentissage par renforcement peut gagner en robustesse en utilisant une pénalisation de l'écart-type de l'objectif. Deux algorithmes basés sur le Reinforcement Learning distributionnel, l'un pour l'espace d'actions discret et l'autre pour l'espace d'actions continu, sont proposés et testés sur des environnements Gym classiques pour démontrer la robustesse des algorithmes.

Seconde contribution algorithmique en RL robuste

Dans le chapitre 5, nous dérivons une nouvelle forme de robustesse implicite en RL en utilisant le bootstrapping d'expectile. L'utilisation de cette technique évite d'estimer une pénalisation comme dans le chapitre 4. De nombreux algorithmes classiques de Reinforcement Learning (RL) reposent sur un opérateur de Bellman, qui implique une espérance sur les états suivants, conduisant au concept de bootstrapping. Pour introduire une forme de pessimisme, nous proposons de remplacer cette espérance par un expectile. En pratique, cela peut être très simplement fait en remplaçant la perte L_2 par une perte d'expectile plus générale pour le critique. L'introduction de pessimisme en RL est souhaitable pour diverses raisons, telles que la résolution du problème de surestimation (pour lequel les solutions classiques sont le double Q-learning ou l'approche twin-critic de TD3) ou le RL robuste (où les transitions sont adverses). Nous étudions empiriquement ces deux cas. Pour le problème de surestimation, nous montrons que l'approche proposée, `ExpectRL`, fournit de meilleurs résultats qu'un twin-critic classique. Sur les benchmarks de RL robuste, impliquant des changements de l'environnement, nous montrons que notre approche est plus robuste que les algorithmes classiques de RL. Nous introduisons également une variante de `ExpectRL` combinée avec la randomisation de domaine qui est compétitive avec les agents de RL robuste de l'état de l'art. Enfin, nous étendons également `ExpectRL` avec un mécanisme pour choisir automatiquement la valeur d'expectile, c'est-à-dire le degré de pessimisme.

Troisième contribution algorithmique en RL robuste

Dans le chapitre 6, nous essayons de dériver un nouvel algorithme sans hypothèses de rectangularité. Les hypothèses de rectangularité en RL L'apprentissage par renforcement robuste traditionnel dépend souvent d'hypothèses de rectangularité, où les mesures de probabilité adverses des états de résultat sont supposées être indépendantes pour différents états et actions. Cette hypothèse, rarement respectée dans la pratique, conduit à des politiques excessivement conservatrices. Pour résoudre ce problème, nous introduisons une nouvelle formulation de MDP robuste à temps contraint (TC-RMDP) qui prend en compte les perturbations multifactorielles, corrélées et dépendantes du temps, reflétant ainsi plus précisément les dynamiques du monde réel. Cette formulation va au-delà du paradigme conventionnel de rectangularité, offrant de nouvelles perspectives et élargissant le cadre analytique pour l'apprentissage par renforcement robuste. Nous proposons trois algorithmes distincts, chacun utilisant différents niveaux d'informations environnementales, et les évaluons de manière approfondie sur des benchmarks de contrôle continu. Nos résultats montrent que ces algorithmes offrent un compromis efficace entre performance et robustesse, surpassant les méthodes traditionnelles d'apprentissage par renforcement robuste en profondeur dans les environnements à temps contraint tout en maintenant la robustesse dans les benchmarks classiques. Cette étude remet en question les hypothèses prédominantes en apprentissage par renforcement robuste et ouvre de nouvelles voies pour le développement d'applications d'apprentissage par renforcement plus pratiques et réalistes.

Enfin, pour obtenir des méthodes reproductibles et évolutives, nous avons créé un benchmark normalisé : RRLS dans le chapitre 7. Nous avons testé notre dernier algorithme TC-MDPs sur ce benchmark pour créer un algorithme reproductible qui peut évoluer avec la dimension et offrir des performances reproductibles.

Contribution en RL Robuste reproductible

Nous introduisons la Robust Reinforcement Learning Suite (RRLS), une suite de benchmarks basée sur des environnements Mujoco. RRLS propose six tâches de contrôle continu avec deux types d'ensembles d'incertitude pour l'entraînement et l'évaluation. Notre benchmark vise à standardiser les tâches d'apprentissage par renforcement robuste, facilitant ainsi des expériences reproductibles et comparables, en particulier celles issues de récentes contributions de pointe, pour lesquelles nous démontrons l'utilisation de RRLS. Il est également conçu pour être facilement extensible à de nouveaux environnements. Le code source est disponible à l'adresse https://github.com/SuReLI/RRLS.

Dans le chapitre 8, nous abordons le problème de la représentation de la distribution postérieure dans le problème de bandit en utilisant des algorithmes d'échantillonnage de Thompson avec une distribution postérieure arbitraire apprise à l'aide de l'inférence variationnelle.

Première contribution sur la théorie des bandits

Nous introduisons et analysons une variante de l'algorithme d'échantillonnage de Thompson (TS) pour les bandits contextuels. À chaque tour, le TS traditionnel nécessite des échantillons de la distribution postérieure actuelle, ce qui est généralement intractable. Pour contourner ce problème, des techniques d'inférence approchée peuvent être utilisées et fournissent des échantillons avec une distribution proche des postérieures. Cependant, les techniques d'approximation actuelles conduisent soit à une estimation de mauvaise qualité (approximation de Laplace), soit peuvent être coûteuses en calcul (méthodes MCMC, échantillonnage d'ensemble, etc.). Dans cet article, nous proposons un nouvel algorithme, l'inférence variationnelle TS (`VITS`), basé sur l'inférence variationnelle gaussienne. Ce schéma fournit des approximations de postérieures puissantes qui sont faciles à échantillonner, et qui sont efficaces sur le plan computationnel, ce qui en fait un choix idéal pour TS. De plus, nous montrons que `VITS` atteint une borne de regret sous-linéaire du même ordre dans la dimension et le nombre de tours que le TS traditionnel pour les bandits contextuels linéaires. Enfin, nous démontrons expérimentalement l'efficacité de `VITS` sur des jeux de données synthétiques et réels.

1.1.1 Détails des différentes contributions du manuscrit

Nous passons maintenant à la description des contributions de cette thèse. Après un bref aperçu des notions utiles dans le chapitre 1, le manuscrit est organisé en deux parties : la première se concentre sur la théorie des MDPs robustes et en particulier sur la question de la complexité de l'échantillon, tandis que la deuxième étudie la RL robuste d'un point de vue pratique. Ces deux directions ne sont pas orthogonales : au contraire, l'étude théorique des MDPs peut donner des idées sur la façon de concevoir des algorithmes de RL robustes efficaces et l'utilisation d'algorithmes de RL robustes peut donner une intuition sur les MDPs robustes.

Contributions .

Les éléments et résultats présentés dans cette thèse ont été publiés ou sont actuellement en cours d'examen dans les travaux suivants :

- **Pierre Clavier**, Erwan Le Pennec, Matthieu Geist **Towards Minimax Optimality of Model-based Robust Reinforcement Learning** Conference on Uncertainty in Artificial

Intelligence (UAI) 2024 (Oral), Clavier et al. (2023). Covered in Chapter 2.

- **Pierre Clavier** Laixi Shi, Eric Mazumdar, Matthieu Geist, Adam Wierman, Erwan Le Pennec **Near-Optimal Distributionally Robust Reinforcement Learning with General Lp Norms**. NeurIPS 2024, Dans le chapitre 3.
- **Pierre Clavier***, Tom Huix*, Alain Durmus **VITS : Variational Inference Thomson Sampling for contextual bandits** International Conference on Machine Learning 2024, Clavier et al. (2023). Dans le chapitre 8.
- **Pierre Clavier**, Stéphanie Allassonnière, Erwan Le Pennec **Robust Reinforcement Learning with Distributional Risk-averse formulation** International Conference on Machine Learning 2024, Workshop in Responsible Decision Making in Dynamic Environments, Clavier et al. (2022). Dans le chapitre 4.
- **Pierre Clavier**, Emmanuel Rachelson, Erwan Le Pennec, Matthieu Geist. **Bootstrapping Expectiles in Reinforcement Learning** https://arxiv.org/abs/2406.04081. Clavier et al. (2024) Dans le chapitre 5.
- Adil Zouitine*, David Bertoin*, **Pierre Clavier***, Matthieu Geist, Emmanuel Rachelson **Time-Constrained Robust MDPs** Zouitine et al. (2024b) NeurIPS 2024. Dans le chapitre 6.
- Adil Zouitine*, David Bertoin*, **Pierre Clavier***, Matthieu Geist, Emmanuel Rachelson **RRLS : Robust Reinforcement Learning Suite**, Zouitine et al. (2024a), https https://arxiv.org/abs/2406.08406. Dans le chapitre 7.

A propos de mes contributions:

- Je suis le seul premier auteur junior des chapitres 2, 3, 4, et 5. J'ai rédigé l'article, développé le code et les expériences, et je suis à l'origine de l'idée principale. Laixi Shi, chercheuse postdoctorale, a contribué en tant que deuxième auteur au chapitre 4, en apportant des commentaires précieux sur la rédaction et des commentaires sur les résultats. Toutes les preuves ont été discutées avec elle, tandis que les idées et la rédaction sont mon propre travail.
- Pour les chapitres 6, 7 et 8, je suis co-premier auteur aux côtés de mes collègues Adil Zouitine, David Bertouin et Tom Huix. Dans les chapitres 6 et 7, je suis coauteur avec Adil et David. J'ai contribué à parts égales avec Adil Zouitine à la rédaction, aux discussions et au développement des idées. Adil et David se sont davantage concentrés sur la mise en œuvre, tandis que j'ai traité les sections théoriques de manière indépendante. Dans le chapitre 8, j'ai partagé le travail à parts égales avec Tom Huix, l'autre coauteur. Nous avons tous deux contribué à la mise en œuvre, aux preuves et à la rédaction du papier. En outre, j'ai apporté des idées personnelles à la fois dans la preuve et dans la mise en œuvre des algorithmes finaux.

*Equal contribution.

1.2 Introduction

> *In the process of trying to imitate an adult human mind we are bound to think a good deal about the process which has brought it to the state that it is in. We may notice three components.*
>
> - *(a)The initial state of the mind, say at birth,*
> - *(b)The education to which it has been subjected*
> - *(c) Other experience, not to be described as education, to which it has been subjected.*
>
> *Instead of trying to produce a program to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain. Presumably the child brain is something like a notebook as one buys it from the stationer's. Rather little mechanism, and lots of blank sheets. (Mechanism and writing are from our point of view almost synonymous.) Our hope is that there is so little mechanism in the child brain that something like it can be easily programmed. The amount of work in the education we can assume, as a first approximation, to be much the same as for the human child.*
>
> *Alan Turing, Computing Machinery and Intelligence (1950)*

The intelligence behind a decision often becomes apparent only after a significant period has passed. In games, or in life in general, decisions can have impacts that reach far beyond the initial moment of choice, and acting with an awareness of future implications is a critical aspect of intelligence. Recent advancements in machine learning have demonstrated impressive abilities in single-step predictions, such as speech-to-text transcription, protein shape prediction, or image content recognition. However, creating algorithms that can adjust their actions based on anticipated future outcomes remains one of the most significant challenges in contemporary artificial intelligence research. The ability to plan and predict a sequence of actions to address this challenge is typically referred to as *sequential decision making.*

In nature, humans and animals are capable of sequential decision making. For example, neurotransmitters such as dopamine, synthesized in the brains and kidneys of both humans and animals, play a role in modulating reward-motivated behaviors (Berridge 2007). The release of dopamine in anticipation of a rewarding stimulus, or in response to a reward that exceeds expectations (Montague et al. 1996), illustrates how neurochemical mechanisms can adapt behavior in response to environmental stimuli and optimize actions to achieve desired outcomes.

From a computer science and mathematical perspective, Bellman remains a pioneer in the science of sequential decision making. Bellman (1966) was among the first to establish the foundation of reinforcement learning in his renowned work, *Dynamic Programming.* While Bellman's contributions are primarily rooted in computational and mathematical principles, subsequent understanding of biological phenomena has provided new insights into this field.

Finally, *reinforcement learning*, as formally defined by Sutton and Barto (2018), is a mathematical framework that enables agents to interact with their environment and learn behaviors that maximize their cumulative reward over time. Through this process, agents learn to avoid actions that may lead to negative consequences and to act in ways that improve their ultimate outcomes within a given environment.

One type of sequential decision making is known as the *bandit problem*. Bandit algorithms represent a class of approaches primarily designed to address the multi-armed bandit problem (Auer et al. 2002, Lattimore and Szepesvári 2020). In its classical form, the multi-armed bandit problem involves selecting a strategy to maximize profit given n single-armed bandit machines, each with payoffs following distinct and unknown probability distributions. A defining characteristic of this problem is that past decisions do not influence the outcomes of future ones, similar to how each roulette spin in a casino is independent of previous spins, for example. However, this unique characteristic also represents a key limitation: it does not account for the influence of past decisions on future outcomes. This limitation is significant in many sequential decision-making applications, such as games like chess, where the state of the board—determined by both players' moves—heavily influences each subsequent move. Therefore, while bandit algorithms are not suitable for solving general decision-making problems, they are perfectly adapted to the multi-armed bandit problem. The final chapter (8) of this thesis will focus on this specific setting.

Reinforcement learning (RL), by contrast, addresses sequential learning problems in which allowing past decisions to influence future ones has yielded impressive achievements across a wide range of human-related domains, including games (Silver et al. 2017), large language model alignment (Ziegler et al. 2019, Achiam et al. 2023), robotics and control (Kober et al. 2013), and healthcare (Liu et al. 2019, Fatemi et al. 2021). These remarkable accomplishments can be attributed to the extensive volume of interactive data leveraged in the policy's learning process.

However, in certain instances, the available data may be insufficient to train an effective policy, leading to policies that fail to generalize well and exhibit suboptimal performance in real-world applications. As data-driven approaches play an increasingly pivotal role in enhancing diverse aspects of human life, what factors should be taken into account when developing data-driven reinforcement learning algorithms?

- *The notion of Robustness.* Robustness to uncertainty is a highly valued attribute in reinforcement learning (RL). This importance stems from the fact that the performance of a learned policy in the training environment can significantly deteriorate when exposed to the uncertainty and variability of a test environment, which may arise from random disturbances, rare occurrences, or even intentional adversarial attacks (Mahmood et al. 2018). Consequently, developing RL algorithms that can effectively manage such uncertainties is essential to ensure that the learned policies generalize reliably to new environments.

- *The sample efficiency* is a crucial aspect of modern reinforcement learning (RL). As RL problems have become increasingly complex, with larger environments and more sophisticated policy models, RL agents often require extensive data to learn effective policies (Silver et al. 2017, Achiam et al. 2023). This challenge is further compounded by the sequential nature of RL problems, where the environment's complexity can grow exponentially with the length of the horizon. Consequently, enhancing sample efficiency has become an essential research direction to enable RL agents to learn effective policies using limited data and computational resources. From a theoretical standpoint, recent efforts have been directed toward developing a finite-sample framework (Kakade 2003) to assess and compare the sample efficiency of RL algorithms in high-dimensional settings. However, current statistical insights and provable performance guarantees are still insufficient in both theory and practice, largely due to technical challenges and the broad, diverse scope of RL. Thus, advancing the sample efficiency of RL algorithms in high-dimensional environments remains a pressing area for further research.

- *Reproducibility* and *scalability* of RL algorithm performance are also of critical importance. In practical applications, the dimensionality of encountered environments is often substantial,

making the scalability of reinforcement learning (RL) algorithms a key consideration, particularly in scenarios with limited memory and computational resources. Additionally, (Robust) RL is frequently criticized for producing results that are challenging to replicate. Thus, it is essential to develop RL algorithms that can effectively and efficiently scale to high-dimensional environments, as well as to design benchmarks that test the robustness of algorithms with reproducible performance outcomes.

The problem of sample efficiency is presented in Fig 1.1b where the first algorithm requires many samples to converge to a good solution whereas in 1.1a it requires less samples to obtain a good policy. Moreover, robustness purposes is illustrated in Fig 1.2. In this figure, we train all agents with a relative physical parameter (the length of the pole) of 1 and then in testing phase, the physical parameter is changed. The question we want to tackle is how design algorithm robust to these changes of physical parameters - which lead to changes of transition kernel. In Fig 1.2, α controls the robustness induced in the RL algorithm. More details on that matter may be found in Chapter 4.

Can we derive RL algorithms that are sample efficient, robust and scale with the size of the problem and with reproducible performances?

In this thesis, we will tackle all these issues within the framework of *Robust Reinforcement Learning*. As previously stated, reinforcement learning (RL) has achieved significant success in recent years; however, it often faces challenges in robustness and generalization. These challenges primarily arise due to agents overfitting to specific training environments, which can lead to poor performance during deployment. RL agents are typically trained in simulation due to the high cost of interacting with physical systems. However, simulations may contain modeling errors and imprecise parameters, leading to a discrepancy between simulation and reality that can be difficult for the trained policy to handle during transition. Even policies trained directly on real systems may struggle with previously unencountered uncertainties or disturbances. Slight deviations in the environment's parameters, such as mass or friction, can significantly impact a policy's performance, which may be the difference between success and failure in test scenarios (Morimoto and Doya 2005, Pinto et al. 2017).

Robust Markov Decision Processes (RMDPs) have been introduced in Iyengar (2005), Nilim and El Ghaoui (2005) to tackle this problem. This framework is natural, versatile, and leverages insights from distributionally robust optimization and supervised learning (Bertsimas et al. 2018, Blanchet and Murthy 2019, Duchi and Namkoong 2021). In contrast to conventional MDPs, the class of RMDPs provides a more extensive modeling framework, enabling the specification of the shape and magnitude of the uncertainty set. Frequently, the uncertainty set is chosen to be a small ball centered around the nominal kernel with an uncertainty radius σ, with its dimensions and form defined by a metric that measures the distance between probability distributions. To facilitate the tractability of solving RMDPs, the uncertainty set is typically assumed to possess certain structural properties. For example, previous works (Iyengar 2005, Wiesemann et al. 2013) have proposed that the uncertainty set can be decomposed into independent subsets for each state or state-action pair, referred to as s- and (s,a)-rectangularity, respectively. In this thesis, we adopt the assumption of (s,a)-rectangularity for the uncertainty set.From a theoretical point of view, these assumptions will be useful in Chapter 2 and 3 while we will try to avoid it from a practical point of view - as it is restrictive - in Chapter 6.

We now turn to the descriptions of the contributions made in this thesis. After a short background summarizing the useful notions in Chapter 1, this manuscript is organized in two parts: the first one focuses on the theory of Robust MDPs and especially the question of sample complexity; the second studies Robust RL from a practical point of view. These two directions are not orthogonal: on the contrary, the theoretical study of MDPs can provide insights on how

to design efficient Robust RL algorithms, and using Robust RL algorithms gives intuition on the sample complexity of Robust MDPs.

First, we will tackle the question of *sample complexity*. Suppose that one has access to data samples generated by an MDP with a nominal transition kernel, obtained through certain sampling mechanisms. The primary objective of traditional RL is to learn the optimal policy that is specifically tailored to the nominal kernel, for which the minimax sample complexity limit has been well-established (Azar et al. 2013a). In contrast, the goal of distributionally robust RL is to learn a more resilient policy using the same set of data samples by optimizing the worst-case performance when the transition kernel is chosen arbitrarily from a predefined uncertainty set around the nominal kernel. Sample complexity for RMDPs has been studied in (Yang et al. 2022, Panaganti and Kalathil 2022a, Shi et al. 2024) but generally does not directly translate to algorithms that scale up to complex evaluation benchmarks. From a sample complexity perspective, we will demonstrate that RMDPs are no more difficult to learn than classical MDPs for small uncertainty radius σ and can even be simpler to learn when the radius is larger.This finding provides motivation for utilizing RMDPs to develop sample-efficient algorithm in Chapter 2 and 3.

First contributions about sample complexity of Robust MDPs

We focus in Part I on understanding the sample complexity of Robust MDPs. More previously, in Chapter 2, we study the sample complexity of obtaining an ϵ-optimal policy in *Robust* discounted Markov Decision Processes (RMDPs), given only access to a generative model of the nominal kernel. This problem is widely studied in the non-robust case, and it is known that any planning approach applied to an empirical MDP estimated with $\tilde{\mathcal{O}}(\frac{H^3SA}{\epsilon^2})$ samples provides an ϵ-optimal policy, which is minimax optimal. Results in the robust case are much more scarce. For *sa*- (resp *s*-) rectangular uncertainty sets, until recently the best-known sample complexity was $\tilde{\mathcal{O}}(\frac{H^4S^2A}{\epsilon^2})$ (resp. $\tilde{\mathcal{O}}(\frac{H^4S^2A^2}{\epsilon^2})$), for specific algorithms and when the uncertainty set is based on the total variation (TV), the KL or the Chi-square divergences. In this paper, we consider uncertainty sets defined with an L_p-ball (recovering the TV case), and study the sample complexity of *any* planning algorithm (with high accuracy guarantee on the solution) applied to an empirical RMDP estimated using the generative model. In the general case, we prove a sample complexity of $\tilde{\mathcal{O}}(\frac{H^4SA}{\epsilon^2})$ for both the *sa*- and *s*-rectangular cases (improvements of S and SA respectively). When the size of the uncertainty is small enough, we improve the sample complexity to $\tilde{\mathcal{O}}(\frac{H^3SA}{\epsilon^2})$, recovering the lower-bound for the non-robust case for the first time and a robust lower-bound. Finally, we also introduce simple and efficient algorithms for solving the studied L_p robust MDPs.

Second contribution about sample complexity of Robust MDPs

In Chapter 3 , we refine the results of Chapter 2, assuming access to a generative model that samples from the nominal MDP, we examine the sample complexity of RMDPs using a class of generalized L_p norms as the 'distance' function for the uncertainty set, under two commonly adopted *sa*-rectangular and *s*-rectangular conditions. Our results imply that RMDPs can be more sample-efficient to solve than standard MDPs using generalized L_p norms in both *sa*- and *s*-rectangular cases, potentially inspiring more empirical research. We provide a near-optimal upper bound and a matching minimax lower bound for the *sa*-rectangular scenarios. For *s*-rectangular cases, we improve the state-of-the-art upper bound and also derive a lower bound using L_∞ norm that verifies the tightness. Compared to Chapter 2, we improve the sample complexity, showing that it is possible to obtain sample complexity that are lower than in classical MDPs. This part gives a promising avenue to derive algorithm that can achieve lower sample complexity while be more robust on perturbations.

Then, from a practical point of view, we derive *robust RL algorithms*. We show that the ideas from Robust MDPs can be used to design Robust RL algorithms using a Nominal-based Risk-Averse formulation. More specifically, the idea of this class of algorithms is to approximate the inner minimum operator present in the robust Bellman operator (1.37).

Previous work has typically employed a dual approach to the minimum problem, whereby the transition probability is constrained to remain within a specified ball around the nominal transition kernel. In this line of work, (Kumar et al. 2022) derived an approximate algorithm for RMDPs with L_p balls, (Liu et al. 2022) for KL divergence, and we attempt to approximate RMDPs with χ^2 in Chapter 4. Practically, robustness is equivalent to regularization (Derman et al. 2021): for example the SAC algorithm (Haarnoja et al. 2018a) has been shown to be robust due to entropic regularization (Eysenbach and Levine 2021). Finally, Wang et al. (2023) proposes a novel online approach to solve RMDP. Unlike previous works that regularize the policy or value updates, Wang et al. (2023) achieves robustness by simulating the worst kernel scenarios for the agent while using any classical RL algorithm in the learning process. The idea that regularisation and robustness are closely linked will be central in this Thesis in the Chapter 4 and 5. The idea in Chapter 5 is that we avoid estimation of a penalisation or regularisation, and rather estimate the expectile of the value function, which create implicitly robustness. These types of algorithms are mathematically well founded but are only using sample from the nominal transition kernel. Closely related the idea of using sample not only from the nominal kernel, domain randomization (DR) (Tobin et al. 2017) learns a value function which maximizes the expected return on average across a fixed (generally uniform) distribution on the uncertainty set. This method that uses sample from all the uncertainty set will be combined with risk averse formulation in Chapter 5. To tackle the aforementioned challenges of using sample uniquely from the nominal and avoid rectangularity assumptions, one approach using the concept of two-player zero-sum games or min-max is proposed in Chapter 6. Our algorithm is based like many Deep Robust algorithms exist such as M2TD3 Tanabe et al. (2022a), M3DDPG (Li et al. 2019a), or RARL (Pinto et al. 2017) on the two player zero-sum game presented in 6.2.

First contributions about practical Robust Reinforcement Learning

Robust Reinforcement Learning tries to make predictions more robust to changes in the dynamics or rewards of the system. This problem is particularly important when the dynamics and rewards of the environment are learned and estimated from the data. In Chapter 4, we try to approximate the Robust Reinforcement Learning constrained with a χ^2-divergence using an approximate Risk-Averse formulation. We show that the classical Reinforcement Learning formulation can be robustified using Standard deviation penalization of the objective. Two algorithms based on Distributional Reinforcement Learning, one for discrete and one for continuous action space are proposed and tested on classical Gym environment to demonstrate the robustness of the algorithms.

Second contributions about practical Robust Reinforcement Learning

Then, we derive in Chapter 5 new form of implicit robustness in RL using expectile boostraping. Using these technique avoid to estimate a penalisation like in 4. Many classic Reinforcement Learning (RL) algorithms rely on a Bellman operator, which involves an expectation over the next states, leading to the concept of bootstrapping. To introduce a form of pessimism, we propose to replace this expectation with an expectile. In practice, this can be very simply done by replacing the L_2 loss with a more general expectile loss for the critic. Introducing pessimism in RL is desirable for various reasons, such as tackling the overestimation problem (for which classic solutions are double Q-learning or the twin-critic approach of TD3) or robust RL (where transitions are adversarial). We study empirically these two cases. For the overestimation problem, we show that the proposed approach, `ExpectRL`, provides better results than a classic twin-critic. On robust RL benchmarks, involving changes of the environment, we show that our approach is more robust than classic RL algorithms. We also introduce a variation of `ExpectRL` combined with domain randomization which is competitive with state-of-the-art robust RL agents. Eventually, we also extend `ExpectRL` with a mechanism for choosing automatically the expectile value, that is the degree of pessimism.

Third contributions about practical Robust Reinforcement Learning

Subsequently in the Chapter 6, we try to derive a new algorithm without rectangularity assumptions. Robust reinforcement learning often depends on rectangularity assumptions, where adverse probability measures of outcome states are assumed to be independent across different states and actions. This assumption, rarely fulfilled in practice, leads to overly conservative policies. To address this problem, we introduce a new time-constrained robust MDP (TC-RMDP) formulation that considers multifactorial, correlated, and time-dependent disturbances, thus more accurately reflecting real-world dynamics. This formulation goes beyond the conventional rectangularity paradigm, offering new perspectives and expanding the analytical framework for robust RL. We propose three distinct algorithms, each using varying levels of environmental information, and evaluate them extensively on continuous control benchmarks. Our results demonstrate that these algorithms yield an efficient tradeoff between performance and robustness, outperforming traditional deep robust RL methods in time-constrained environments while preserving robustness in classical benchmarks. This study revisits the prevailing assumptions in robust RL and opens new avenues for developing more practical and realistic RL applications.

Finally to do reproducible method find algorithm that can scale, a normalised benchmark RRLS in 7, we test our last algorithm TC-MDPs on this benchmark to create reproducible algorithm that can scale with dimension and with reproducible performances.

Contributions about reproductiblity issues of Robust RL

We introduce the Robust Reinforcement Learning Suite (RRLS), a benchmark suite based on Mujoco environments. RRLS provides six continuous control tasks with two types of uncertainty sets for training and evaluation. Our benchmark aims to standardize robust reinforcement learning tasks, facilitating reproducible and comparable experiments, in particular those from recent state-of-the-art contributions, for which we demonstrate the use of RRLS. It is also designed to be easily expandable to new environments. The source code is available at https://github.com/SuReLI/RRLS.

Finally, in the 8, we tackle the problem of representation of the posterior in the bandit problem using Thompson sampling algorithms with arbitrary posterior distribution learned using Variational inference.

Contributions in Bandit Theory

We introduce and analyze a variant of the Thompson sampling (TS) algorithm for contextual bandits. At each round, traditional TS requires samples from the current posterior distribution, which is usually intractable. To circumvent this issue, approximate inference techniques can be used and provide samples with distribution close to the posteriors. However, current approximate techniques yield to either poor estimation (Laplace approximation) or can be computationally expensive (MCMC methods, Ensemble sampling...). In this paper, we propose a new algorithm, Varational Inference TS (`VITS`), based on Gaussian Variational Inference. This scheme provides powerful posterior approximations which are easy to sample from, and is computationally efficient, making it an ideal choice for TS. In addition, we show that `VITS` achieves a sub-linear regret bound of the same order in the dimension and number of round as traditional TS for linear contextual bandit. Finally, we demonstrate experimentally the effectiveness of `VITS` on both synthetic and real world datasets.

1.2.1 Overview of the manuscript and contributions

We turn to the descriptions of the contributions made in this thesis. After a short background summarizing the useful notions in Chapter 1, the manuscript is organized in two parts: the first one focuses on theory of Robust MDPs and especially question of sample complexity, the second study Robust RL from a practical point of view. These two directions are not orthogonal: on the contrary, the theoretical study of MDPs can give idea on how design efficient Robust RL algorithm and using Robust RL algorithm gives intuition on sample complexity of Robust MDPs.

Contributions The elements and results presented in this thesis have been published or are currently under review in the following works:

- **Pierre Clavier**, Erwan Le Pennec, Matthieu Geist **Towards Minimax Optimality of Model-based Robust Reinforcement Learning** Conference on Uncertainty in Artificial Intelligence (UAI) 2024 (Oral), Clavier et al. (2023). Covered in Chapter 2.
- **Pierre Clavier** Laixi Shi, Eric Mazumdar, Matthieu Geist, Adam Wierman, Erwan Le

Pennec **Near-Optimal Distributionally Robust Reinforcement Learning with General Lp Norms**. NeurIPS 2024, Covered in Chapter 3.

- **Pierre Clavier***, Tom Huix*, Alain Durmus **VITS : Variational Inference Thomson Sampling for contextual bandits** International Conference on Machine Learning 2024, Clavier et al. (2023). Covered in Chapter 8.
- **Pierre Clavier**, Stéphanie Allassonnière, Erwan Le Pennec **Robust Reinforcement Learning with Distributional Risk-averse formulation** International Conference on Machine Learning 2024, Workshop in Responsible Decision Making in Dynamic Environments, Clavier et al. (2022). Covered in Chapter 4.
- **Pierre Clavier**, Emmanuel Rachelson, Erwan Le Pennec, Matthieu Geist. **Bootstrapping Expectiles in Reinforcement Learning** https://arxiv.org/abs/2406.04081. Clavier et al. (2024) Covered in Chapter 5.
- Adil Zouitine*, David Bertoin*, **Pierre Clavier***, Matthieu Geist, Emmanuel Rachelson **Time-Constrained Robust MDPs**, Zouitine et al. (2024b), NeurIPS 2024. Covered in Chapter 6.
- Adil Zouitine*, David Bertoin*, **Pierre Clavier***, Matthieu Geist, Emmanuel Rachelson **RRLS : Robust Reinforcement Learning Suite**, Zouitine et al. (2024a), https https://arxiv.org/abs/2406.08406. Covered in Chapter 7.

About my contributions:

- I am the sole junior first author of Chapters 2, 3, 4, and 5. I wrote the paper, developed the code for implementations and experiments, and originated the main idea. Laixi Shi, a postdoctoral researcher, contributed as a second author to Chapter 4, providing valuable feedback on writing and insights on the results. All proofs were discussed with her, while the ideas and writing are my own work.
- For Chapters 6, 7, and 8, I am co-first author alongside junior colleagues Adil Zouitine, David Bertouin, and Tom Huix. In Chapters 6 and 7, I share co-first authorship with Adil and David. I contributed equally with Adil Zouitine in the writing, discussions, and development of ideas. Adil and David focused more on implementation, while I handled the theoretical sections independently. In Chapter 8, I shared the work equally with Tom Huix, the other co-first author. We both contributed to the implementation, proofs, and writing of the paper. Additionally, I contributed my own ideas to the proof, the implementation and the final algorithms.

*Equal contribution.

1.3 Background

1.3.0.1 Sequential Decision Making and Bandit Problem

In traditional Multi-Armed Bandit (MAB) problems, an agent has to sequentially choose between several actions (referred to as "arms"), from which it receives a reward from the environment. The arm selection process is induced by a sequence of policies, which are inferred and refined at each round from past observations. These policies are designed to optimize the cumulative rewards over the entire process. The main challenge in this task is to effectively manage a suitable exploitation-exploration trade-off (Robbins 1952, Katehakis and Veinott 1987, Berry and Fristedt 1985, Auer et al. 2002, Lattimore and Szepesvári 2020, Kveton et al. 2020). Here, exploitation refers to selecting an arm that is currently believed to be the best based on past observations, while exploration refers to selecting arms that have not been selected frequently in the past in order to gather more information. The classical Bandit problem can be represented in Fig 1.3. Bandits have many applications, such as in agriculture (Gautron et al. 2024), health (Réda 2022), recommendation systems (Li et al. 2010), or model selection in Machine Learning (Pacchiano et al. 2020).

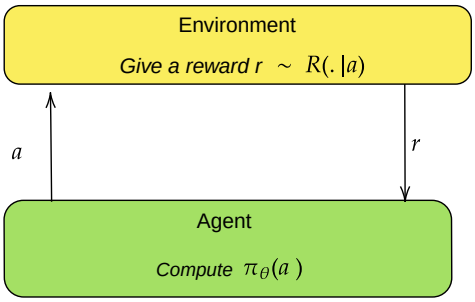


Figure 1.3: A bandit problem

Contextual bandit problems are a particular instance of the MAB problem, which assumes that, at each round, the set of arms and the corresponding rewards depend on a d-dimensional feature vector called a contextual vector or context. This scenario has been extensively studied over the past decades, and learning algorithms have been developed to address this problem (Langford and Zhang 2007, Abbasi-Yadkori et al. 2011, Agrawal and Goyal 2013, Kveton et al. 2020). These algorithms have been successfully applied in several real-world problems, such as recommender systems, mobile health, and finance (Li et al. 2010, Agarwal et al. 2016, Tewari and Murphy 2017, Bouneffouf et al. 2020). The existing algorithms for addressing contextual bandit problems can be broadly categorized into two groups. The first category is based on maximum likelihood and the principle of optimism in the face of uncertainty (OFU) and has been studied in (Auer et al. 2002, Chu et al. 2011, Abbasi-Yadkori et al. 2011, Li et al. 2017, Ménard and Garivier 2017, Zhou et al. 2020, Foster and Rakhlin 2020, Zenati et al. 2022).

The second category consists of randomized probability matching algorithms, which are based on Bayesian belief and posterior sampling. *Thompson Sampling (TS)* is one of the most famous algorithms that falls into this latter category. Since its introduction by Thompson (1933), it has been widely studied, both theoretically and empirically (Agrawal and Goyal 2012, Kaufmann et al. 2012, Agrawal and Goyal 2013, Russo and Van Roy 2014; 2016, Lu and Van Roy 2017, Riquelme et al. 2018, Jin et al. 2021). Despite the fact that OFU algorithms offer better theoretical

guarantees compared to classic TS-based algorithms, traditional TS methodologies still appeal to us due to their straightforward implementation and empirical advantages. In Agrawal and Goyal (2012), the authors claimed that: "In applications like display advertising and news article recommendation, TS is competitive with or better than popular methods such as UCB." Similarly, Chapelle and Li (2011) has examined the empirical performances of TS on both simulated and real data. Their experiments demonstrate that TS outperforms OFU methods, leading them to conclude: "In any case, TS is very easy to implement and should thus be considered as a standard baseline." Taking all these factors into account, we focus on TS-based algorithms for addressing contextual bandit problems in this thesis. The contextual bandit problem can be represented in Fig1.4.

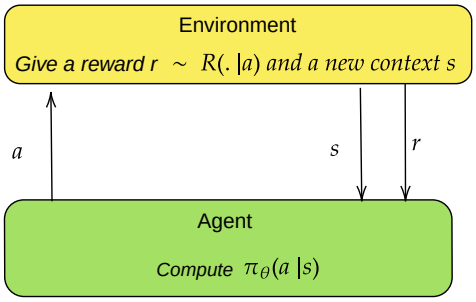


Figure 1.4: Contextual Bandit problem

Thompson sampling for contextual bandits : We now present in more details the contextual bandit framework. Let $\mathcal{S}$ be a contextual space and consider $\mathsf{A} : \mathcal{S} \to 2^{\mathcal{A}}$ a set-valued action map, where $2^{\mathcal{A}}$ stands for the power set of the action space $\mathcal{A}$. For simplicity, we assume here that $\sup_{s\in\mathcal{S}} \mathrm{Card}(\mathsf{A}(s)) < +\infty$. A (deterministic or random) function $\pi : \mathcal{S} \to \mathcal{A}$ is said to be a policy if for any $s \in \mathcal{S}$, $\pi(s) \in \mathsf{A}(s)$. Then, for a fixed horizon $T \in \mathbb{N}^\star$, a contextual bandit process can be defined as follows: at each iteration $t \in [T]$ and given the past observations $\mathrm{D}_{t-1} = \{(s_i, a_i, r_i)\}_{i<t}$:

- The agent receives a contextual feature $s_t \in \mathcal{S}$;
- The agent chooses an action $a_t = \pi_t(s_t)$ where π_t is a policy sampled from $\mathbb{Q}_t(\cdot|\mathrm{D}_{t-1})$;
- Finally, the agent receives a reward r_t sampled from $\mathcal{R}(\cdot|s_t, a_t)$ given D_{t-1}. Here, $\mathcal{R}$ is a Markov kernel on $(\mathcal{A} \times \mathcal{S}) \times \mathsf{R}$, where $\mathsf{R} \subset \mathbb{R}$.

For a fixed family of conditional distributions $\mathbb{Q}_{1:T} = \{\mathbb{Q}_t\}_{t\leq T}$, this process defines a random sequence of policies, $\pi_{1:T} = \{\pi_t\}_{t\leq T}$ with distribution still denoted by $\mathbb{Q}_{1:T}$ by abuse of notation. Let's defined the optimal expected reward for a contextual vector $x \in \mathsf{X}$ and the expected reward given x and any action $a \in \mathcal{A}(s)$ as follow

$$f_\star(s) = \max_{a\in\mathsf{A}(s)} f(s,a) \ , f(s,a) = \int r\mathcal{R}(\mathrm{d}r|s,a) \ . \tag{1.1}$$

The main challenge of a contextual bandit problem is to find the distribution $\mathbb{Q}_{1:T}$ that minimizes the cumulative regret defined as

$$\begin{aligned} &\mathrm{CRegret}(\mathbb{Q}_{1:T}) = \textstyle\sum_{i\leq T} \mathrm{Regret}_i^{\pi_i} \\ &\text{with} \quad \mathrm{Regret}_s^{\pi_i} = f_\star(s_i) - f(x_s, \pi_s(s_i)) \ . \end{aligned} \tag{1.2}$$

The main difficulty in the contextual bandit problem arises from the fact that the reward distribution $\mathcal{R}$ is intractable and must be inferred to find the best policy to minimize the instantaneous regret $\pi \mapsto f_\star(s) - f(s, \pi(s))$ for a context $s \in \mathcal{S}$. However, the estimation of $\mathcal{R}$ may contradict the primary objective of minimizing the cumulative regret (8.2), since potentially non-effective arms must be chosen to obtain a complete description of $\mathcal{R}$. Therefore, bandit learning algorithms have to achieve an appropriate trade-off between the exploitation of arms that have been confidently learned and the exploration of misestimated arms.

Thompson sampling: To achieve such a trade-off, we consider the popular Thompson Sampling (TS) algorithm. Consider a parametric model $\mathcal{R}_\theta, :, \theta \in \mathbb{R}^d$ for the reward distribution, where for any θ, $\mathcal{R}_\theta$ is a Markov kernel on $(\mathcal{A} \times \mathcal{S}) \times \mathbb{R}$ parameterized by $\theta \in \mathbb{R}^d$. We assume in this paper that $\mathcal{R}_\theta$ admits a density with respect to some dominating measure λ_{ref}. For instance, it is possible to use the exponential family defined in 8.3

With the introduced notations, the likelihood function associated to the observations D_t at step $t > 1$ is given by

$$\mathrm{L}_t(\theta) \propto \exp\left\{\sum_{i=1}^{t-1} \ell(\theta|s_i, a_i, r_i)\right\} , \tag{1.3}$$

where the log-likelihood is given by $\ell(\theta|s_i, a_s, s_i) = \log(\mathrm{d}\mathcal{R}_\theta/\mathrm{d}\lambda_{\text{ref}})(r_i|s_i, a_i)$. The symbol $\propto$ denotes a quantity proportional to another. Choosing a prior on θ with density p_0 with respect to Leb, and applying Bayes formula, the posterior distribution at round $t \in [T]$ is given by

$$\hat{\mathrm{p}}_t = \mathrm{L}_t(\theta)\mathrm{p}_0(\theta)/\mathfrak{Z}_t , \tag{1.4}$$

where $\mathfrak{Z}_t = \int \mathrm{L}_t(\theta)\mathrm{p}_0(\theta)\mathrm{d}\theta$ denotes the normalizing constant and we used the convention that $\hat{\mathrm{p}}_1 = \mathrm{p}_0$. Moreover we define the potential function $U(\theta) \propto -\log \hat{\mathrm{p}}_t(\theta)$. Then, at each iteration $t \in [T]$, TS consists in sampling a sample θ_t from the posterior $\hat{\mathrm{p}}_t$ and from it, use as a policy, $\pi_t^{(\mathrm{TS})}(s)$ defined for any s by

$$\pi_t^{(\mathrm{TS})}(s) = a^{\theta_t}(s) \ , a^\theta(s) = \arg\max_a \int r\mathcal{R}_\theta(\mathrm{d}r|s, a) . \tag{1.5}$$

Since $\mathfrak{Z}_t$ is generally intractable, sampling from the posterior distribution is not in general an option. This is why we will use of Variational Inference to approximate the posterior distribution. Other methods such as Laplace (Chapelle and Li 2011), Langevin (Xu et al. 2022) have been proposed approximate and a details overview is presented in Chapter 8. The TS algorithm for contextual bandit is described in Alg. 1.

Algorithm 1: Thompson Sampling for Contextual Bandit

for $t = 1, ..., T$ **do**
  Receive from environment the context s_t.
  Sample θ_t from $\hat{\mathrm{p}}_t$.
  Select a_t such as $a_t = \pi_t^{(\mathrm{TS})}(s_t)$.
  Receive $r_t \sim R(.|s_t, a_t)$.
  Update $\hat{\mathrm{p}}_{t+1}$ using new point (s_t, a_t, r_t).
end for

Variational inference TS: To address this challenge, practitioners often employ approximate inference methods to generate samples from a distribution that is expected to be "close" to the actual posterior distribution. In this context, we specifically concentrate on the application of

Variational Inference (VI). In this scenario, we consider a variational family $\mathcal{G}$, which is a set of probability densities with respect to the Lebesgue measure, from which it is typically easy to sample. Then, ideally, at each round $t \in [T]$, the posterior distribution $\hat{\mathrm{p}}_t$ is approximated by the variational posterior distribution $\tilde{\mathrm{q}}_t$, which is defined as:

$$\tilde{\mathrm{q}}_t = \arg\min_{\mathrm{p}\in\mathcal{G}} \mathrm{KL}(\mathrm{p}|\hat{\mathrm{p}}_t) \ , \tag{1.6}$$

where KL is the Kullback-Leibler divergence is defined in 1.30. In Chapter 8, We will detail how to derive algorithms using Variational Inference in the Thompson Sampling algorithm and evaluate the performance of our algorithm against other approximations of the posterior, such as Laplace in the LMC-TS algorithm (Xu et al. 2022). The main difference between contextual bandits and Markov Decision Processes, which will be central in the rest of the thesis, is that the action chosen in contextual bandits does not affect the next state, contrary to MDPs, making it much harder to find the best policy π.

1.3.1 Reinforcement Learning and Markov Decision Processes

1.3.1.1 Markov Decision Processes

We define a Markov Decision Process to model the interaction between the environment and the agent in Reinforcement Learning. Usally, we use a a discounted, infinite horizon, Markov Decision Process (MDP) $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho)$, specified by:

- $\mathcal{S}$, the state space that can be either finite or infinite. In Chapter 2 and 3, we will assume it is finite for mathematical convenience but we will in Deep Reinforcement Learning Chapter 4,5,6,7 assume it possibly infinite.
- $\mathcal{A}$ the action space, which also may be discrete or infinite. For mathematical convenience, we will assume that $\mathcal{A}$ is finite except in Chapter 4, 5, 6 and 7.
- $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$, the reward function which is the immediate reward associated with taking action a in state s. The reward he $r(s, a)$ could be a random variable where the distribution depends on s, a such as in Chapter 4. However we will focus on the case where $r(s, a)$ is deterministic in more theoretical Chapter 2 and Chapter 3.
- $\gamma \in [0, 1)$, the discount factor which defines a horizon for the problem.
- $\rho \in \Delta(\mathcal{S})$ the initial state distribution which specifies the initial state s_0 sampled.
- $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, the transition function . $P(s' \mid s, a)$ is the probability of transitioning into state s' upon taking action a in state s. We will use $P_{s,a}$ to denote the vector $P(\cdot \mid s, a)$.

1.3.1.2 Value, policy and optimality

Policy: Throughout this thesis, time is assumed to be discrete. A policy, denoted by π, is defined as a mapping from states to distributions over actions. The space of all policies is denoted as $\Pi \in \Delta(\mathcal{A})^{\mathcal{S}}$. A *deterministic* policy assigns a single action to a given state, while a *stochastic* policy may assign positive probabilities to multiple actions for a given state. Finally, the probability assigned by policy π to action a in state s is denoted by $\pi(a \mid s)$. One possible and classical objective is to learn an optimal policy, denoted by π^*, that maximizes the expected cumulative discounted reward, defined as follows:

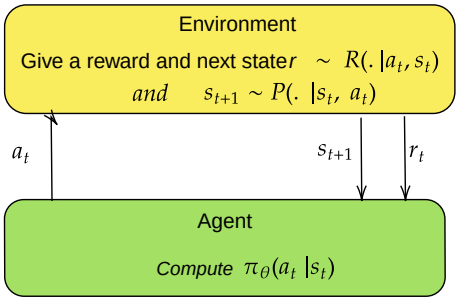


Figure 1.5: Reinforcement Learning framework

$$\pi^* \in \underset{\pi \in \Pi}{\operatorname{argmax}} \mathbb{E}\left[\sum_{t \geq 0} \gamma^t r\left(s_t, a_t\right) \mid s_0 \sim \rho, a_t \sim \pi\left(\cdot \mid s_t\right), s_{t+1} \sim P\left(\cdot \mid s_t, a_t\right)\right]. \tag{1.7}$$

The discounted sum of reward $\sum_{t \geq 0} \gamma^t r\left(s_t, a_t\right)$ is also called the return. Finally, we denote Z the distribution over the return. along a trajectory or rollout τ. Using π from state s using initial action a is defined as the the random sequence $\tau^{P,\pi|s_0,a_0} = \left(\left(s_0, a_0, r_0\right), \left(s_1, a_1, r_1\right), \ldots\right)$ with $s_0 = s, a_0 = a, a_t \sim \pi\left(\cdot \mid s_t\right)$ and $r_t, s_{t+1} \sim P\left(\cdot, \cdot \mid s_t, a_t\right)$; we denote the distribution over rollouts by $\mathbb{P}(\tau)$ with $\mathbb{P}(\tau) = \rho\left(s_0\right) \prod_{t=0}^{T} P\left(s_{t+1}, r_t \mid s_t, a_t\right) \pi\left(a_t \mid s_t\right) d\tau$ and usually write $\tau \sim \mathbb{P} = (\pi, P)$.

Value function: To characterize the cumulative reward, the value function $V^{\pi,P}$ for any policy π under the transition kernel P is defined by $\forall s \in \mathcal{S}$:

$$V^{\pi,P}(s) \coloneqq \mathbb{E}_{(\pi,P)}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \,\middle|\, s_0 = s\right]. \tag{1.8}$$

The expectation is taken over the randomness of the trajectory $\{s_t, a_t\}_{t=0}^{\infty}$ generated by executing the policy π under the transition kernel P, such that $a_t \sim \pi(\cdot \,|\, s_t)$ and $s_{t+1} \sim P(\cdot \,|\, s_t, a_t)$ for all $t \geq 0$. In the same way, the Q function $Q^{\pi,P}$ associated with any policy π under the transition kernel P is defined using expectation taken over the randomness of the trajectory under policy π as

$$Q^{\pi,P}(s,a) \coloneqq \mathbb{E}_{(\pi,P)}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \,\middle|\, (s_0, a_0) = (s, a)\right], \tag{1.9}$$

Moreover, both the Value and Q function follow the so called Bellman equation (Bellman 1957) such as :

$$V^{\pi,P}(s) = \mathbb{E}_{(\pi,P)}\left[\sum_{t \geq 0} \gamma^t r(s_t, a_t) \mid s_0 = s\right] \tag{1.10}$$

$$= \mathbb{E}_{(\pi,P)}\left[r(s,a) + \gamma \sum_{t \geq 1} \gamma^t r(s_t, a_t) \mid s_0 = s, a \sim \pi(s)\right] \tag{1.11}$$

$$= \sum_{a \in \mathcal{A}} \pi(a \mid s) \sum_{s' \in \mathcal{S}} P\left(s' \mid s, a\right) \left(r(s,a) + \gamma \mathbb{E}_{(\pi,P)}\left[\sum_{t \geq 0} \gamma^t r(s_t, a_t) \mid s_0 = s'\right]\right) \tag{1.12}$$

$$= \sum_{a \in \mathcal{A}} \pi(a \mid s) \sum_{s' \in \mathcal{S}} P\left(s' \mid s, a\right) \left(r(s,a) + \gamma V^{\pi,P}\left(s'\right)\right) \tag{1.13}$$

The aforementioned equation creates a connection between the value of a particular state and the values of its ensuing states. This connection is pivotal in dynamic programming and reinforcement learning, as it permits the propagation of value from one state to others within the state space. Similarly, the Q-function, which denotes the quality of an action at a specific state, follows an analogous principle, thereby allowing the transmission of action-value information among diverse states.

$$Q^{\pi,P}(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P\left(s' \mid s,a\right) \sum_{a' \in \mathcal{A}} \pi\left(a' \mid s'\right) Q^{\pi,P}\left(s',a'\right) \tag{1.14}$$

Moreover, a policy π^* is said to be optimal for a MDP if and only if

$$\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}_{s\sim\rho}[V^{\pi,P}(s)] \ ,$$

with ρ the initial state distribution or equivalently (Sutton and Barto 2018), a policy π^* is optimal if and only if

$$\forall \pi \in \Pi, \forall s \in \mathcal{S}, V^{\pi^*,P}(s) \geq V^{\pi,P}(s) \ .$$

1.3.1.3 Bellman Operators and Optimality

The value function $V^{\pi,P}$ for policy π, is the fixed point of the Bellmen operator $\mathcal{T}\pi, P$, defined for any $V \in \mathbb{R}^S$ as

$$\mathcal{T}^{\pi,P}V(s) = \sum_a \pi(a|s)[R(s,a) + \gamma \sum_{s'} P\left(s'|s,a\right) V\left(s'\right)].$$

We also define the optimal Bellman operator: $\mathcal{T}^{*,P}V(s) = \max_{\pi_s \in \Delta_{\mathcal{A}}} \left(\mathcal{T}^{\pi_s,P}V\right)(s)$. Both optimal and classical Bellman operators are γ-contractions (Sutton and Barto 2018). This is why sequences $\{V_n^\pi \mid n \geq 0\}$, and $\{V_n^* \mid n \geq 0\}$, defined as

$$V_{n+1}^\pi := \mathcal{T}^{\pi,P}V_n^\pi \text{ and } \quad V_{n+1}^* := \mathcal{T}^{*,P}V_n^*,$$

converge linearly to $V^{\pi,P}$ and $V^{*,P}$, respectively the value function following π and the optimal value function. Now, we introduce the concept of a greedy policy that connects the optimal policy and its value $V \in \mathbb{R}^{\mathcal{S}}$. We say that a policy is considered greedy with respect to a value function if it always selects the action that maximizes the expected reward based on that value function. In other words, a greedy policy makes locally optimal decisions at each state, assuming that the value function accurately represents the long-term reward. More formally, π is greed with respect to V if an only if

$$T^{\pi,P}V = T^{*,P}V \ .$$

Finally, we define the the space of greedy policies as $\mathcal{G}(V)$. The greediness can be understood state-wise as, for all $a \in \mathcal{A}$ and $s \in \mathcal{S}$,

$$\pi(a \mid s) > 0 \Rightarrow a \in \operatorname*{argmax}_{a' \in \mathcal{A}} \left(r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P\left(s' \mid s,a'\right) V\left(s'\right) \right). \tag{1.15}$$

The action that leads to the state where the value is maximized according to a greedy policy. The interesting property is that optimal policy is greedy with regards to its own value, ie $\pi^* \in \mathcal{G}\left(V^*\right)$,

so it is possible to compute the optimal using associated optimal value function. The optimality can also computed using definition of Q-function. The major interest of definition greedeyness defined like this, is that it allow defining greediness without having access the transition kernel P. So for any $Q \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$, we say that a policy π is greedy w.r.t. Q if and only if

$$\forall s \in \mathcal{S}, \forall a \in \mathcal{A}, \quad \pi(a \mid s) > 0 \Rightarrow a \in \underset{a' \in \mathcal{A}}{\operatorname{argmax}} Q\left(s, a'\right).$$

With this definition, the greedy policy can be found without any knowledge of transition kernel P. Thus, the notion of greediness using Q-values is simpler to define as it is simply the action maximizing the estimated Q-value for each state. Finally we define the Bellman Operator $T^{\pi,P}$ and the optimal Bellman Operator $T^{\pi^*,P}$ for the Q-function as :

$$T^{\pi,P}Q(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P\left(s' \mid s, a\right) \mathbb{E}_{a' \sim \pi(\cdot|s')}\left[Q\left(s', a'\right)\right] , \tag{1.16}$$

$$T^{\pi^*,P}Q(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P\left(s' \mid s, a\right) \max_{a' \in \mathcal{A}} Q\left(s', a'\right) . \tag{1.17}$$

These operators are also a γ- contraction. So we can iterate this operator to converge to the optimal policy π^* and defined $Q^{*,P}$ the fix point with $Q^{*,P} = Q^{\pi^*,P}$.

1.3.1.4 (Approximate) Value Iteration (AVI)

In the previous section, we have established first a method for determining the value of the optimal policy, and then a relationship between the optimal policy and its corresponding value. Combining these elements results in *Value Iteration* which is a Dynamic Programming algorithm that calculates the optimal policy for a MDP. This scheme begins with any initial value $V_0 \in \mathbb{R}^{\mathcal{S}}$, and at each iteration step $k \in \mathbb{N}$:

$$\begin{cases} V_{k+1} = T^{*,P}V_k \\ \pi_{k+1} \in \mathcal{G}\left(V_{k+1}\right). \end{cases} \tag{1.18}$$

To be more accurate, it would be interesting to quantify a stoping criterion such as a number of step to reach arbitrary small error or finding $\epsilon > 0$ such that $\|V_{k+1} - V_k\|_\infty < \epsilon$. The fact that VI asymptotically compute π^* and this error ϵ tend to zero is simply a consequence of Banach's fixed point theorem applied to the optimal Bellman Operator $T^{*,P}$ which is a γ-contraction. As the convergence is asymptotic, VI will never be able to compute the optimal policy π^* as we can bound by

$$\left\|V^{\pi_k,P} - V^{*,P}\right\|_\infty \leq \frac{2}{(1-\gamma)^2}\gamma^k , \tag{1.19}$$

because the current reward $r(s,a)$ belongs to $[0,1]$. So VI converges exponentially fast with linear rate as $\gamma \in [0,1)$ but the error can be sometime very large as $1/(1-\gamma)$ or horizon factor can be very large when γ is close to 1. Moreover, VI can be rewritten using Q function in a model free setting as :

$$\begin{cases} \pi_{k+1} \in \mathcal{G}\left(Q_k\right) \\ Q_{k+1} = T^{\pi_{k+1},P}Q_k . \end{cases} \tag{1.20}$$

Then we will see the influence of error in VI. To take into account errors in the process, we use the *Approximate Dynamic Programming* framework. It defines and analyzes Dynamic Programming schemes, and incorporates additional arbitrary errors such as

$$\begin{cases} \pi_{k+1} \in \mathcal{G}\left(Q_k\right) \\ Q_{k+1} = T^{\pi_{k+1},P} Q_k + \epsilon_{k+1} \ , \end{cases} \tag{1.21}$$

where $\forall k \in \mathbb{N}$, $\epsilon_k \in \mathbb{R}$ are for errors made when computing the Q-values. Firstly, note that this definition does not specify the origin of errors, which can come from various noise sources. Typically, errors are classified into three types:

- **Estimation errors :** these occur due to the large or continuous state space, which prevents an exact tabular representation of Q_k, necessitating function approximation methods such as Neural Network.
- **Sampling errors:** they arise because the transition probability P is unknown in RL setting, requiring states to be sampled from $P(\cdot \mid s, a)$ instead of computing exact expectations.
- **Greediness errors:** errors in computing the greedy policy are not considered in this scheme but this is based on the assumption that the action space is small and discrete, allowing for the straightforward computation of a greedy policy by finding the maximum over a small set. However, in the case of continuous control, these errors must be taken into account.

To better understand the influence of errors on the behavior of a scheme, we will focus on error propagation in AVI. Specifically, we aim to connect the discrepancy between the value of the computed policy and the optimal policy to the errors incurred during iterations. This analysis helps us understand various phenomena, such as how errors accumulate over iterations and the conditions under which we can demonstrate convergence or establish bounds on the distance to the optimal policy. At step or iteration k, Bertsekas (2017) show that

$$\left\| Q^{\pi_k,P} - Q^{*,P} \right\|_\infty \leq \frac{2}{(1-\gamma)^2} \left(\gamma^k + (1-\gamma) \sum_{j=1}^{k} \gamma^{k-j} \left\| \epsilon_k \right\|_\infty \right) \ .$$

In this upper bound, we can recognise two terms : the fist one proportional to γ^k is similar the the scheme without errors and tend to zero when k grows. The second term proportional to $\sum_{j=1}^{k} \gamma^{k-j} \left\| \epsilon_k \right\|_\infty$ does not tend to zero as it is an exponential average of the norms of every errors. It shows that errors have an impact on the current solution and that recent errors have more impact than the older ones. Moreover it is important to know that this upper bound is tight according to Scherrer and Lesner (2012).

A nice modification on this scheme comes from Vieillard et al. (2020) that use KL regularisation of the policy to obtain bound that depend on the average of the errors $\left\| \frac{1}{k} \sum_{j=1}^{k} \epsilon_j \right\|_\infty$ and not $\sum_{j=1}^{k} \gamma^{k-j} \left\| \epsilon_k \right\|_\infty$. This modification is central in many state of the art Deep RL algorithm such as TRPO or PPO (Schulman et al. 2015; 2017a) or more recently in Munchausen algorithm (Vieillard et al. 2020). Using this modification, if we assume that errors have zero mean lead to convergence in average which is not the case without this modification and this modified scheme play an averaging thought steps of the iterations. However, for example when the model used in far from the initial one, the error have not zero mean and iterating over the step k do not necessary converge to a good solution. To give an example, assume that we are trying to find

the best policy but having access to a transition kernel P' which slightly differ from P. Then we obtain iterating AVI:

$$\begin{cases} \pi_{k+1} \in \mathcal{G}\left(Q_k\right) \\ Q_{k+1} = T^{\pi_{k+1},P'} Q_k + \epsilon_{k+1} \, . \end{cases} \tag{1.22}$$

with using vector notation $P_{s,a}$ for kernel starting at state action (s,a) definition of

$$\epsilon_k = T^{\pi_{k+1},P} Q_k - T^{\pi_{k+1},P'} Q_k = \gamma(P^{\pi_{k+1}} - P^{',\pi_{k+1}}) Q_k$$

with $P^{\pi}_{s,a} = \sum_{s' \in \mathcal{S}} P\left(s' \mid s,a\right) \mathbb{E}_{a' \sim \pi(\cdot|s')} \left[Q_k\left(s',a'\right)\right]$. This error is not centered as long as the expectation of $(P^{\pi_{k+1}} - P'^{\pi_{k+1}})$ is not zero due to model misspecification, or changes in the transitions kernel thought iterations.

Starting from this fact, *our goal is to develop a reinforcement learning procedure that is more robust to errors arising from model misspecifications*, especially in the transition kernel, thereby motivating the need for robustness in RL from a theoretical point of view. Achieving this requires modifications to the underlying algorithm. A significant portion of this manuscript is devoted to identifying and implementing these modifications. To solve this problem, a central idea for creating robustness would be to develop an algorithm that, in practice, has a lower value function but can generalize better in an environment that is not exactly the same as the training environment. Two questions arise from a theoretical point of view:

1. *Can we design algorithms which are robust to these model misspecifications and errors?*
2. *Can we estimate the number of data N we need to get arbitrary small error ϵ in (robust) RL algorithm?*

While the first question will be address in more practical Deep RL algorithm in chapter 4, 5 and 6 we will first focus on the sample complexity question.

1.3.1.5 AVI with a generative model in model based setting

In this part we try to answer the second question of the previous part about sample complexity. The next paragraph discuss sample complexity related work in RL.

Classical reinforcement learning with finite-sample guarantees. A recent surge in attention for RL has leveraged the methodologies derived from high-dimensional probability and statistics to analyze RL algorithms in non-asymptotic scenarios. Substantial efforts have been devoted to conducting non-asymptotic sample analyses of standard RL in many settings. Illustrative instances encompass investigations employing Probably Approximately Correct (PAC) bonds in the context of *generative model* settings (Kearns and Singh 1999, Beck and Srikant 2012, Li et al. 2022, Chen et al. 2020, Azar et al. 2013b, Sidford et al. 2018, Agarwal et al. 2020, Li et al. 2023; 2020, Wainwright 2019) and the *online setting* via both in PAC-base or regret-based analyses (Jin et al. 2018, Bai et al. 2019, Li et al. 2021, Zhang et al. 2020, Dong et al. 2019, Jin et al. 2020, Li et al. 2023, Jafarnia-Jahromi et al. 2020, Yang et al. 2021) and finally *offline setting* (Rashidinejad et al. 2021, Xie et al. 2021, Yin et al. 2021, Shi et al. 2022, Li et al. 2022, Jin et al. 2021, Yan et al. 2022).

In the rest of this introduction and in Chapter 2 and 3, we assume having access to a *generative model*. Following (Kearns and Singh 1999), we assume access to a generative model or

a simulator which allows us to collect N independent samples for each state-action pair generated based on the *nominal* kernel P: $\forall(s,a) \in \mathcal{S} \times \mathcal{A}$, $s_{i,s,a} \overset{i.i.d}{\sim} P(\cdot \,|\, s,a), \quad i = 1, 2, \cdots, N$. The total sample size is, therefore, NSA. We consider a model-based approach tailored to MDPs, which first constructs an empirical nominal transition kernel based on the collected samples and then applies value iteration to compute an optimal policy. As we will decouple the statistical estimation error and the optimization error, we need to exhibit an algorithm that can achieve arbitrary small error ϵ_{opt} in the empirical MDP defined as an empirical nominal transition kernel $\widehat{P} \in \mathbb{R}^{SA \times S}$ that can be constructed on the basis of the empirical frequency of state transitions, i.e. $\forall(s,a) \in \mathcal{S} \times \mathcal{A}$

$$\widehat{P}(s'|s,a) \coloneqq \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\{s_{i,s,a} = s'\} \; . \tag{1.23}$$

From an AVI point of view we get:

$$\begin{cases} \pi_{k+1} \in \mathcal{G}\left(Q_k\right) \\ Q_{k+1} = \hat{T}^{\pi_{k+1},P} Q_k + \epsilon_{k+1} \; , \end{cases} \tag{1.24}$$

where $\epsilon_{k+1} = T^{\pi_{k+1},P} Q_k - \hat{T}^{\pi_{k+1},P} Q_k = \gamma(P^{\pi} - \hat{P}^{\pi}) Q_k$. As we see, in this setting, if the estimate kernel converge to true transition kernel, we can control the error yo be arbitrary small. So a question is can we find a minimum number of data N such as ϵ is the sufficiently small ? Specifically, given some target accuracy level $\varepsilon > 0$, the goal is to seek an ε-optimal robust policy $\widehat{\pi}$, the policy estimated in the empirical MDP obeying

$$\forall s \in \mathcal{S} : \quad Q^{\star,P}(s,a) - Q^{\widehat{\pi},P}(s,a) \leq \varepsilon \qquad \text{with} \quad \widehat{Q}^{\widehat{\pi}^{\star},P} - \widehat{Q}^{\widehat{\pi},P} \leq \varepsilon_{\mathsf{opt}} \; . \tag{1.25}$$

This formulation allows plugging any solver of MDPs in this bound as long as we get $\varepsilon_{\mathsf{opt}}$ error. Using VI, we can bound the optimisation term $\varepsilon_{\mathsf{opt}}$ by $\frac{2\gamma^k}{(1-\gamma)^2}$ using (1.19) at iteration k of our algorithm, but we could also plug any algorithm and consider arbitrary $\hat{\pi}$, using this decomposition.

$$\begin{aligned} Q^{\star,P} - Q^{\widehat{\pi},P} &= \left(Q^{\pi^{\star},P} - \widehat{Q}^{\pi^{\star},P}\right) + \left(\widehat{Q}^{\pi^{\star},P} - \widehat{Q}^{*,P}\right) + \left(\widehat{Q}^{*,P} - \widehat{Q}^{\widehat{\pi},P}\right) + \left(\widehat{Q}^{\widehat{\pi},P} - Q^{\widehat{\pi},P}\right) \\ &\overset{\text{(i)}}{\leq} \left(Q^{\pi^{\star},P} - \widehat{Q}^{\pi^{\star},P}\right) + \left(\widehat{Q}^{*,P} - \widehat{Q}^{\widehat{\pi},P}\right) + \left(\widehat{Q}^{\widehat{\pi},P} - Q^{\widehat{\pi},P}\right) \end{aligned}$$

where we use the fact (i) that $\widehat{Q}^{\pi^{\star},P} \leq \widehat{Q}^{*,P}$. Then a natural decomposition using triangular inequality is

$$\|Q^{*,P} - Q^{\hat{\pi},P}\|_{\infty} \leq \underbrace{\|Q^{*,P} - \hat{Q}^{\pi^{*},P}\|_{\infty}}_{\text{statistical error I}} + \underbrace{\|\hat{Q}^{*,P} - \hat{Q}^{\hat{\pi},P}\|_{\infty}}_{\text{optimisation error}} + \underbrace{\|\hat{Q}^{\hat{\pi},P} - Q^{\hat{\pi},P}\|_{\infty}}_{\text{statistical error II}} .$$

Here the problem is to find the number of data N needed to get an arbitrary small error on the two statistical error terms which do not depend on the number of steps for arbitrary policy from the data $\hat{\pi}$. On the contrary the optimisation term decrease when the number of step increase but does not depend on N. Indeed, in model free setting we construct and estimate of the data called $\hat{P}$ find a planner in this empirical MDP. In Agarwal et al. (2020), for $\delta \geq 0$ and for an appropriately chosen absolute constant c, we have with probability greater than $1 - \delta$:

$$\underbrace{\|Q^{*,P} - \hat{Q}^{\pi^{*},P}\|_{\infty}}_{\text{statistical error I}} \leq \gamma \sqrt{\frac{c}{(1-\gamma)^3} \frac{\log(cSA/\delta)}{N}} + \frac{c\gamma}{(1-\gamma)^3} \frac{\log(c|SA/\delta)}{N} \; .$$

The second statistical error $\|\hat{Q}^{\hat{\pi},P} - Q^{\hat{\pi},P}\|_\infty$ can also be bounded with the same term with additional term from $\varepsilon_{\mathsf{opt}}$. Finally, Agarwal et al. (2020) provided for $\epsilon \leq \sqrt{\frac{1}{1-\gamma}}$, that the number of samples from a generative model requires are

$$\#\text{ samples from generative model } = |\mathcal{S}||\mathcal{A}|N \geq \frac{c|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3}\frac{\log(cSA/\delta)}{\epsilon^2} ,$$

then with probability greater than $1-\delta$,

$$\|Q^{*,P} - \hat{Q}^{\pi^*,P}\|_\infty \leq \epsilon .$$

So the overall sample complexity needed to get arbitrary small statistical error ϵ is

$$\#\text{ samples from generative model } = \tilde{\mathcal{O}}\Big(\frac{SA}{(1-\gamma)^3\epsilon^2}\Big) . \tag{1.26}$$

Moreover a minimax lower bound with the same complexity exist from Azar et al. (2013b). However in practice, the sample complexity can be very large as for γ close to 1, $1/(1-\gamma)^3$ is very big. The question is,

Can we find other formulation of RL with smaller sample complexity to converge quicker from a theoretical point of view?

Ideally, to obtain a solution with smaller sample complexity, the value function would have less variability while converging to a reasonable solution. In the minimax lower bound, dependency S A are difficult to improve as the number of samples from generative model is equals to NSA, which is linear in the number of state and action space. From a theoretical point of view, Bernstein's concentration inequality is used to control statistical terms. More formally, the statistical error, up to constant, logarithmic term and second order term, is controlled using Bernstein's inequality (Vershynin 2018) by $\sqrt{\frac{\mathbb{V}_P(V)}{N}}$ where P is a transition kernel and V a value function. The only factor here that could easily be reduced would be the variance of the value function using a new formulation.

Surprisingly, we will see that the issue of reducing sample complexity, reducing the variability of the value function and developing algorithms that are robust to model misspecification are closely related. Indeed, we will see that the formulation of Robust MDPs will also reduce the variance of the value function and the sample complexity. This idea will be developed further in Chapter 3. Before this, we introduce some elements of Deep Reinforcement learning in the next part.

1.3.2 Deep Reinforcement Learning

In this section, we introduce elements of classical Deep Reinforcement Learning that will be useful to derive Deep Robust algorithm in the following of the thesis. First, we introduce Fitted Q-learning and Q-learning, which will be useful to tackle the problem if MDPs but with continuous state space.

1.3.2.1 Fitted Q-learning and Q-learning

First, we describe the Neural Fitted Q-learning introduced by Riedmiller (2005), then we will see the difference with classical Deep Q-learning algorithm. First, we consider the following approximation scenario. Suppose the state space is continuous or too large for a tabular representation (we still assume the action space is small and finite). To learn an appropriate

Q-function, we must use function approximation. We start by describing a simple setting. Assume we have a fixed dataset of transitions $\mathcal{D} = \{(s_t, a_t, r_t, s'_t)\}$, where for each timestep t, $s_t \sim P(\cdot \mid s_t, a_t)$ and $r_t = r(s_t, a_t)$. The actions a_t are chosen by an arbitrary policy, which we do not consider here.

We can then parameterize a function to represent the Q-value within a hypothesis space, typically using neural networks. We denote such a parameterized function as Q_θ. For simplicity, we define the reward function over $\mathcal{S} \times \mathcal{A}$. This can be viewed as taking the expectation over the resulting states of a reward function defined on $\mathcal{S} \times \mathcal{A} \times \mathcal{S}$, in which case the evaluation would be stochastic. From an *Approximate Dynamic Programming* point of view, Q_θ models Q_{k+1} in the VI scheme. Then, we maintain a fixed version of Q_θ to model Q_k, denoted as $Q_{\bar{\theta}}$, where the weights $\bar{\theta}$ are periodically updated by copying from θ. Subsequently, we iteratively minimize the following loss using classical gradient descent, demoting $\hat{\mathbb{E}}_\mathcal{D}$ the empirical expectation over transitions in dataset such as

$$\mathcal{L}(\theta) = \hat{\mathbb{E}}_\mathcal{D}\left[\left(r_t + \max_{a' \in \mathcal{A}} Q_{\bar{\theta}}\left(s'_t, a'\right) - Q_\theta\left(s_t, a_t\right)\right)^2\right] . \tag{1.27}$$

Finally, minimizing this loss can be seen as a method called *Temporal differences*. The classic temporal difference (TD) approach consist in estimating the quantity $Q_k(s, a)$ by performing a regression on targets of the form $r(s, a) + \gamma \sum_{a' \in \mathcal{A}} \pi_k\left(a' \mid s'\right) Q_{k-1}\left(s', a'\right)$. This can be formally express as calculating $Q_{k+1} = T^{\pi_k, P} Q_{k-1} + \epsilon_k$.

Algorithm 2: Neural Fitted-Q

Input , dataset of transitions: $\mathcal{D}$, learning steps: $K \in \mathbb{N}$, update period: $I \in \mathbb{N}$, learning rate: $\eta \in \mathbb{R}$, batch size: $B \in \mathbb{N}$, discount factor: γ
Output θ_{NFQ}
Initialize online weights: θ
Initialize target weights: θ'
for $k \in \{0, \ldots, K-1\}$ **do**
  for $i \in \{0, \ldots, I-1\}$ **do**
    Draw uniformly a batch $\mathcal{B} = \left\{\left(s_j, a_j, r_j, s'_j\right)\right\}_{j=1}^{B}$ from $\mathcal{D}$
    Compute the targets: $\forall 1 \leq j \leq B, \quad y_j \leftarrow r_j + \gamma \max_{a' \in \mathcal{A}} Q_{\theta'}\left(s_j, a'\right)$
    Compute: $\mathcal{L}(\theta) \leftarrow \sum_{j=1}^{B}\left(Q_\theta\left(s_j, a_j\right) - y_j\right)^2$
    Update the online weights: $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta)$
  end for
  Update target network: $\theta' \leftarrow \theta$
end for
return $\theta_{\text{NFQ}} = \theta$

Deep Q-Networks The Deep Q-Network (DQN) is an implementation of Approximate Value Iteration (AVI) that utilizes a neural network as the function approximation during the regression step (learning). Unlike Neural Fitted-Q, where the dataset $\mathcal{D}$ is fixed, data collection (acting) in DQN is a continuous process that occurs concurrently with learning. Specifically, the dataset $\mathcal{D}$, also known as the replay buffer, is managed as a First-In First-Out (FIFO) queue. The data is collected by interacting with the environment using an ϵ-greedy policy $\pi_{\theta,\epsilon}$ (here ϵ is not the error as before), defined as:

$$\pi_{\theta,\epsilon} = (1-\epsilon)\pi_\theta + \epsilon \pi_{\text{U}} ,$$

where $\pi_\theta \in \mathcal{G}(Q_\theta)$ and π_U is the uniform policy. The DQN algorithm consists of two main processes: *acting and learning*. These processes share the weights θ of the online network and the replay buffer $\mathcal{D}$. This Deep Q-Network (DQN) algorithm introduced by Mnih et al. (2013;

Algorithm 3: Acting process in DQN

Input : replay buffer: $\mathcal{D}$, environment: E
Shared : online weights: $\theta \in \mathbb{R}^{\mathcal{N}}$
while True, do **do**
  $a \leftarrow \text{Sample}\left(\pi_{\theta,\epsilon}(. \mid s)\right)$
  $r(s,a), s' \leftarrow \text{Step}(E,a)$
  Put $(s, a, r(s,a), s')$ in $\mathcal{D}$
  $s \leftarrow s'$
end while

2015) serves as the foundation for many of the methods explored in this manuscript. DQN is a groundbreaking approach in reinforcement learning (RL), particularly recognized for its success in establishing a functional deep RL framework on the Atari benchmark (Bellemare et al. 2013). Some algorithm enhancements have been introduced such that :

- *Double DQN (DDQN)* addresses the issue of target overestimation (Van Hasselt et al. 2016).
- *Prioritized Experience Replay* prioritizes sampling transitions with higher temporal-difference (TD) errors (Schaul et al. 2015).
- *Architectural Enhancements* : the dueling Architecture provides less-biased estimates of actions not taken by the agent (Wang et al. 2016).
- *Distributional Reinforcement Learning* aims to learn the entire distribution of returns, rather than just the expected returns, using either a categorical approach (Bellemare et al. 2017) or a quantile approach (Dabney et al. 2018a). In Chapter 4 we will use this improvement to derive a risk averse version of DQN to create Robustness.
- *Regularization techniques* such as the Munchausen algorithm have been proposed to enhance the classical DQN algorithm in the munchausen algorithm Vieillard et al. (2020). To improve robustness, we will also use regularisation but in another manner as we will not regularise with the policy but with the value function itself in Chapter 4.

1.3.2.2 Actor-Critic Methods

Actor-critic methods differ from those derived from Q-learning. These methods typically involve two main components: a value network (critic) that estimates the value of the current state and a policy network (actor) that selects actions based on the current state and. The policy network is updated using policy gradients method (Williams 1992), with the critic's value serving as a baseline to reduce update variance. The value network, similar to Q-learning, is updated using standard temporal-difference (TD) updates described in 1.3.2.1, which involve bootstrapping. The main advantage of these methods is that the tackle the problem of continuous actions space contrary to DQN based methods using policy gradient. Unlike Q-learning methods such as DQN, which can utilize off-policy data from a replay buffer, standard actor-critic methods are on-policy. This means they learn exclusively from interaction data generated by the current policy.

There is significant literature on deep reinforcement learning actor critic that employs regularization techniques. Several algorithms have been developed based on the principle that constraining policy updates to be smooth can enhance performance. Notable examples include are SQL (Azar et al. 2011) , TRPO, (Schulman et al. 2015), PPO (Schulman et al. 2017a), SAC (Haarnoja et al. 2018b) algorithms. Moreover, other algorithm without regularisation but based on policy gradient such as TD3 (Fujimoto et al. 2018) can achieve state of the art performances on continuous control. Based on SAC (Haarnoja et al. 2018b) and TD3 (Fujimoto et al. 2018) algorithm we will derived new algorithm that are robust with continuous action and state space in Chapter 4, 5, and 6.

1.3.3 Robust Markov Decision Processes

Motivated both in theory and practice in Section 1.3.1.5 and 1.3.2.2, we consider distributionally robust MDPs (RMDPs) in the discounted infinite-horizon setting, denoted by $\mathcal{M}_{\mathsf{rob}} = \{\mathcal{S}, \mathcal{A}, \gamma, \mathcal{U}^{\sigma}_{\|.\|}(P^0), r\}$, where $\mathcal{S}, \mathcal{A}, \gamma, r$ are the same sets and parameters as in standard MDPs. The main difference compared to standard MDPs is that instead of assuming a fixed transition kernel P, it allows the transition kernel to be arbitrarily chosen from a prescribed uncertainty set $\mathcal{U}^{\sigma}_{\|.\|}(P^0)$ centered around a *nominal* kernel $P^0 : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, where the uncertainty set is specified using some metric denoted $\|.\|$ defined in of radius $\sigma > 0$. In particular, given the nominal transition kernel P^0 and some uncertainty level σ, the uncertainty set—with arbitrary metric $\| \| : \mathbb{R}^{\mathcal{S}} \times \to \mathbb{R}^{+}$ in *sa* rectangular case or from $\mathbb{R}^{\mathcal{S} \times \mathcal{A}}$ in the *s*-rectangular case, is specified as $\mathcal{U}^{\sigma}_{\|.\|}(P^0) \coloneqq \otimes_{s,a} \mathcal{U}^{\mathsf{sa},\sigma}_{\|.\|}(P^0_{s,a})$, illustrated in Fig 1.6 and defined bellow

$$\mathcal{U}^{\mathsf{sa},\sigma}_{\|.\|}(P^0_{s,a}) \coloneqq \left\{ P_{s,a} \in \Delta(\mathcal{S}) : \left\| P_{s,a} - P^0_{s,a} \right\| \leq \sigma \right\}, \tag{1.28}$$

$$P_{s,a} \coloneqq P(\cdot \mid s, a) \in \mathbb{R}^{1 \times S}, P^0_{s,a} \coloneqq P^0(\cdot \mid s, a) \in \mathbb{R}^{1 \times S}. \tag{1.29}$$

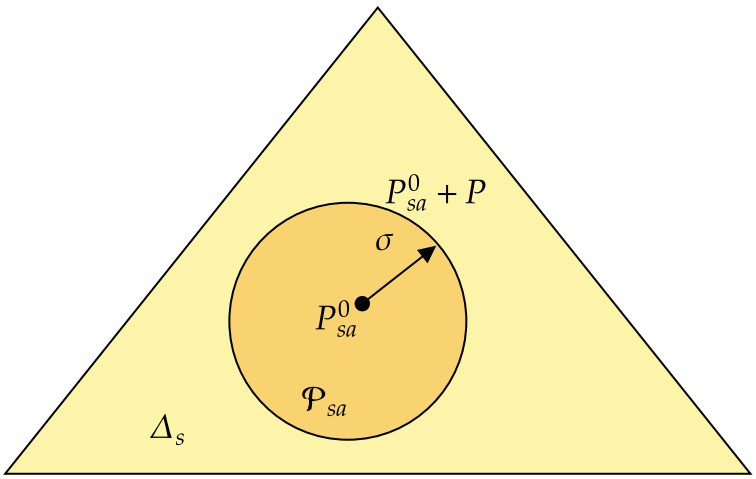


Figure 1.6: One *sa*-uncertainty set for transition probability

Note that we could also consider any divergence ρ, such as KL or χ^2 rather than a metric $\|.\|$

$$\mathrm{KL}\left(P_{s,a}, P^0_{s,a}\right) := \sum_{s' \in \mathcal{S}} P\left(s' \mid s, a\right) \log\left(\frac{P\left(s' \mid s, a\right)}{P^0\left(s' \mid s, a\right)}\right), \tag{1.30}$$

$$\chi^2\left(P_{s,a}, P^0_{s,a}\right) := \sum_{s' \in \mathcal{S}} P^0\left(s' \mid s, a\right) \left(1 - \frac{P\left(s' \mid s, a\right)}{P^0\left(s' \mid s, a\right)}\right)^2 \tag{1.31}$$

but in Chapter 2 and 3 we will consider metric such as L_p. In other words, the uncertainty is imposed in a decoupled manner for each state-action pair, obeying the so-called *sa*-rectangularity (Zhou et al. 2021, Wiesemann et al. 2013). More generally, we define *s*-rectangular MDPs as $\mathcal{U}^{\sigma}_{\|.\|}(P) = \otimes_s \mathcal{U}^{\mathsf{s},\widetilde{\sigma}}_{\|.\|}(P_s)$, for the general norm $\|.\|$. The uncertainty is imposed in a decoupled manner for each state pair, and a fixed budget given a state for all action is defined. To get a similar meaning for the radius of the ball between *sa*-rectangular and *s*-rectangular assumptions, we need to rescale the radius depending on the norm like in Yang et al. (2022). The *s*- uncertainty set is then defined using the rescaled radius $\tilde{\sigma}$ as

$$\mathcal{U}^{\mathsf{s},\widetilde{\sigma}}_{\|.\|}(P_s) := \left\{P'_s \in \Delta(\mathcal{S})^{\mathcal{A}} : \|P'_s - P_s\| \leq \tilde{\sigma} = \sigma\|1_A\|\right\}, \tag{1.32}$$

$$P_s := P(\cdot, \cdot \mid s) \in \mathbb{R}^{1 \times SA}, \quad P^0_s := P^0(\cdot, \cdot \mid s) \in \mathbb{R}^{1 \times SA}, \tag{1.33}$$

where $1_A \in \mathbb{R}^A$ denotes the unitary vector. For the specific case of respectively L_1,L_p and L_∞ norm, $\tilde{\sigma}$ is equal to $|\sigma\mathcal{A}|, \sigma|\mathcal{A}|^{1/p}$ and σ. Note that this scaling allows for a fair comparison between *sa*- and *s*-rectangular MDPs. In RMDPs, we are interested in the worst-case performance of a policy π over all the possible transition kernels in the uncertainty set. This is measured by the *robust value function* $V^{\pi,\sigma}$ and the *robust Q-function* $Q^{\pi,\sigma}$ in $\mathcal{M}_{\mathsf{rob}}$, defined respectively as $\forall (s,a) \in \mathcal{S} \times \mathcal{A}$

$$V^{\pi,\sigma}(s) := \inf_{P \in \mathcal{U}^{\mathsf{sa},\sigma}_{\|.\|}(P^0)} V^{\pi,P}(s)\ , \quad Q^{\pi,\sigma}(s,a) := \inf_{P \in \mathcal{U}^{\mathsf{sa},\sigma}_{\|.\|}(P^0)} Q^{\pi,P}(s,a)\ . \tag{1.34}$$

Similarly for *s*-rectangularity, the value function is denoted $V^{\pi,\tilde{\sigma}}(s) := \inf_{P \in \mathcal{U}^{\mathsf{s},\widetilde{\sigma}}_{\|.\|}(P^0)} V^{\pi,P}(s)$.

Optimal robust policy and robust Bellman operator. As a generalization of properties of standard MDPs in the *sa*-rectangular robust case, it is well-known that there exists at least one deterministic policy that maximizes the robust value function (resp. robust Q-function) simultaneously for all states (resp. state-action pairs) (Iyengar 2005, Nilim and El Ghaoui 2005) but not in the *s*-rectangular case. Therefore, we denote the *optimal robust value function* (resp. *optimal robust Q-function*) as $V^{\star,\sigma}$ (resp. $Q^{\star,\sigma}$), and the optimal robust policy as $\pi^\star$, which satisfy $\forall (s,a) \in \mathcal{S} \times \mathcal{A}$

$$V^{\star,\sigma}(s) := V^{\pi^\star,\sigma}(s) = \max_\pi V^{\pi,\sigma}(s), \quad Q^{\star,\sigma}(s,a) := Q^{\pi^\star,\sigma}(s,a) = \max_\pi Q^{\pi,\sigma}(s,a). \tag{1.35a}$$

A key concept in RMDPs is a generalization of Bellman's optimality principle, encapsulated in the following *robust Bellman consistency equation* (resp. *robust Bellman optimality equation*):

$$\forall (s,a) \in \mathcal{S} \times \mathcal{A}, \quad Q^{\pi,\sigma}(s,a) = r(s,a) + \gamma \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}_{\|.\|}(P^0_{s,a})} \mathcal{P}V^{\pi,\sigma}, \tag{1.36a}$$

$$\forall (s,a) \in \mathcal{S} \times \mathcal{A} \quad Q^{\star,\sigma}(s,a) = r(s,a) + \gamma \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}_{\|.\|}(P^0_{s,a})} \mathcal{P}V^{\star,\sigma}\ , \tag{1.36b}$$

for the *sa*-rectangular case and same equation replacing $P^0_{s,a}$ by P^0_s and σ by $\tilde{\sigma}$. The robust Bellman operator (Iyengar 2005, Nilim and El Ghaoui 2005) is denoted by $\mathcal{T}^{\pi,\sigma}$ or $\mathcal{T}^{*,\sigma}(\cdot) : \mathbb{R}^{SA} \to \mathbb{R}^{SA}$ or for the optimal robust Bellman operator

$$\mathcal{T}^{\pi,\sigma}(Q^\pi)(s,a) \coloneqq r(s,a) + \gamma \inf_{\mathcal{P}\in\mathcal{U}^{\mathsf{sa},\sigma}_{\|.\|}(P^0_{s,a})} \mathcal{P}V^\pi, \quad \text{with} \quad V^\pi(s) \coloneqq \mathbb{E}_{a'\sim\pi}[Q^\pi(s,a)]\,, \tag{1.37}$$

$$\mathcal{T}^{*,\sigma}(Q^\pi)(s,a) \coloneqq r(s,a) + \gamma \inf_{\mathcal{P}\in\mathcal{U}^{\mathsf{sa},\sigma}_{\|.\|}(P^0_{s,a})} \mathcal{P}V, \quad \text{with} \quad V(s) \coloneqq \max_\pi Q^\pi(s,a)\,, \tag{1.38}$$

for *sa*-rectangular MDPs. When the radius is not defined, we will also denote in this Thesis the Robust Bellman Operator as $\mathcal{T}^\pi_{\mathcal{U}}$ for uncertainly set $\mathcal{U}$.

Distributionally Robust Value Iteration (DRVI) Given that $Q^{\star,\sigma}$ is the unique-fixed point of $\mathcal{T}^\sigma$ one can recover the optimal robust value function and Q-function using a procedure termed *distributionally robust value iteration* ($DRVI$). Generalizing the standard value iteration, $DRVI$ starts from some given initialization and recursively applies the robust Bellman operator until convergence. As has been shown previously, this procedure converges rapidly due to the γ-contraction property of $\mathcal{T}^{*,\sigma}$ with respect to the L_∞ norm (Iyengar 2005, Nilim and El Ghaoui 2005).

$$\begin{cases} \pi_{k+1} \in \mathcal{G}\left(Q_k\right) \\ Q_{k+1} = T^{\pi_{k+1},\sigma} Q_k \end{cases} \tag{1.39}$$

Two questions raised once this framework defined to solve Robust MDps problem :

1. As for classical MPDs, the question of sample complexity using DRVI (and not VI) will be adressed in Chapter 2 and 3 or how to find

$$\forall s \in \mathcal{S}: \quad V^{\star,\sigma}(s) - V^{\widehat{\pi},\sigma}(s) \leq \varepsilon\,, \tag{1.40}$$

$$\widehat{V}^{\widehat{\pi}^\star,\sigma} - \widehat{V}^{\widehat{\pi},\sigma} \leq \varepsilon_{\mathsf{opt}}\,. \tag{1.41}$$

   Here the problem is slightly different as the target we try to learn in the robust value function and not the classical one.

2. The question in practice of how to approximate the infimum operator in (1.37) (1.38) is central in RMPDs. This question will be discussed in the next paragraph. Moreover, the algorithm DRVI will be used in practice in Chapter 4 and 5 and 6.

Related work on Robust MDPs Reinforcement learning has had notable achievements but has also exhibited significant limitations, particularly when the learned policy is susceptible to deviations in the deployed environment due to perturbations, model discrepancies, or structural modifications. To address these challenges, the idea of robustness in RL algorithms has been studied. Robustness could concern uncertainty or perturbations across different Markov Decision Processes (MDPs) components, encompassing reward, state, action, and the transition kernel. Moos et al. (2022) gives a recent overview of the different work in this field.

The distributionally robust MDP (RMDP) framework has been proposed (Iyengar 2005) to enhance the robustness of RL. In addition to this work, various other research efforts, including, but not limited to, Zhang et al. (2020; 2021), Han et al. (2022), Qiaoben et al. (2021), explore robustness regarding state uncertainty. In these scenarios, the agent's policy is determined on the basis of perturbed observations generated from the state, introducing restricted noise, or

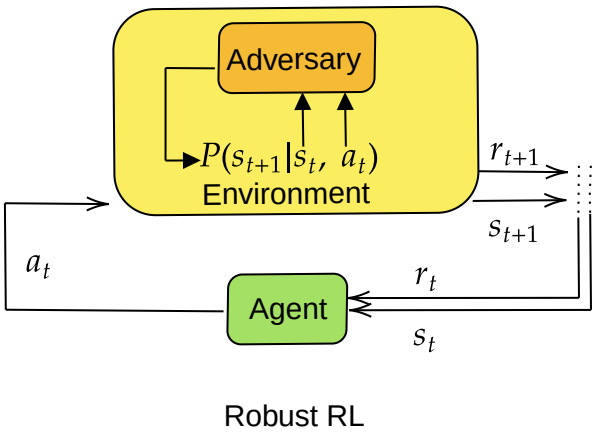


Figure 1.7: Robust Reinforcement Learning

undergoing adversarial attacks. Finally, robustness considerations extend to uncertainty in the action domain. Works such as Tessler et al. (2019), Tan et al. (2020) consider the robustness of actions, acknowledging potential distortions introduced by an adversarial agent.

Given the focus of our work, we provide a more detailed background on progress related to distributionally robust RL. The idea of distributionally robust optimization has been explored within the context of supervised learning (Rahimian and Mehrotra 2019, Gao 2020, Duchi and Namkoong 2018, Blanchet and Murthy 2019) and has also been extended to distributionally robust dynamic programming and Distributionally Robust Markov Decision Processes (DRMDPs) such as in (Iyengar 2005, Xu and Mannor 2012, Wolff et al. 2012, Kaufman and Schaefer 2013, Ho et al. 2018, Smirnova et al. 2019a, Ho et al. 2021, Goyal and Grand-Clement 2022, Derman and Mannor 2020, Tamar et al. 2014, Badrinath and Kalathil 2021). Despite the considerable attention received, both empirically and theoretically, most previous theoretical analyses in the context of RMDPs adopt an asymptotic perspective (Roy et al. 2017) or focus on planning with exact knowledge of the uncertainty set (Iyengar 2005, Xu and Mannor 2012, Tamar et al. 2014). Many works have focused on the finite-sample performance of verifiable robust Reinforcement Learning (RL) algorithms. These investigations encompass various data generation mechanisms and uncertainty set formulations over the transition kernel.

Various forms of uncertainty sets have been explored, showcasing the versatility of approaches. Divergence such as Kullback-Leibler (KL) divergence is another prevalent choice, extensively studied by Yang et al. (2021), Panaganti and Kalathil (2022b), Zhou et al. (2021), Shi and Chi (2022), Xu et al. (2023), Wang et al. (2023), Blanchet et al. (2023), who investigated the sample complexity of both model-based and model-free algorithms in simulator or offline settings. Xu et al. (2023) considered various uncertainty sets, including those associated with the Wasserstein distance. The introduction of an R-contamination uncertainty set Wang and Zou (2021), has been proposed to tackle a robust Q-learning algorithm for the online setting, with guarantees analogous to standard RL. Finally, the finite-horizon scenario has been studied by Xu et al. (2023), Dong et al. (2022) with finite-sample complexity bounds for (RMDPs) using TV and χ^2 divergence. More broadly, other related topics have been explored, such as the iteration complexity of policy-based methods (Li et al. 2022, Kumar et al. 2023), and regularization-based robust RL (Yang et al. 2023). Finally, Badrinath and Kalathil (2021) examined a general *sa*-rectangular form of the uncertainty set, proposing a model-free algorithm for the online setting with linear function approximation to address large state spaces.

1.3.3.1 From robust MDPs to practical algorithm using regularisation

In this section, the question is how to approximate or compute in a friendly way the infimum in the Robust Bellman operator in (1.37). We will discuss robustness of kernel and not of reward function that can be tackle by penalising the reward function by a certain penalty such as in Derman et al. (2021). Moreover, for simplicity we will consider *sa*-rectangular case.

$$\mathcal{U}^{\sigma}_{\|.\|}(P^0) := \left(P^0 + \mathcal{P}\right), \mathcal{P} = \times_{s\in\mathcal{S}, a\in\mathcal{A}}\mathcal{P}_{s,a}$$

where $\times$ denotes the Cartesian product. Finally we use as a notation for the infimum in Robust Bellman Operator in (1.37) $\kappa_{\mathcal{D}}(v) = \inf\left\{u^\top v : u \in \mathcal{D}\right\}$. The classical way to approximate the infimum is to compute the dual of the initial problem. First Iyengar (2005) derive practical 1-dimensional form of the dual for TV case and for χ^2 divergence. In the case of TV a quantity appears in the dual is called span semi-norm and is defined bellow.

Definition 1.3.1 (Span seminorm (Puterman 1990))**.** *Let $p \geq 1$ a real number and q be such that it satisfies the Holder's equality, i.e. $\frac{1}{p} + \frac{1}{q} = 1$. Let q-variance or span-seminorm function $\mathrm{sp}_q(.) : \mathcal{S} \to \mathbb{R}$ and q-mean function $\omega_q : \mathcal{S} \to \mathbb{R}$ be defined as*

$$\mathrm{sp}_q(v) := \min_{\omega\in\mathbb{R}} \|v - \omega\mathbf{1}\|_q \ , \quad \omega_q(v) := \arg\min_{\omega\in\mathbb{R}} \|v - \omega\mathbf{1}\|_q \ .$$

This is a measure of dispersion of the value function. Moreover, we define the upper truncated function of V by alpha as

$$[V]_\alpha := \begin{cases} \alpha, & \text{if } V(s) > \alpha \\ V(s), & \text{otherwise.} \end{cases}$$

- For TV uncertainty set with *sa*-rectangularity, we can represent $\mathcal{P}_{s,a}$ as Iyengar (2005)

$$\mathcal{P}_{s,a} = \{P_{s,a} : \mathcal{S} \to \mathbb{R} \mid \sum_{s'} P_{s,a}(s') = 0, P^0_{s,a} + P_{s,a} \geq 0, , \|P_{s,a}\|_1 \leq \sigma_{s,a}\} \tag{1.42}$$

  and we obtain for $\alpha \in \mathbb{R}^+$ (Iyengar 2005):

$$\kappa_{\mathcal{P}_{s,a}}(V) = \max_{\alpha\geq 0}\{P^0_{s,a}[V]_\alpha - \sigma_{s,a}\mathrm{sp}([V]_\alpha)_\infty\} \ ,$$

  which is a 1-dimensional optimisation problem.

- Using χ^2 divergence, for $\alpha \in \mathbb{R}^+$, the associated Robust set defined with which can be rewritten as

$$\mathcal{P}_{s,a} = \{P_{s,a} : \mathcal{S} \to \mathbb{R} \mid \sum_{s'} P_{s,a}(s') = 0, P^0_{s,a} + P_{s,a} \geq 0, \chi^2(P_{s,a} + P^0_{s,a}|P^0_{s,a}) \leq \sigma_{s,a}\} \tag{1.43}$$

  and lead to (Iyengar 2005) the dual form

$$\kappa_{\mathcal{P}_{s,a}}(V) = \max_{\alpha\geq 0}\{P^0_{s,a}[V]_\alpha - \sqrt{\sigma_{s,a}\mathbb{V}_{P^0_{s,a}}([V]_\alpha)}\}$$

  denoting classical variance as $\mathbb{V}$.

- For KL divergence, the dual can be rewritten also as for an uncertainly set

$$\mathcal{P}_{s,a} = \{P_{s,a} : \mathcal{S} \to \mathbb{R} \mid \sum_{s'} P_{s,a}(s') = 0, P^0_{s,a} + P_{s,a} \geq 0, , \mathrm{KL}(P_{s,a} + P^0_{s,a} | P^0_{s,a}) \leq \sigma_{s,a}\} \tag{1.44}$$

  like TV and χ^2, for $T \geq 0$, the dual for KL (Iyengar 2005) can be reduced as

$$\kappa_{\mathcal{P}_{s,a}}(V) = \max_{T \geq 0}\{-\sigma_{s,a}T - T \log \mathbb{E}_{P^0_{s,a}}[\exp(-V/T)]\}.$$

  Again there is a 1-dimentional optimisation problem in the dual which comes from that probabilities of the adversarial kernel is positive.

- Using L_p, for $\alpha \in \mathbb{R}^S$, the dual form is slightly more difficult to represent as using this uncertainty set

$$\mathcal{P}_{s,a} = \{P_{s,a} : \mathcal{S} \to \mathbb{R} \mid \sum_{s'} P_{s,a}(s') = 0, P^0_{s,a} + P_{s,a} \geq 0, , \|P_{s,a}\|_p \leq \sigma_{s,a}\}$$

  the infimum can be rewritten as according to Lemma in Appendix of Chapter 2.5 as

$$\begin{aligned}\kappa_{\mathcal{P}_{s,a}}(V) &= \max_{\mu^{\lambda,\omega}_{P^0_{s,a}} \in \mu^{\lambda,\omega}_{P^0_{s,a}}} \{P^0_{s,a}(V^* - \mu^{\lambda,\omega}_{P^0_{s,a}}) - \sigma_{s,a}\mathrm{sp}_q(V - \mu^{\lambda,\omega}_{P^0_{0,s,a}})\} \\ &= \max_{\alpha^{\lambda,\omega}_{P^0_{s,a}} \in \mathrm{A}^{\lambda,\omega}_{P^0_{s,a}}} P^0_{s,a}[V]_{\alpha^{\lambda,\omega}_{P^0_{s,a}}} - \sigma_{s,a}\mathrm{sp}_q([V]_{\alpha^{\lambda,\omega}_{P^0_{s,a}}}).\end{aligned}$$

  where

$$\mathrm{A}^{\lambda,\omega}_P = \{\alpha^{\lambda,\omega}_P : \alpha^{\lambda,\omega}_P(s) = \omega + \lambda|\nabla\|P\|_p|(s) : \lambda > 0, w > 0, P \in \Delta(S), \alpha^{\lambda,\omega}_P \in \left[0, \frac{1}{1-\gamma}\right]^S\} \tag{1.45}$$

$$\mathcal{M}^{\lambda,\omega}_P = \{\mu^{\lambda,\omega}_P = V - \alpha^{\lambda,\omega}_P, \lambda, \omega \in \mathbb{R}^+, P \in \Delta(S), \mu^{\lambda,\omega}_P \in \left[0, \frac{1}{1-\gamma}\right]^S\} \tag{1.46}$$

$$\tag{1.47}$$

Here α is not anymore a scalar but a vector only parameterized by only two parameters ω and λ. Moreover, the truncation for $\alpha \in \mathbb{R}^S$is defined as

$$[V]_\alpha(s) := \begin{cases}\alpha, & \text{if } V(s) > \alpha(s), \\ V(s), & \text{otherwise.}\end{cases} \tag{1.48}$$

The first remark is that there is no simple dual for KL, L_P or χ^2 divergence or for our knowledge any divergence with close form dual. When the state space is finite, it is possible to approximate easily the maximum such as in Iyengar (2005) to obtain DRVI algorithm or Q-learning based algorithm which is robust using KL divergence ball. However when the state-action space is continuous there is no simple solution to compute the dual. Thus, the question arises:

Could we derive simple/close form of the dual to compute Robust Bellman Operator easily ? Two ideas exist to get simple expression and they are all based on relaxation.

1) Use a relaxation of the problem without non negative probability constraint

Relaxation of the problem with probability of the adversary that can be possibly negative have been proposed in Kumar et al. (2022). In there algorithm, the uncertainty set is defined removing the constraint $P^0_{s,a} + P_{s,a} \geq 0$ such that

$$\mathcal{P}_{s,a} = \{P_{s,a} : \mathcal{S} \to \mathbb{R} \mid \sum_{s'} P_{s,a}(s') = 0, \|P_{s,a}\|_p \leq \sigma_{s,a}\}$$

using this formulation the dual is simple and only depend on the the span semi norm.

$$\kappa_{\mathcal{P}_{s,a}}(V) = P^0_{s,a}V - \sigma_{s,a}\mathrm{sp}_q(V) \ . \tag{1.49}$$

This formulation allows to derive practical algorithm using DRVI wiht L_p formulation (Kumar et al. 2022) or policy gradient Kumar et al. (2023). Using this relaxation, robustness is equivalent to regularisation using value function. The first work to establish the connection between regularization and Robustness in RL has been Derman et al. (2021) and in their work they do no assume any conditions on the adversary, which leads to a slightly different regularisation with uncertainty set of the form

$$\mathcal{P}_{s,a} = \{P_{s,a} : \mathcal{S} \to \mathbb{R} \mid \|P_{s,a}\|_p \leq \sigma_{s,a}\}$$

which lead to

$$\kappa_{\mathcal{P}_{s,a}}(V) = P^0_{s,a}V - \sigma_{s,a}\|V\|_q \ .$$

In fact, it is possible to do the same for example with the χ^2 divergence constraint and remove the positivity of the constraint to obtain simple risk averse mean minus standard deviation optimisation:

$$\mathcal{P}_{s,a} = \{P_{s,a} : \mathcal{S} \to \mathbb{R} \mid \sum_{s'} P_{s,a}(s') = 0, \chi^2(P_{s,a} + P^0_{s,a}|P^0_{s,a}) \leq \sigma_{s,a}\} \ ,$$

we obtain

$$\kappa_{\mathcal{P}_{s,a}}(V) = P^0_{s,a}V - \sqrt{\sigma_{s,a}\mathbb{V}_{P^0_{s,a}}(V)} \ .$$

With this formulation, we obtain simple mean minus standard deviation for the the infimum. Once the supremum cancelled in the dual of Robust Bellman Operator using a relaxation, the question of the estimation of the penalisation here is different. We will also see another way of getting close form relaxing the constrain.

2) Use alternative definition such as Soft Robust MDPs

Another way of avoiding supremum in the dual would be to use Soft Robust MDPs. In this setting introduced by Zhang et al. (2023), the distance constraint to the nominal is relaxed and is added as an objective. The uncertainty set become simply the simplex

$$\mathcal{P}_{s,a} = \{P_{s,a} : \mathcal{S} \to \mathbb{R} \mid \sum_{s'} P_{s,a}(s') = 0, P_{s,a} + P^0_{s,a} \geq 0\}$$

and the infimum is regularised with KL for example :

$$\inf_{P_{s,a} \in \mathcal{P}_{s,a}} (P_{s,a}V - \gamma\sigma^{-1}_{s,a}\,\mathrm{KL}(P_{s,a}|P^0_{s,a})) = -\gamma\sigma^{-1}_{s,a}\log \mathbb{E}_{s' \sim P^0_{s,a}} e^{-\beta V^\pi(s')} \ ,$$

which leads to Q-function of the form : $Q^\pi(s,a) = r(s,a) - \gamma\beta^{-1}\log \mathbb{E}_{s'\sim P_{s,a}} e^{-\beta V(s')}$. This idea is nice to obtain close form but the scheme does not scale yes to large or continuous action space (Zhang et al. 2023) while it is a promising avenue for Deep RL algorithm. Now from a practical point of the view to derive algorithm the question is :

How to estimate the regularisation in Deep Robust RL and does what regularisation make sense from a practical point of view?

- For L_p constraints, even using relaxation of equation (1.49), the span semi norm which is a quantity depending on all state s expect for L_1 were the semi dual span is simply the range $(\max V - \min V)/2$ need to be computed. As we consider in Deep RL in a model free, we cannot estimate this quantity easily. As the penalty is a span semi norm depending on all state it is very difficult to estimate it, even if a possible solution to approximate the penalty using samples from the replay Buffer have been proposed in (Derman, Men, Geist, and Mannor Derman et al.).
- The KL and χ^2 formulations are interesting because the penalty involve samples from the nominal kernel $P^0_{s,a}$ and not all state like in L_p. Using relaxation in χ^2 would be an alternative if we could get a good estimate of the variance of the the Q-function in the next state or an approximation using policy iteration such as in Zhang et al. (2021).
- While KL is interesting, the dual loss involves exponential term which are difficult to implement from stability point of view in Deep RL.
- A first interesting point idea is that SAC algorithm is shown to be robust to some perturbation. Indeed Eysenbach and Levine (2021) show that SAC Haarnoja et al. (2018b) is robust both in practice and in theory to some perturbation of the robust kernel.

Finally in practice, one drawback of regularisation is the the coefficient proportional to the penalty or radius of the uncertainty σ set need to be carefully chosen which is one additional hyperparameter in practice. So direct penalisation to improve robustness has two main drawbacks, estimation of the penalty and find the good uncertainty radius σ to obtain robust policy while not be too pessimistic and decrease drastically performances. The Figure 1.8 illustrates this idea where we try in Chapter 4 to estimate a penalty with coefficient α which is proportional to σ the radius of the ball. As showed in Figure 1.8, when α is too big, our algorithm cannot learn correctly as the penalisation is too strong.

We will give two alternative with easy implementation Deep Robust RL algorithm to answer the question on tow to tackle estimation problem in Robust/Regularised RL and obtain relevant penalisation in practice.

1. *Retro-engineering and design relevant penalisation* in practice, and then look at the robust set.

   In Chapter 5, this idea will be developed using Expectile statistics with lower expectile boostraping. Using this formulation allow to create implicit Robustness in Reinforcement Learning. Moreover, the hyperparmeter tuning is much easier as expectile are more interpretable than magnitude of the regularisation. We will propose a version with automatic fine tuning in Chapter 5.

2. We will try to *derive practical penalisation that easy to estimate using Distribution of returns*. One of the problem in Robust Bellman Operator is that the expectation is taken over next state s'. Using Robust Bellman Operator will lead to penalisation depending

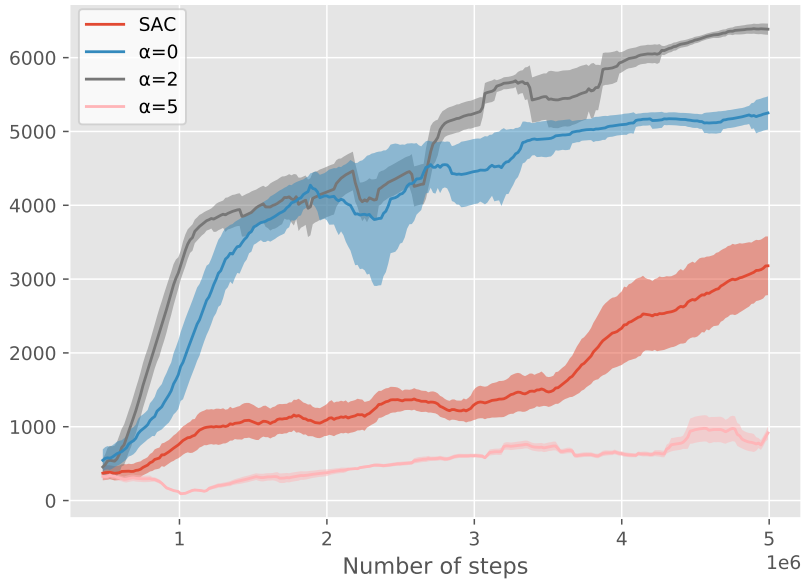


Figure 1.8: Walker-v3

on the next state, it could be expectation, variance, norm etc. However these quantities are quite difficult to estimate in practice in a model free setting as we only have access to sample from the buffer. Recall that the Robust Bellman Operator is defined as :

$$T^{\pi,P}Q(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P\left(s' \mid s,a\right) \mathbb{E}_{a' \sim \pi(\cdot|s')} \left[Q\left(s',a'\right)\right] \tag{1.50}$$

where the Q-function is simply the average return of the distribution of return given s, a. A rollout or trajectory using π from state s using initial action a is defined as the the random sequence $\tau_{s_0,a_0} = \left((s_0,a_0,r_0),(s_1,a_1,r_1),\ldots\right)$ with $s_0 = s, a_0 = a, a_t \sim \pi\left(\cdot \mid s_t\right)$, r_t the reward function and the $s_{t+1} \sim P\left(\cdot \mid s_t, a_t\right)$. The Q-function ca be rewritten as :

$$\begin{aligned} Q^{P,\pi}(s,a) &:= \mathbb{E}[Z^{P,\pi}(s,a)] && (1.51)\\ &= \mathbb{E}_{\tau_{s,a} \sim \mathbb{P}}\left[R(\tau) \mid a_t \sim \pi\left(\cdot \mid s_t\right), r_t, s_{t+1} \sim P\left(\cdot,\cdot \mid s_t,a_t\right), s_0 = s, a_0 = a\right] . && (1.52)\end{aligned}$$

Then, taking the infimum over trajectory starting from s, a called $\tau_{s,a}$, the classical Bellman Operator can be rewritten as

$$T^{\pi,P}Z(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P\left(s' \mid s,a\right) \mathbb{E}_{a' \sim \pi(\cdot|s')} \left[\mathbb{E}_{\tau_{s',a'}}[Z\left(s',a'\right)]\right] . \tag{1.53}$$

An idea would be to compute a minimum over the next trajectory against a reference trajectory denoted τ_0 that follow a given nominal kernel P^0 and π.

$$T^{\pi,P}Z(s,a) = r(s,a) + \gamma \min_{\tau_{s',a'}:\rho(\mathbb{P}(\tau_{s',a'}),\mathbb{P}(\tau_{0,s',a'}))} \sum_{s' \in \mathcal{S}} P\left(s' \mid s,a\right) \mathbb{E}_{a' \sim \pi(\cdot|s')} \left[\mathbb{E}_{\tau_{s',a'}}[Z\left(s',a'\right)]\right] \tag{1.54}$$

$$= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} P\left(s' \mid s,a\right) \mathbb{E}_{a' \sim \pi(\cdot|s')} \left[\min_{\tau_{s',a'}:\rho(\mathbb{P}(\tau_{s',a'}),\mathbb{P}(\tau_{0,s',a'}))} \mathbb{E}_{\tau_{s',a'}}[Z\left(s',a'\right)]\right] , \tag{1.55}$$

where ρ a divergence between two trajectory probability and $\mathbb{P}(\tau)$ the probability distribution of the trajectory. Finding a relevant formulation for the expectation would give a

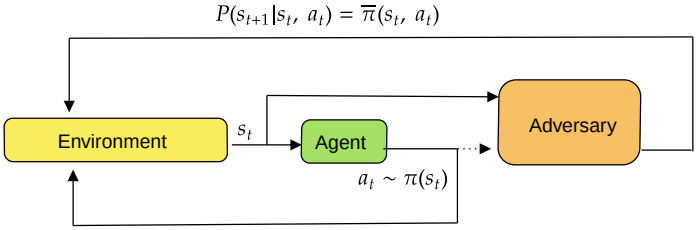


Figure 1.9: Robust RL and Zero-sum Markov Game

risk averse formulation involving the distribution of returns. For example in Chapter 4, we derive mean standard deviation error based on χ^2 divergence constraint. In practice, Distributional RL introduced by Bellemare et al. (2017) will allow use to get an approximation of the distribution of returns and gives simple estimate of the regularisation based on the distribution.

Surprisingly, most of the Robust RL algorithms are not based in these risk averse formulation that comes from Theory of Robust MDPs. One of the reason for this is that here we assume having access of sample uniquely from the nominal P^0 whereas other algorithm in Deep Robust RL, use sample from the entire uncertainty set as they modify parameters of the generative model such as in Mujoco. A interesting question is :

Can we derive algorithm using risk averse based method and combine it with sample from the entire ball and not only in the nominal?

This questions will be address in Chapter 5. In the following we will do a related work on Deep Robust Rl methods that play with sample not only from the nominal kernel P^0 but from the entire uncertainty set.

1.3.4 Deep Robust RL as a zero-sum game

Deep Robust RL as two-player games is a common approach for solving robust RL problem, representing the problem as a zero-sum two-player Markov games (Littman 1994, Tessler et al. 2019) where $\bar{\mathcal{S}}, \bar{\mathcal{A}}$ are respectively the state and action set of the adversarial player. In a zero-sum Markov game, the adversary tries to minimize the reward or maximize $-r$. Writing $\bar{\pi} : \bar{\mathcal{S}} \to \bar{\mathcal{A}} := \Delta(S)$ the policy of this adversary, the robust MDP problem turns to $\max_\pi \min_{\bar{\pi}} V^{\pi,\bar{\pi}}$, where $V^{\pi,\bar{\pi}}(s)$ is the expected sum of discounted rewards obtained when playing π (agent actions) against $\bar{\pi}$ (transition models) at each time step from s. In the specific case of robust RL as a two player-game, $\bar{\mathcal{S}} = \mathcal{S} \times \mathcal{A}$. This enables introducing the robust value iteration sequence of functions

$$V_{n+1}(s) := T^{**}V_n(s) := \max_{\pi(s)\in\Delta_A} \min_{\bar{\pi}(s,a)\in\Delta(S)} (T^{\pi,\bar{\pi}}V_n)(s) \tag{1.56}$$

where $T^{\pi,\bar{\pi}} := \mathbb{E}_{a\sim\pi(s)}[r(s,a) + \gamma \mathbb{E}_{s'\sim\bar{\pi}(s,a)} V_n(s')]$ is a zero-sum Markov game operator. These operators are also $\gamma-$contractions and converge to their respective fixed point $V^{\pi,\bar{\pi}}$ and V^{**} (Tessler et al. 2019). This two-player game formulation will be used in TC-MDPs algorithm and in the evaluation of the RRLS bechmark in Section 7 and 8.

A first family of methods define $\bar{\pi}(s_t) = P^0 + \Delta(s_t)$, where P^0 denotes the reference (nominal) transition function. Among this family, Robust Adversarial Reinforcement Learning (RARL) (Pinto et al. 2017) applies external forces at each time step t to disturb the reference dynamics. For instance, the agent controls a planar monopod robot, while the adversary applies a 2D force on the

foot. In noisy action robust MDPs (NR-MDP) (Tessler et al. 2019) the adversary shares the same action space as the agent and disturbs the agent's action $\pi(s)$. Such gradient-based approaches incur the risk of finding stationary points for π and $\bar{\pi}$ which do not correspond to saddle points of the robust MDP problem. To prevent this, Mixed-NE (Kamalaruban et al. 2020) defines mixed strategies and uses stochastic gradient Langevin dynamics. Similarly, Robustness via Adversary Populations (RAP) (Vinitsky et al. 2020) introduces a population of adversaries, compelling the agent to exhibit robustness against a diverse range of potential perturbations rather than a single one, which also helps prevent finding stationary points that are not saddle points.

Aside from this first family, State Adversarial MDPs (Zhang et al. 2020; 2021, Stanton et al. 2021) involve adversarial attacks on state observations, which implicitly define a partially observable MDP. This case aims not to address robustness to the worst-case transition function but rather against noisy, adversarial observations.

A third family of methods considers the general case of $\bar{\pi}(s_t, a_t) = P_t$ or $\bar{\pi}(s_t) = P_t$, where $P_t \in \mathcal{P}$. Minimax Multi-Agent Deep Deterministic Policy Gradient (M3DDPG) (Li et al. 2019b) is designed to enhance robustness in multi-agent reinforcement learning settings but boils down to standard robust RL in the two-agents case. Max-min TD3 (M2TD3) (Tanabe et al. 2022a) considers a policy π, defines a value function $Q(s, a, P)$ which approximates $Q^{\pi,P}(s, a) = \mathbb{E}_{s' \sim P}[r(s, a, s') + \gamma V^{\pi,P}(s')]$, updates an adversary $\bar{\pi}$ so as to minimize $Q(s, \pi(s), \bar{\pi}(s))$ by taking a gradient step with respect to $\bar{\pi}$'s parameters, and updates the policy π using a TD3 gradient update in the direction maximizing $Q(s, \pi(s), \bar{\pi}(s))$. As such, M2TD3 remains a robust value iteration method that solves the dynamic problem by alternating updates on π and $\bar{\pi}$, but since it approximates $Q^{\pi,P}$, it is also closely related to the method we introduce in the next section.

Domain randomization. Domain randomization (DR) (Tobin et al. 2017) learns a value function $V(s) = \max_\pi \mathbb{E}_{P \sim \mathcal{U}(\mathcal{P})} V^{\pi,P}(s)$ which maximizes the expected return *on average* across a fixed distribution on $\mathcal{P}$. As such, DR approaches do not optimize the worst-case performance. Nonetheless, DR has been used convincingly in applications (Mehta et al. 2020a, OpenAI et al. 2019). Similar approaches also aim to refine a base DR policy for application to a sequence of real-world cases (Lin et al. 2020, Dennis et al. 2020, Yu et al. 2018). For a more complete survey of recent works in robust RL, we refer the reader to the work of Moos et al. (2022).

We will use the idea of using sample from the entire uncertainty set. One recurrent problem with min max adversary formulation is that the adversary may lead to very bad policy. Moreover, rectangularity assumptions defined in (1.28) are not realistic in practice. So the issue is :

Can we relax classical assumptions of rectangularity to obtain more realistic transition and weaker adversary policy ?

We address this question in in the Chapter 6 where the problem of rectangularity used in theory may be not suitable in practice sometimes. To set ideas, let us consider the robust MDP of a pendulum, described by its mass and rod length. Varying this mass and rod length spans the uncertainty set of transition models. The rectangularity assumption induces that $\bar{\pi}(s_t, a_t)$ can pick a measure in $\Delta(S)$ corresponding to a mass and a length that are completely independent from the ones picked in the previous time step. While this might be a good representation in some cases, in general it yields policies that are very conservative as they optimize for adversarial configurations which might not occur in practice. We first step away from the rectangularity assumption and define a parametric robust MDP as an RMDP whose transition kernels are spanned by varying a parameter vector ψ (typically the mass and rod length in the previous example). Choosing such a vector couples together the probability measures on successor states from two distinct (s, a) and (s', a') pairs. The main current robust deep RL algorithms actually optimize policies for such parametric robust MDPs but still allow the parameter value at each time step to be picked independently of the previous time step.

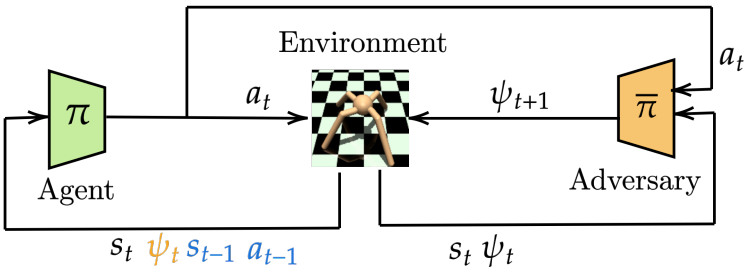


Figure 1.10: TC-RMDP training involves a temporally-constrained adversary aiming to maximize the effect of temporally-coupled perturbations. Conversely, the agent aims to optimize its performance against this time-constrained adversary. In orange, the oracle observation, and in blue the stacked observation.

Parametric MDPs. A parametric RMDP is given by the tuple $(\mathcal{S}, \mathcal{A}, \Psi, P_\psi, r)$ where the transition kernel $P_\psi(s,a) \in \Delta(S)$ is parameterized by ψ, and Ψ is the set of values ψ can take, equipped with an appropriate metric. This yields the robust value iteration update :

$$V_{n+1}(s) = \max_{\pi(s)\in\Delta_A} \min_{\psi\in\Psi} (T^\pi_\psi V_n)(s) := \max_{\pi(s)\in\Delta(A)} \min_{\psi\in\Psi} \mathbb{E}_{a\sim\pi(s)}[r(s,a) + \gamma \mathbb{E}_{s'\sim P_\psi(s,a)} V_n(s')].$$

A parametric RMDP remains a Markov game and the Bellman operator remains a contraction mapping as long as P_ψ can reach only elements in the simplex of $\Delta(S)$, where the adversary's action set is the set of parameters instead of a (possibly *sa*-rectangular) set of transition kernels. The idea to tackle this problem is to defined *Time-constrained RMDPs (TC-RMDPs).*

Time-constrained RMDPs (TC-RMDPs). We will in Chapter 6 introduce TC-RMDPs as the family of parametric RMDPs whose parameter's evolution is constrained to be Lipschitz with respect to time. More formally a TC-RMDP is given by the tuple $(S, A, \Psi, P_\psi, r, L)$, where $\|\psi_{t+1} - \psi_t\| \leq L$, that is the parameter change is bounded through time. In the previous pendulum example, this might represent the wear of the rod which might lose mass or stretch length. Similarly, and for a larger scale illustration, TC-RMDPs enable representing the possible evolutions of traffic conditions in a path planning problem through a busy town. Starting from an initial parameter value ψ_{-1}, the pessimistic value function of a policy π is non-stationary, as ψ_0 is constrained to lay at most L-far away from ψ_{-1}, ψ_1 from ψ_0, and so on.

Part I

Theory of Robust Markov Decision Processes

Chapter 2

Towards Minimax Sample Complexity of Robust RL

Contents

En un mot ma mémoire n'est pas mauvause, mais elle serait insuffisante pour faire de moi un bon joueur d'echecs. Pourquoi donc ne me fait-elle pas défaut dans un raisonnement mathématique difficile où la plupart des jouerurs d'echecs se perdraient? C'est évidemment parce qu'elle est guidée par la marche générale du raisonmment. Une démonstration mathématiques n'est pas une simple juxtaposition de syllogismes, ce sont des syllogismes placés dans un certian ordre, et l'ordre dans lequel ces éléments sont placés est beaucoup plus important que le sont les éléments eux-mêmes. Si j'ai le sentiment, l'intuition pour ainsi dire de cet ordre, de façon à apercevoir d'un coup d'oeil l'ensemble du raisonnement, je ne dois plus craindre d'oublier l'un des éléments , chacun d'eux viendra se placer de lui-même, dans le cadre qui lui est préparé, et sans que j'aie à faire auncun effort de mémoire

Henri Poincaré, Science et Methode (1908)

2.1 Introduction

Reinforcement learning (RL) (Sutton and Barto 2018), often modelled as learning and decision-making in a Markov decision process (MDP), has attracted increasing interest in recent years due to its remarkable success in practice. A major goal of RL is to find a strategy or policy, based on a collection of data samples, that can predict the expected cumulative rewards in an MDP, without direct access to a detailed description of the underlying model. However, Mannor et al. (2004) showed that the policy and the value function could sometimes be sensitive to estimation errors of the reward and transition probabilities, meaning that a very small perturbation of the reward and transition probabilities could lead to a significant change in the value function.

Robust MDPs (Iyengar 2005, Nilim and El Ghaoui 2005) (RMDPs) have been proposed to handle these problems by letting the transition probability vary in an uncertainty (or ambiguity) set. In this way, the solution of robust MDPs is less sensitive to model estimation errors with a properly chosen uncertainty set. An RMDP problem is usually formulated as a max-min problem, where the objective is to find the policy that maximizes the value function for the worst possible model that lies within an uncertainty set around a nominal model. Initially, RMPDs (Iyengar 2005, Nilim and El Ghaoui 2005) were developed because the solution of MDPs can be very sensitive to the model parameters (Zhao et al. 2019, Packer et al. 2018). However, as the solution of robust MDPs is NP-hard for general uncertainty sets Nilim and El Ghaoui (2005), the uncertainty set is usually assumed to be rectangular (meaning that it can be decomposed as a product of uncertainty sets for each state or state-action pair), which allows tractability Iyengar (2005), Ho et al. (2021). These two kinds of sets are called respectively s- and sa-rectangular sets. A fundamental difference between them is that the greedy and optimal policy in sa-rectangular robust MDPs is deterministic, as in non-robust MDPs, but can be stochastic in the s-rectangular case Wiesemann et al. (2013). Compared to sa-rectangular robust MDPs, s-rectangular robust MDPs are less restrictive but much more difficult to handle. Under this rectangularity assumption, many structural properties of MDPs remain intact Iyengar (2005) and methods such as robust value iteration, robust modified policy iteration, or partial robust policy iteration Ho et al. (2021) can be used to solve them. It is also known that the uncertainty in the reward can be easily handled, while handling uncertainty in the transition kernel is much more difficult Kumar et al. (2022), Derman et al. (2021). Finally, Deep Robust RL algorithms Pinto et al. (2017), Clavier et al. (2022), Tanabe et al. (2022b) have been proposed to tackle the problem of Robust MDPS with continuous state-action space.

In this work, we consider robust MDPs, with both sa- and s-rectangular uncertainty sets, consisting of L_p-balls centered around the nominal model P_0. We assume access to a generative model, which can sample a next state from any state-action pair from the nominal model. The question we address is to know how many samples are required to compute an ϵ-optimal policy. This classic abstraction, which allows studying the sample complexity of planning over a long horizon, is widely studied in the non-robust setting Singh and Yee (1994), Sidford et al. (2018), Azar et al. (2013a), Agarwal et al. (2020), Li et al. (2020), Kozuno et al. (2022), but much less in the robust setting (Yang et al. 2021, Panaganti and Kalathil 2022a, Shi and Chi 2022, Xu et al. 2023, Shi et al. 2023). We consider more specifically model-based robust RL. We call the generative model the same number of times for each state-action pair, to build a maximum likelihood estimate of the nominal model, and use any planning algorithm for robust MDPs (with high accuracy guarantee on the solution) on this empirical model. This setting will be discussed further later, but we insist right away that it is especially meaningful in the robust setting, as it is a good abstraction of sim2real. The research question we address is:

How many samples are required for guaranteeing an ϵ-optimal policy with high probability?

Our **first contribution** is to prove that for both s and sa-rectangular sets based on L_p-balls, the sample complexity of the proposed approach is $\tilde{\mathcal{O}}(\frac{H^4SA}{\epsilon^2})$, with $H = (1-\gamma)^{-1}$ being the horizon term. Previous works (Yang et al. 2021, Panaganti and Kalathil 2022a, Shi and Chi 2022, Xu et al. 2023) study different sets, based on the Kullback-Leibler (KL) divergence, Chi-square divergence, and total variation (TV). We have the TV in common (L_1-ball up to a normalizing factor), and, in this case, we improve these existing results by S for the sa-rectangular case, and by SA for the s-rectangular case, which is significant for large state-action spaces. On the technical side, our results build heavily upon the dual view of robust Bellman operators (Derman et al. 2021, Kumar et al. 2022). However, we deviate from this line of work by enforcing the uncertainty set to belong to the simplex. This allows ensuring that the robust operators are overly conservative while ensuring they are γ-contractions, which is important for the theoretical analysis. On the negative side, the algorithms they introduce are no longer applicable, which calls for new algorithmic design.

Our **second contribution** is to show that, if the uncertainty set is small enough, then we have a sample complexity of $\tilde{\mathcal{O}}(\frac{H^3SA}{\epsilon^2})$. This is a further improvement by H of the previous bound, and it matches the known lower bound for the non-robust case (Azar et al. 2013a). On the technical side, it again builds upon the dual view of robust Bellman operators with the deviation mentioned above.(Derman et al. 2021, Kumar et al. 2022). In addition to that, it adapts two proof techniques of the non-robust case: The total variance technique of Azar et al. (2013a) to reduce the dependency to the horizon, and the *absorbing MDP* construction of Agarwal et al. (2020) to allow for a wider range of valid ϵ.As mentioned earlier,(Derman et al. 2021, Kumar et al. 2022) algorithms are not applicable to the more realistic uncertainty sets we consider.

Our **third contribution** is an algorithm `DRVI Lp` (see Alg. 11, for Distributionally Robust Value Iteration for L_P in sarectangular case that solves exactly RMDPs in the case of valid robust transition that belongs to the simplex contrary to Kumar et al. (2022).

2.2 Related Work

The question of sample complexity when having access to a generative model has been widely studied in the non-robust setting Singh and Yee (1994), Sidford et al. (2018), Azar et al. (2013a), Agarwal et al. (2020), Li et al. (2020), Kozuno et al. (2022). Notably, Azar et al. (2013a) provide a lower-bound of this sample complexity, $\tilde{\Omega}(\frac{SAH^3}{\epsilon^2})$, and show that (tabular) model-based RL reaches this lower-bound, making it minimax optimal (up to polylog factors). This bound relies on the so-called total variance technique, that we adapt to the robust setting. However, their result is only true for small enough ϵ, in the range $(0, \sqrt{H/S})$. This was later improved to $(0, \sqrt{H})$ by Agarwal et al. (2020), thanks to a novel *absorbing MDP* construction, that we also adapt to the robust setting.

Closer to our contributions are the works that study the sample complexity in the *robust* setting Yang et al. (2021), Panaganti and Kalathil (2022a), Xu et al. (2023), Shi and Chi (2022). The study of sample complexity of specific algorithms (respectively either empirical robust value or Robust Phased Value Learning) is studied by Panaganti and Kalathil (2022a), Xu et al. (2023), while our results apply to any oracle planning (applied to the empirical model), as long as it provides a solution with enough accuracy. We consider both s- and sa-rectangular uncertainty sets, as Yang et al. (2021), while Panaganti and Kalathil (2022a), Xu et al. (2023), Shi and Chi (2022) only consider the simpler sa-rectangular sets. They all study either TV, KL or Chi-square balls, while we study L_p-balls. Shi and Chi (2022) improved the KL bound compared to Yang et al. (2021), Panaganti and Kalathil (2022a) in the sa rectangular case. The framework of Xu et al. (2023) is slightly different as they consider finite horizon which adds a factor H in all bounds. All previous results are not minimax optimal in terms of the horizon factor.

Table 2.1: Sample Complexity of TV for *s*- or *sa* rectangular with σ (see Def 2.3.2) the radius of uncertainty set (see also Tab. 9.1 in the appendix for a complete table with different norms)

	Panaganti and Kalathil (2022a)	Yang et al. (2021)	Our $\sigma \geq 0$	Our $1/(2H\gamma) > \sigma > 0$	Shi et al. (2023)
sa-rect.	$\tilde{\mathcal{O}}\left(\frac{S^2AH^4}{\epsilon^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{S^2AH^4(2+\sigma)^2}{\epsilon^2\sigma^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{SAH^4}{\epsilon^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{SAH^3}{\epsilon^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{SAH^2}{\epsilon^2\min(1/H,\sigma)}\right)$
s-rect.	$\times$	$\tilde{\mathcal{O}}\left(\frac{S^2A^2H^4(2+\tilde{\sigma})^2}{\epsilon^2\sigma^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{SAH^4}{\epsilon^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{SAH^3}{\epsilon^2}\right)$	$\times$

We rely more specifically on a simple optimization dual expression of the minimization problem over models. As such, we do not cover the KL and Chi-square cases, which do not have such a simple form even if there can also be written as simple scalar optimization problem. However, we have in common with Yang et al. (2021), Panaganti and Kalathil (2022a) the total variation case, which corresponds to a (scaled) L_1-ball. For this case, we can compare our sample complexities. Without assumption on the size of the uncertainty set, we improve the existing sample complexities by S and SA respectively (for *sa*- or *s*-rectangularity). Also, our bounds have no dependency on the size of the uncertainty set. Notice that as we consider a generic oracle planning algorithm, our bounds apply to the algorithms they consider in Panaganti and Kalathil (2022a), Xu et al. (2023). If we further assume that the uncertainty set is small enough, then we improve the bound by an additional H factor, reaching the minimax sample complexity of the non-robust case. Table 2.1 summarizes the difference in sample complexity, and we will discuss them again after stating our theorems.

Finally, the archival version of this contribution predates the concurrent work of Shi et al. (2023) that studies the sample complexity of RMDPs for TV and χ^2 divergence. In the very specific case of *sa*- rectangular for TV which in this case coincides with L_1 norm, Shi et al. (2023) retrieves our upper bound which is minimax optimal in the regime where the radius of the uncertainty set is small and improves our result in the regime where the radius of the uncertainty set is bigger than $1-\gamma$. However, our results hold more generally for the *s*-rectangular case are still state-of-the-art for *s*-rectangular case with $p \geq 1$ and for $sa-$rectangular with $p > 1$. Notice also that the proof techniques are very different, and it is an interesting research direction to know if their bound for the regime where the radius of the uncertainty set is bigger than $1-\gamma$ or their lower-bound would extend to the more general case studied here.

2.3 Preliminaries

For finite sets S and A, we write respectively S and A their cardinality. We write $\Delta(A) := \{p : A \to \mathbb{R} \mid p(a) \geq 0, \sum_{a\in A} p(a) = 1\}$ the simplex over A. For $v \in \mathbb{R}^S$ the classic L_q norm is $\|v\|_q^q = \sum_s v(s)^q$. The unitary vector of dimension S is denoted 1_S. Finally, we denote $\tilde{\mathcal{O}}$ the $\mathcal{O}$ notation up to logarithm factor.

2.3.1 Markov Decision Process

A Markov Decision Process (MDP) is defined by $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho)$ where S and A are the finite state and action spaces, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition kernel, $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the reward function, $\rho \in \Delta_{\mathcal{S}}$ is the initial distribution over states and $\gamma \in [0, 1)$ is the discount factor. A stationary policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ maps states to probability distributions over actions. We write $P_{s,a}$ the vector $P(\cdot|s, a)$. We also define P^π to be the transition matrix on state-action pairs induced by a policy π: $P^\pi_{(s,a),(s',a')} = P(s' \mid s, a)\,\pi(a' \mid s')$. Slightly abusing notations,

for $V \in \mathbb{R}^S$, we define the vector $\mathrm{Var}_P(V) \in \mathbb{R}^{\mathcal{S}\times A}$ as $\mathrm{Var}_P(V)(s,a) := \mathrm{Var}_{P(\cdot|s,a)}(V)$, so that $\mathrm{Var}_P(V) = P(V)^2 - (PV)^2$ (with the square understood component-wise). Usually, the goal is to estimate the value function defined as:

$$V^{P,r,\pi}(s) := \mathbb{E}\left[\sum_{n=0}^{\infty} \gamma^n r\left(s_n, a_n\right) \mid s_0 = s, \pi, P\right].$$

The value function $V^{P,R,\pi}$ for policy π, is the fixed point of the Bellmen operator $\mathcal{T}^{P,R,\pi}$, defined as

$$\mathcal{T}^{P,r,\pi}V(s) = \sum_a \pi(a|s)[r(s,a) + \gamma \sum_{s'} P\left(s'|s,a\right) V\left(s'\right)].$$

We also define the optimal Bellman operator: $\mathcal{T}^{P,r,*}V(s) = \max_{\pi_s \in \Delta(A)} \left(\mathcal{T}^{P,r,\pi_s}V\right)(s)$. Both optimal and classical Bellman operators are γ-contractions (Sutton and Barto 2018). This is why sequences $\{V_n^\pi \mid n \geq 0\}$, and $\{V_n^* \mid n \geq 0\}$, defined as

$$V_{n+1}^\pi := \mathcal{T}^{P,r,\pi}V_n^\pi \text{ and } \quad V_{n+1}^* := \mathcal{T}^{P,r,*}V_n^* \ ,$$

converge linearly to $V^{P,r,\pi}$ and $V^{P,r,*}$, respectively the value function following π and the optimal value function. Finally, we can define the Q-function,

$$Q^{P,r,\pi}(s,a) := \mathbb{E}\left[\sum_{n=0}^{\infty} \gamma^n r\left(s_n, a_n\right) \mid s_0 = s, a_0 = a, \pi, P\right] \ .$$

The value function and Q-function are linked with the relation $V^{P,r,\pi}(s) = \langle(\pi_s, Q^{P,R,\pi}(s)\rangle_A$. With these notations, we can define Q-functions for transition probability transition P following policy π such as

$$Q^{P,r,\pi} = r + \gamma P V^{P,r,\pi} = r + \gamma P^\pi Q^{P,r,\pi} = \left(I - \gamma P^\pi\right)^{-1} r.$$

2.3.2 Robust Markov Decision Process

Once classical MDPs defined, we can define robust (optimal) Bellman operators $\mathcal{T}_\mathcal{U}^\pi$ and $\mathcal{T}_\mathcal{U}^*$

$$\mathcal{T}_\mathcal{U}^\pi V(s) := \min_{r,P\in\mathcal{U}} \left(\mathcal{T}^{P,r,\pi}V\right)(s) \ ,$$

$$\left(\mathcal{T}_\mathcal{U}^* V\right)(s) := \max_{\pi_s \in \Delta_\mathcal{A}} \min_{r,P\in\mathcal{U}} \left(\mathcal{T}^{P,r,\pi_s}V\right)(s) \ ,$$

where P and r belong to the uncertainty set $\mathcal{U}$. The optimal robust Bellman operator $\mathcal{T}_\mathcal{U}^*$ and robust Bellman operator $\mathcal{T}_\mathcal{U}^\pi$ are γ-contraction maps for any policy π (Iyengar 2005, Thm. 3.2) if the adversarial kernel $P \in \Delta(S)$ to obtain a valid transition kernel :

$$\begin{aligned}
\|\mathcal{T}_\mathcal{U}^* v - \mathcal{T}_\mathcal{U}^* u\|_\infty &\leq \gamma \|u - v\|_\infty, \\
\|\mathcal{T}_\mathcal{U}^\pi v - \mathcal{T}_\mathcal{U}^\pi u\|_\infty &\leq \gamma \|u - v\|_\infty, \quad \forall \pi.
\end{aligned}$$

Finally, for any initial values V_0^π, V_0^*, sequences defined as $V_{n+1}^\pi := \mathcal{T}_\mathcal{U}^\pi V_n^\pi$ and $V_{n+1}^* := \mathcal{T}_\mathcal{U}^* V_n^*$ converge linearly to their respective fixed points, that is $V_n^\pi \to V_\mathcal{U}^\pi$ and $V_n^* \to V_\mathcal{U}^*$. This makes robust value iteration an attractive method for solving robust MDPs. In order to obtain tractable forms of RMDPs, one has to make assumptions about the uncertainty sets and give them a rectangularity structure Iyengar (2005). In the following, we will use an L_p norm as the distance between distributions. The *s*- and *sa*-rectangular assumptions can be defined as follows, with r_0 and P^0 being called the nominal reward and kernel.

Assumption 2.3.1. *(sa-rectangularity) We define sa-rectangular L_p-constrained uncertainty set as*

$$\mathcal{U}^{sa,\sigma}_{\|.\|_p}(P^0) := (r_0 + \mathcal{R}) \times \left(P^0 + \mathcal{P}\right), \mathcal{R} = \times_{s\in\mathcal{S},a\in\mathcal{A}}\mathcal{R}_{s,a},$$
$$\mathcal{P} = \times_{s\in\mathcal{S},a\in\mathcal{A}}\mathcal{P}_{s,a}, \mathcal{R}_{s,a} = \{r_{s,a} \in \mathbb{R} \mid |r_{s,a}| \leq \alpha_{s,a}\}$$
$$\mathcal{P}_{s,a} = \{P_{s,a} : \mathcal{S} \to \mathbb{R} \mid \sum_{s'} P_{s,a}(s') = 0, P^0_{s,a} + P_{s,a} \geq 0, \|P_{s,a}\|_p \leq \sigma_{s,a}\}$$

Assumption 2.3.2. *(s-rectangularity) We define s-rectangular L_p-constrained uncertainty set as*

$$\mathcal{U}^{s,\sigma}_{\|.\|_p} = (r_0 + \mathcal{R}) \times \left(P^0 + \mathcal{P}\right), \mathcal{P} = \times_{s\in\mathcal{S}}\mathcal{P}_s,$$
$$\mathcal{R} = \times_{s\in\mathcal{S}}\mathcal{R}_s, \quad \mathcal{R}_s = \left\{r_s : \mathcal{A} \to \mathbb{R} \mid \|r_s\|_p \leq \alpha_s\right\}$$
$$\mathcal{P}_s = \{P_s : \mathcal{S} \times \mathcal{A} \to \mathbb{R} \mid \sum_{s'} P_s(s', a) = 0, \forall a \in A, P_s(., a) + P^0_s \geq 0, \|P_s\|_p \leq \tilde{\sigma}_s\}$$

We write $\sigma = \sup_{s,a} \sigma_{s,a}$ for *sa*-rectangular assumptions or $\tilde{\sigma} = \sup_s \tilde{\sigma}_s$ for *s*-rectangular assumptions and with the same manner $\alpha = \sup_{s,a} \alpha_{s,a}$. Moreover, we write $P \in \mathcal{P}^0_{s,a}$ for $P = P^0_{s,a} + P'$ with $P' \in \mathcal{P}_{s,a}$ and $P \in \mathcal{P}^0_s$ for $P = P^{0,\pi}_s + P'$ with $P' \in \mathcal{P}_s$, $P^{0,\pi}_s(s') = \sum_a \pi(a|s)P^0_{s,a}(s') \in \mathbb{R}^S$.

In comparison to *sa*-rectangular robust MDPs, *s*-rectangular robust MDPs are less restrictive but much more difficult to deal with. Using rectangular assumptions and constraints defined with L_p-balls, it is possible to derive simple dual forms for the (optimal) robust Bellman operators for the minimization problem that involves the seminorm defined below:

Definition 2.3.1 (Span seminorm (Puterman 1990))**.** *Let q be such that it satisfies the Holder's equality, i.e. $\frac{1}{p} + \frac{1}{q} = 1$. Let q-variance or span-seminorm function $\mathrm{sp}_q(.) : \mathcal{S} \to \mathbb{R}$ and q-mean function $\omega_q : \mathcal{S} \to \mathbb{R}$ be defined as*

$$\mathrm{sp}_q(v) := \min_{\omega\in\mathbb{R}} \|v - \omega\mathbf{1}\|_q, \quad \omega_q(v) := \arg\min_{\omega\in\mathbb{R}} \|v - \omega\mathbf{1}\|_q.$$

One can think of those span-seminorms as semi-mean-centered-norms. The main problem is that these quantities represent the dispersion of a distribution around its mean, and there are no order relations for this type of object. Seminorms appear in the (non-robust) RL community for other reasons Puterman (1990), Scherrer (2013). For p =1, 2 and ∞, a closed form can be derived, corresponding to median, variance and range. This is not the case for arbitrary p but span-seminorms can be efficiently computed in practice, see Kumar et al. (2022). Once span-seminorms defined, we introduced the dual of the inner minimization problem.

Lemma 2.3.3 (Duality for *sa* rectangular case with L_p norm)**.** *For any $V \in \mathbb{R}^S, P^0_{s,a} = P^0(.|s, a) \in \mathbb{R}^S$ and $\mu \in \mathbb{R}^S$*

$$\min_{P\in\mathcal{P}^0_{s,a}} PV = \max_{\mu\geq 0} P^0_{s,a}(V - \mu) - \sigma_{s,a}\mathrm{sp}_q(V - \mu)$$

Lemma 2.3.4 (Duality for *s* rectangular case.)**.** *Consider the probability kernel $P^\pi_{0,s} = \Pi^\pi P^0_{s,a} \in \mathbb{R}^s$ with Π^π a projection matrix associated with a given policy π such that $P^{0,\pi}_s(s') = \sum_a \pi(a|s)P^0_{s,a}(s') \in \mathbb{R}^S$. For any $V \in \mathbb{R}^S$:*

$$\min_{P\in\mathcal{P}^0_s} PV = \max_{\mu\geq 0} P^{0,\pi}_s(V - \mu) - \sigma_s \|\pi_s\|_q \mathrm{sp}_q(V - \mu)$$

Proofs car be found in Appendix 2.5 ,2.3.4. These results allow computing robust value and Q-functions. Close to our work, Derman et al. (2021), Kumar et al. (2022) do not assume that robust kernel belongs to the simplex and in that sense, their formulation is a relaxation of the framework of RMPDs. Using this relaxation, closed form of robust Bellman operator can be obtained, see Th. 1 in Kumar et al. (2022). In our work, we assume a valid transition kernel in the simplex ($P_{s,a} \geq 0$ or $P_s \geq 0$ for respectively $sa-$ or $s-$ rectangular case.) that leads to dual form that has not a closed form but which is a simple scalar optimization problem. A complete discussion can be found in Appendix 1.2.

Finally, we denote robust Q function for $sa-$ and $s-$ rectangular respectively $Q^{\pi,\sigma}$ and $Q^{\pi,\tilde{\sigma}}$ and we define them from robust value function $V^{\pi,\sigma}$, $V^{\pi,\tilde{\sigma}}$ as :

$$V^{\pi,\tilde{\sigma}}(s) = \sum_a \pi(a|s) Q^{\pi,\sigma}(s,a), \quad V^{\pi,\sigma}(s) = \sum_a \pi(a|s) Q^{\pi,\sigma}(s,a)$$

Lemma 2.3.5. *For $sa-$ and $s-$ rectangular,*

$$Q^{\pi,\sigma}(s,a) = r_{Q^\pi}^{(s,a)} + \gamma P_{s,a}^0 V^{\pi,\sigma},$$
$$Q^{\pi,\tilde{\sigma}}(s,a) = r_{Q^\pi}^{s} + \gamma P_{s,a}^0 V^{\pi,\tilde{\sigma}}$$

with

$$r_{Q^\pi}^{(s,a)} = r_0(s,a) - \alpha_{s,a} + \gamma \min_{P \in \mathcal{P}_{s,a}} P V^{\pi,\sigma}$$
$$r_{Q^\pi}^{s} = r_0(s,a) - \Big(\frac{\pi_s(a)}{\|\pi_s\|_q}\Big)^{q-1} \alpha_s + \gamma \min_{P^\pi \in \mathcal{P}_s} P^\pi V^{\pi,\tilde{\sigma}}$$

Robust Q functions and dual forms of the robust Bellman operators will be central to our analysis of the sample complexity of model-based robust RL. They allow improving the bound by a factor S or SA compared to existing results (Sec. 2.4). With additional technical subtleties, adapted from the non-robust setting, and assuming the uncertainty set is small enough, they even allow improving the bound by a factor SH or SAH (Sec. 2.5).

2.3.3 Generative Model Framework

We consider the setting where we have access to a generative model, or sampler, that gives us samples $s' \sim P^0(\cdot \mid s,a)$, from the nominal model and from arbitrary state-action couples. Suppose we call our sampler N times on each state-action pair (s,a). Let $\widehat{P}$ be our empirical model, the maximum likelihood estimate of P^0,

$$\widehat{P}(s' \mid s,a) = P_{s,a}(s') = \frac{\text{count}(s',s,a)}{N},$$

where $\text{count}(s',s,a)$ represents the number of times the state-action pair (s,a) transitions to state s'. Moreover, we define $\widehat{M}$ as the empirical RMDP identical to the original M except that it uses $\widehat{P}$ instead of P^0 for the transition kernel. We denote by $\widehat{V}^\pi$ and $\widehat{Q}^\pi$ the value functions of a policy π in $\widehat{M}$, and $\widehat{\pi}^\star, \widehat{Q}^\star$ and $\widehat{V}^\star$ denote the optimal policy and its value functions in $\widehat{M}$. It is assumed that the reward function R_0 is known and deterministic and therefore exactly identical in M and $\widehat{M}$. Moreover, we write $P \in \hat{\mathcal{P}}_{s,a}$ for $P = \hat{P}_{s,a} + P'$ with $P' \in \mathcal{P}_{s,a}$ and $P \in \hat{\mathcal{P}}_s$ for $P = \hat{P}_s^\pi + P'$ with $P' \in \mathcal{P}_s$, $\hat{P}_s^\pi(s') = \sum_a \pi(a|s) \hat{P}_{s,a}(s') \in \mathbb{R}^S$.

Notice that our analysis would easily account for an estimated reward (the hard part being handling the estimated transition model). This generative model framework, when we can only sample from the nominal kernel, is classic and appears for both non-robust and robust

MDPs (Agarwal et al. 2020, Panaganti et al. 2022, Azar et al. 2013a, Xu et al. 2023). In the robust case, it is especially relevant as an abstraction of "sim-to-real", the simulator giving access to the nominal kernel for learning a robust policy to be deployed in the real world (assumed to belong to the uncertainty set).

The question of how to solve RMDPs and the related computational complexity are complementary, but different from Theorems 2.4.1and 2.5.1. Indeed, an important point that differentiates us from (Panaganti and Kalathil 2022a) is the use of a *robust optimization oracle*. In (model-based) sample complexity analysis, the goal is to determine the smallest sample size N such that a planner executed in $\widehat{M}$ yields a near-optimal policy in the RMDP M. To decouple the statistical and computational aspects of planning with respect to an approximate model $\widehat{M}$, we will use an optimization oracle that takes as input an (empirical) RMDP and returns a policy $\hat{\pi}$ that satisfies $\|\hat{Q}^* - \hat{Q}^{\hat{\pi}}\|_\infty \leq \epsilon_{\text{opt}}$. Our final bound will depend on ϵ, the error made from finite sample complexity, and ϵ_{opt} . In practice, the error ϵ_{opt} is typically decreasing at a linear speed of γ^k at the k^{th} iteration of the algorithm, as in classical MDPs because (optimal) Bellman operators are γ-contraction in both classic and robust settings when robust kernel in assuming in the simplex.

The computational cost of RMDPs is addressed by Iyengar (2005) but not in the L_p. Kumar et al. (2022) address this question, in this case, using the regularized form of robust MDPs obtained with relaxed hypothesis on the kernel (See Appendix 1.2). The conclusions of the latter are that L_p robust MDPs are computationally as easy as non-robust MDPs for regularized forms, at least for some choices of p for their relaxation. However, in their analysis, the use of γ-contraction of the Robust Bellman Operator is needed, whereas this is not always the case for sufficiently large σ. Indeed, assuming robust kernel is not anymore in the simplex, Robust Bellman Operator is not anymore a γ-contraction but an $\epsilon-$contraction for ϵ close to 1 and only for a small range of σ. (See Derman et al. (2021) Th. 5.1). We address the question of solving RMPDs in the L_p case with a valid robust kernel in Alg. 11 as it is required to obtain an ϵ_{ops} solution in our analysis.

2.4 Sample Complexity with L_p-balls

The aim of this section is to obtain an upper-bound on the sample complexity of RMDPs. This result is true for *sa*- and *s*-rectangular sets and for any L_p norm with $p \geq 1$. We remove the upperscipt σ or $\tilde{\sigma}$ as following Theorem is true both for *sa* and *s* rectangular assumptions, independently of σ or $\tilde{\sigma}$.

Theorem 2.4.1. *Assume $\delta > 0$, $\epsilon > 0$ and $\sigma > 0$. Let $\hat{\pi}$ be any ϵ_{opt} -optimal policy for $\widehat{M}$, i.e. $\|\widehat{Q}^{\widehat{\pi}} - \widehat{Q}^{\star}\|_\infty \leq \epsilon_{opt}$. With N calls to the sampler per state-action pair, such that $N \geq \frac{C\gamma^2 L''}{(1-\gamma)^4\epsilon^2}$, with $L'' = \log\left(\frac{32SAN\|1_s\|_q}{\delta(1-\gamma)}\right)$ we obtain the following guarantee for policy $\hat{\pi}$,*

$$\left\|Q^* - Q^{\hat{\pi}}\right\|_\infty \leq \epsilon + \frac{3\gamma\epsilon_{opt}}{1-\gamma}$$

with probability at least $1-\delta$, where C is an absolute constant. Finally, for $N_{total} = N|\mathcal{S}||\mathcal{A}|$ and $H = 1/(1-\gamma)$, we get an overall complexity of

$$N_{total} = \tilde{\mathcal{O}}\left(\frac{H^4 SA}{\epsilon^2}\right) .$$

2.4.1 Discussion

This result says that the policy $\hat{\pi}$ computed by the planner on the empirical RMDP $\hat{M}$ will be $(\epsilon_{\text{opt}} + \epsilon)$-optimal in the original RMDP M. As explained before, 11 planning algorithms for RMDPs that guarantee arbitrary small ϵ_{opt}, such as robust value iteration considered by Panaganti and Kalathil (2022a). It will also apply to future planners, as long as they come with a convergence guarantee. The error term ϵ is controlled by the number of samples: $N_{\text{tot}} = \tilde{\mathcal{O}}(H^4 SA\epsilon^{-2})$ calls to the generative models allow guaranteeing an error ϵ. This is a gain in terms of sample complexity of S compared to Panaganti and Kalathil (2022a), for the sa-rectangular assumption. Our bound also holds for both s- and sa-rectangular uncertainty sets. Panaganti et al. (2022) do not study the s-rectangular case, while Yang et al. (2021) do, but have a worst dependency to A in this case. Their bounds also have additional dependencies on the size of the uncertainty set, which we do not have. We recall that we do not cover the same cases, we do not analyze the KL and Chi-Square robust set, while they do not analyze the L_p robust set for $p > 1$. However, the above comparison holds for the total variation case that we have in common ($p = 1$). These bounds are clearly stated in Table 2.1. In the non-robust setting, Azar et al. (2013a) show that there exist MDPs where the sample complexity is at least $\tilde{\Omega}\left(\frac{H^3 AS}{\epsilon^2}\right)$. Section 2.5 gives a new upper-bound in H^3 which matches this lower-bound for non-robust MDPs with an extra condition on the range of σ (the uncertainty set should be small enough).

2.4.2 Sketch of Proof

This first proof is the simpler one, it relies notably on Hoeffding's concentration arguments. We provide a sketch, the full proof can be found in Appendix 2. The resulting bound is not optimal in terms of the horizon H, but it also does not impose any condition on the range of ϵ or σ, contrary to the (better) bound of Sec. 2.5. We would like to bound the supremum norm of the difference between the optimal Q-function and the one of the policy computed by the planner in the empirical RMDP, according to the true RMDP, $\|Q^* - Q^{\hat{\pi}}\|_\infty$. Using a simple decomposition and the fact that π^* is not optimal in the empirical RMDP ($\hat{Q}^{\pi^*} \leq \hat{Q}^* = \hat{Q}^{\hat{\pi}^*}$), we have that

$$Q^* - Q^{\hat{\pi}} = Q^* - \hat{Q}^* + \hat{Q}^* - \hat{Q}^{\hat{\pi}} + \hat{Q}^{\hat{\pi}} - Q^{\hat{\pi}}.$$

As $Q^* - \hat{Q}^* \leq Q^* - \hat{Q}^{\pi^*}$, a triangle inequality yields

$$\|Q^* - Q^{\hat{\pi}}\|_\infty \leq \|Q^* - \hat{Q}^{\pi^*}\|_\infty + \|\hat{Q}^* - \hat{Q}^{\hat{\pi}}\|_\infty + \|\hat{Q}^{\hat{\pi}} - Q^{\hat{\pi}}\|_\infty.$$

The second term is easy to bound, by the assumption of the planning oracle we have $\|\hat{Q}^* - \hat{Q}^{\hat{\pi}}\|_\infty \leq \epsilon_{\text{opt}}$. The two other terms are similar in nature. They compare the Q-functions of the same policy (either π^* the optimal one of the original RMDP, or $\hat{\pi}$ the output of the planning algorithm) but for different RMPDs, either the original one or the empirical one. For bounding the remaining terms, we need to introduce the following notation. For any set $\mathcal{D}$ and a vector v, let define $\kappa_{\mathcal{D}}(v) = \inf\left\{u^\top v : u \in \mathcal{D}\right\}$. This quantity corresponds to the inf form of the robust Bellman operator. The following lemma provides a data-dependent bound of the two terms of interest.

Lemma 2.4.2. *We have with $\mathcal{P}_{s,a}$ defined in Assumption 2.3.1 and $\hat{\mathcal{P}}_{s,a}$ the robust set centered around the empirical MDPs that*

$$\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\|_\infty \leq \frac{\gamma}{1-\gamma} \max_{s,a} |\kappa_{\hat{\mathcal{P}}_{s,a}}(\hat{V}^{\hat{\pi}}) - \kappa_{\mathcal{P}_{0,s,a}}(\hat{V}^{\hat{\pi}})|$$
$$\|Q^* - \hat{Q}^{\pi^*}\|_\infty \leq \frac{\gamma}{1-\gamma} \max_{s,a} |\kappa_{\hat{\mathcal{P}}_{s,a}}(V^*) - \kappa_{\mathcal{P}_{0,s,a}}(V^*)|.$$

For proving these inequalities, we rely on fundamental properties of the (robust) Bellman operator, such as γ-contraction. This lemma is written for *sa*-rectangular assumption but is also true for *s*-rectangular assumption, replacing notation of robust set $\mathcal{P}_{s,a}$ by $\mathcal{P}_s$. Now, we need to bound the resulting terms, which is done by the following lemma.

Lemma 2.4.3. *With probability at least* $1-\delta$*, we have*

$$\max_{s,a} |\kappa_{\hat{\mathcal{P}}_{s,a}}(\hat{V}^{\hat{\pi}}) - \kappa_{\mathcal{P}_{0,s,a}}(\hat{V}^{\hat{\pi}})| \leq \frac{10}{(1-\gamma)}\Big(\sqrt{\frac{L''}{2N}} + \frac{L''S^{1/q}\|1_S\|_q(p-1)}{N}\Big) + 2\epsilon_{opt}$$

with $L'' = \log\Big(\frac{32SAN\|1\|_q}{\delta(1-\gamma)}\Big)$.

Again, this also holds for *s*-rectangular sets. This inequality relies on Hoeffding's based concentration argument coupled with absorbing MDPs of Agarwal et al. (2020) and smoothness of the L_p norm. Putting everything together, we have just shown that :

$$\|Q^* - Q^{\hat{\pi}}\|_\infty \leq \frac{3\gamma\epsilon_{\text{opt}}}{1-\gamma} + \frac{20\gamma}{(1-\gamma)^2}\Big(\sqrt{\frac{L''}{2N}} + \frac{L''S^{1/q}\|1_S\|_q(p-1)}{N}\Big)\,.$$

Solving in ϵ for the second term of the right-hand side gives the stated result as the term proportional to $1/N$ is small compared to the second one for sufficiently small ϵ.

2.5 Toward minimax optimal sample complexity

Now, we provide a better bound in terms of the horizon H, reaching (up to log factors) the lower-bound in H^3 for non-robust MDPs. Recall $\sigma = \sup_{s,a}\sigma_{s,a}$ for the *sa*-rectangular assumption or $\tilde{\sigma} = \sup_s \tilde{\sigma}_s$ for the *s*-rectangular assumption. For the following result to hold, we need to assume that the uncertainty set is small enough: we will require

$$\sigma \leq \frac{1-\gamma}{2\gamma S^{1/q}} = \frac{1}{2(H-1)S^{1/q}}\,.$$

or the same condition for $\tilde{\sigma}$.The following theorem is true for both *sa*- and *s*-rectangular uncertainty sets, and for any L_p norm with $p \geq 1$.

Theorem 2.5.1. *let* $\sigma_0 \in (0, \frac{1}{2(H-1)S^{1/q}}]$*, for any* $\kappa > 0$ *and any* $\epsilon_0 \leq \kappa\sqrt{H}$ *it exists a* $C_{\sigma_0,\epsilon_0} > 0$ *independent of* H *such that for any* $\sigma \in (0,\sigma_0)$ *and any* $\epsilon \in (0,\epsilon_0)$*, whenever* N *the number of calls to the sampler per state-action pair satisfies* $N \geq C_{\sigma_0,\epsilon_0}\frac{L\gamma^2H^3}{\epsilon^2}$ *where* $L = \log(8|\mathcal{S}||\mathcal{A}|/((1-\gamma)\delta))$*, it holds that if* $\widehat{\pi}$ *is any* ϵ_{opt} *-optimal policy for* $\widehat{M}$*, that is when* $\|\widehat{Q}^{\widehat{\pi}} - \widehat{Q}^\star\|_\infty \leq \epsilon_{opt}$*, then*

$$\left\|Q^* - Q^{\hat{\pi}}\right\|_\infty \leq \epsilon + \frac{8\epsilon_{opt}}{1-\gamma}$$

with probability at least $1-\delta$*.So* $N_{total} = N|\mathcal{S}||\mathcal{A}|$ *as an overall sample complexity*

$$\tilde{\mathcal{O}}\left(\frac{H^3SA}{\epsilon^2}\right)$$

for any $\epsilon < \epsilon_0$*. The result is true with* $\tilde{\sigma}$ *replacing* σ *for the s-rectangular case.*

2.5.1 Discussion

The constants of Theorem 2.5.1 are explicitly given in Appendix 3. For instance, for $\sigma_0 = \frac{1}{8(H-1)}$ and $\epsilon_0 = \sqrt{16H}$, we have $C = 1024$, other choices being possible. Recall that in the non-robust case, the lower-bound is $\tilde{\Omega}\left(\frac{H^3SA}{\epsilon^2}\right)$ Azar et al. (2013a). Our theorem states that any model-based robust RL approach, in the generative model setting, with an accurate enough planner applied to the empirical RMDP, reaches this lower bound, up to log terms. As far as we know, it is the first time that one shows that solving an RMDP in this setting does not require more samples than solving a non-robust MDP, provided that the uncertainty set is small enough. Our bound on ϵ is similar to the one of Agarwal et al. (2020) in the robust case with their range $[0, \sqrt{H})$, we differ only by giving more flexibility in the choice of the constant C. The best range of ϵ for non-robust MDPs is $(0, H)$ (Li et al. 2020), we let its extension to the robust case for future work. So far, we discussed the lower-bound for the non-robust case, that we reach. Indeed, non-robust MDPs can be considered as a special case of MDPs with $\sigma = 0$. As far as we know, the only robust-specific lower-bounds on the sample complexity have been proposed by Yang et al. (2021). They propose two lower-bounds accounting for the size of the uncertainty set, one for the Chi-square case, and one for the total variation case, which coincide with our L_p framework for $p = 1$ This bound is

$$\tilde{\Omega}\left(\frac{SA(1-\gamma)}{\varepsilon^2}\min\left\{\frac{1}{(1-\gamma)^4}, \frac{1}{\sigma^4}\right\}\right).$$

This lower bound has two cases, depending on the size of the uncertainty set. If $\sigma \leq (1-\gamma) = 1/H$, we retrieve the non-robust lower bound $\tilde{\Omega}\left(\frac{SAH^3}{\varepsilon^2}\right)$. Therefore, for a L_1-ball, our upper-bound matches the lower-bound, and we have proved that model-based robust RL in the generative model setting is minimax optimal for any accurate enough planner. Their condition for this bound, $\sigma \leq 1/H$, is close to our condition, $\sigma < 1/(4(H-1)$. This suggests that our condition on σ is not just a proof artifact. In the second case, if $\sigma > 1-\gamma$, the lower-bound is $\tilde{\Omega}\left(\frac{SA|(1-\gamma)}{\varepsilon^2\sigma^4}\right)$. In this case, our theorem does not hold, and we only currently get a bound in H^4 (see Sec. 2.4), which doesn't match this lower-bound.

In the case of TV, we know from posterior work Shi et al. (2023) that it is possible to get a tighter bound in the regime $\sigma > 1-\gamma$ but in the case of L_P norm, it is still an open question. In the case where σ is too large, the question arises whether RMDPs are useful as long as there is little to control when the transition kernel can be too arbitrary.

To sum up, to the best of our knowledge, with a small enough uncertainty set, our work delivers the first-ever minimax-optimal guarantee for RMDPs according to the non-robust lower-bound for L_p-balls, and the first ever minimax-optimal guarantee according to the robust lower-bound for the total variation case for a sufficiently small radius of the uncertainty set, which has been later on the larger set of σ by Shi et al. (2023). ‘

2.5.2 Sketch of proof

The full proof is provided in Appendix 3. As in Sec. 2.4.2, we start from the inequality

$$\|Q^* - Q^{\hat{\pi}}\|_\infty \leq \|Q^* - \hat{Q}^{\pi^*}\|_\infty + \|\hat{Q}^* - \hat{Q}^{\hat{\pi}}\|_\infty + \|\hat{Q}^{\hat{\pi}} - Q^{\hat{\pi}}\|_\infty,$$

where the second term of the right-hand side can again be readily bounded, $\|\hat{Q}^* - \hat{Q}^{\hat{\pi}}\|_\infty \leq \epsilon_{\text{opt}}$. To bound the remaining two terms, if we want to obtain a tighter final bound, the contracting property of the robust Bellman operator will not be enough, we need a finer analysis. To achieve this, we rely on the total variance technique introduced by Azar et al. (2013a) for the non-robust case, combined with the *absorbing MDP* construction of Agarwal et al. (2020), also for the

non-robust case, which allows improving the range of valid ϵ. The key underlying idea is to rely on a Bernstein concentration inequality rather than a Hoeffding one, therefore considering the variance of the random variable rather than its range, tightening the bound. Working with a Bernstein inequality will require controlling the variance of the return. A key result was provided by Azar et al. (2013a), that we extend to the robust setting,

$$\left\| \left(I - \gamma P^{0,\pi}\right)^{-1} \sqrt{\mathrm{Var}_{P^0}\left(V^\pi\right)} \right\|_\infty \leq \sqrt{\frac{2}{(1-\gamma)^3}}. \tag{2.1}$$

Naively bounding the left-hand side would provide a bound in H^2, while this (non-obvious) bound in $\sqrt{H^3}$ is crucial for obtaining on overall dependency in H^3 in the end. Now, we come back to the terms $\|Q^* - \hat{Q}^{\hat{\pi}^*}\|_\infty$ and $\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\|_\infty$ that we have to bound. This bound should involve a term proportional to $(I - \gamma P^{0,\pi})^{-1}$ to leverage later Eq. (2.1). The following lemma is inspired by Agarwal et al. (2020), and its proof relies crucially on having a simple dual of robust Bellman operator.

Lemma 2.5.2.

$$\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\|_\infty \leq \gamma \|(I - \gamma P^{0,\hat{\pi}})^{-1}(P^0 - \widehat{P})\hat{V}^{\hat{\pi}}\|_\infty + \frac{2\gamma\sigma S^{1/q}}{1-\gamma}\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\|_\infty.$$

We see that the term σ appears in the bound. This comes from the need to control the difference in penalization between seminorms of value functions, from a technical viewpoint. Indeed, the terms $\frac{2\gamma\sigma}{1-\gamma}\|Q^\pi - \hat{Q}^\pi\|_\infty$ (with π being either $\hat{\pi}$ or π^*) are not present in the non-robust version of the bound, and are one of the main differences from the derivation of Agarwal et al. (2020). The first term of the right-hand side of each bound $\|(I - \gamma P^{0,\pi})^{-1}(P_0 - \widehat{P})\hat{V}^\pi\|_\infty$ (with π being either $\hat{\pi}$ or π^*, again) will be upper-bounded using a Bernstein argument, leveraging also Eq. (2.1). The resulting lemma is the following.

Lemma 2.5.3. *With probability at least $1-\delta$, we have*

$$\left\|Q^{\widehat{\pi}} - \widehat{Q}^{\widehat{\pi}}\right\|_\infty < (C_N + C_\sigma)\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\|_\infty + 4\gamma\sqrt{\frac{L}{N(1-\gamma)^3}} + \frac{\gamma\Delta'_{\delta,N}}{1-\gamma} + \frac{\gamma\epsilon_{\text{opt}}}{1-\gamma}\left(2 + \sqrt{\frac{8L}{N}}\right),$$

with $C_\sigma = \frac{2\gamma\sigma S^{1/q}}{1-\gamma}$ and $C_N = \frac{\gamma}{1-\gamma}\sqrt{\frac{8L}{N}}$ and where $\Delta'_{\delta,N} = \sqrt{\frac{cL}{N}} + \frac{cL}{(1-\gamma)N}$
with $L = \log(8SA/((1-\gamma)\delta))$.

For this result to be exploitable, we have to ensure that $C_N + C_\sigma < 1$, which leads to $\sigma \leq \frac{1-\gamma}{2\gamma S^{1/q}}$, and then $C_N + C_\sigma < 1$ leads to a constraint on N in Theorem 2.5.1. Eventually, injecting the result of this last lemma in the initial bound, keeping the dominant term in $1/\sqrt{N}$ and solving for ϵ provides the stated result, cf Appendix 3.

2.6 Conclusion

In this paper, we have studied the question of the sample complexity of model-based robust reinforcement learning. To decouple this from the problem of exploration, we have considered the classic (in non-robust RL) generative model setting, where a sampler can provide next-state samples from the nominal kernel and from arbitrary state-action couples. We focused our study more specifically on *sa*- and *s*-rectangular uncertainty sets corresponding to L_p-balls around the nominal.

Without any restriction on the size of uncertainty set (σ), we have shown that the sample complexity of the studied general setting is $\tilde{\mathcal{O}}(\frac{SAH^4}{\epsilon^2})$, already significantly improving existing results (Yang et al. 2021, Panaganti and Kalathil 2022a). Our bound holds for both the *sa*- and *s*-rectangular cases, and improves existing results (for the total variation) by respectively S and SA. By assuming a small enough uncertainty set, and for a small enough ϵ, we further improved this bound to $\tilde{\mathcal{O}}(\frac{SAH^3}{\epsilon^2})$, adapting proof techniques from the non-robust case (Azar et al. 2013a, Agarwal et al. 2020). This is a significant improvement. Our bound again holds for both the *sa*- and *s*- rectangular cases, it matches the lower-bound for the non-robust case Azar et al. (2013a), and it matches the total variation lower-bound for the robust case when the uncertainty set is small enough (Yang et al. 2021). We think this is an important step towards minimax optimal robust reinforcement learning.

There are a number of natural perspectives, such as knowing if we could extend our results to other kinds of uncertainty sets, or to extend our last bound to larger uncertainty sets (despite the fact that if the dynamics are too unpredictable, there may be little left to be controlled). Our results build heavily on the simple dual form of the robust Bellman operator, which prevents us from considering, for the moment, uncertainty sets based on the KL or Chi-square divergence. Beyond their theoretical advantages, these simple dual forms also provide practical and computationally efficient planning algorithms. Therefore, another interesting research direction would be to know if one could derive additional useful uncertainty sets relying primarily on the regularization viewpoint. In the next Chapter, we will refine our result in term of upper bound while providing also lower bound to better understand the question of sample complexity in Robust MDPs.

Chapter 3

Near-Optimal Distributionally Robust Reinforcement Learning with General L_p Norms

Contents

3.1 Introduction

Reinforcement learning (RL) (Sutton 1988) is a popular paradigm in machine learning, particularly noted for its success in practical applications. The RL framework, usually modeled within the context of a Markov decision process (MDP), focuses on learning effective decision-making strategies based on interactions with an environment. However, the work of Mannor et al. (2004), among others, has highlighted a vulnerability in RL strategies, revealing the sensitivity to estimation errors in the reward and transition probabilities. A specific example of this is when, because of a sim-to-real gap, policies learned in idealized environments catastrophically fail when deployed in settings with slight changes or adversarial perturbations (Klopp et al. 2017, Mahmood et al. 2018).

To address this issue, robust MDPs (RMDPs), proposed by Iyengar (2005) and Nilim and El Ghaoui (2005), have attracted considerable attention. RMDPs are formulated as max-min problems, seeking policies that are resilient to model estimation errors within a specified uncertainty set. Despite the robustness benefits, solving RMDPs is NP-hard for general uncertainty sets (Nilim and El Ghaoui 2005). To overcome this challenge, the assumption of rectangularity is often adopted, with uncertainty sets structured as products of independent subsets for each state or state-action pair, denoted as *s*-rectangular or *sa*-rectangular assumptions (see Definitions 3.4 and 3.5). These assumptions facilitate the use of methods such as robust value iteration and robust policy iteration, preserving many structural properties of MDPs (Ho et al. 2021). The *s*-rectangular sets, though less restrictive, pose greater challenges, while the *sa*-rectangular sets allow for deterministic optimal policies akin to non-robust MDPs (Wiesemann et al. 2013). Note

that, while uncertainty in the reward can be easily handled, dealing with uncertainty in the transition kernel is much more difficult (Kumar et al. 2022, Derman et al. 2021).

The question of sample efficiency is central in RL problems ranging from practice to theory. Although minimax rates are achieved in (Azar et al. 2013b, Li et al. 2023) in the context of classical MDPs, this goal remains open, in general, in the context of RMDPs. Specifically, there exists prior work studying the sample complexity of distributionally robust RL for a few specific divergences such as total variation (TV), χ^2, KL, and Wasserstein (see a further discussion in Appendix 4) (Yang et al. 2022, Zhou et al. 2021, Panaganti and Kalathil 2022b), while such results remain unclear for more general classes of L_p norms defined in 3.2.1.To this point, to the best of our knowledge, the results of sample complexity that achieve minimax optimality for the full range of uncertainty level are limited to only one case — TV distance (Shi et al. 2023).

In this work, we focus on understanding the sample complexity of RMDPs with a general smooth L_p that will be defined in Def. 3.2.1. This generalization is appealing for both practice and theory. In practice, numerous applications are based on optimizations or learning approaches that involve general norms beyond those that have already been studied. Additionally, optimizing norm weighted ambiguity sets for Robust MDPs has been proposed in the context of RMDPs in Russel et al. (2019), which justifies our formulation. Theoretically, prior work has characterized the sample complexity of RMDPs for some specific norms have suggested intriguing insights about the statistical implications of distributional robustness in RL. It is interesting to further understand the statistical cost of robust RL in more general scenarios.One area of focus is the contrast between the sample efficiency of solving distributionally robust RL and solving standard RL. In particular, for the specific case of TV distance, Shi et al. (2023) shows that the sample complexity for solving robust RL is at least the same as and sometimes (when the uncertainty level is relatively large) could be smaller than that of standard RL. This motivates the following open question:

Is distributionally robust RL more sample efficient than standard RL for norms defined in Def. (3.2.1) ?

A second question is about the comparisons between the sample complexity of solving s-rectangular RMDPs and that of solving sa-rectangular RMDPs. Note that s-rectangular RMDPs have more complicated optimization formulations with additional variables (uncertainty levels for each action) to optimize. This leads to a richer class of optimal policy candidates—stochastic policies in s-rectangular cases, in contrast to the class of deterministic policies for sa-rectangular cases. In addition, existing sample complexity upper bounds for solving s-rectangular RMDPs are larger than that for solving sa-rectangularity (Yang et al. 2022) for the investigated cases. This motivates the curious question:

Does solving s-rectangular RMDPs require more samples than solving sa-rectangular RMDPs with general smooth L_p norms defined in Def. 3.2.1?

Main contributions. In this paper, we address each of the two questions discussed above. In particular, we provide the first sample complexity analysis for RMDPs with general L_p norms defined in 3.2.1 under both the s- and sa-rectangularity conditions. For convenience, we present a detailed comparison between the existing state-of-the-art and our results in Table 3.1 for quick reference and discuss the contributions and their implications below.

• Considering the first question, we illustrate our results in both sa- and s-rectangular case in Figure 3.1. In the case of sa-rectangularity, we derive a sample complexity upper bound for RMDPs using general smooth L_p norms (cf. Theorem 3.4.1) in the order of

$$\widetilde{O}\left(\frac{SA}{(1-\gamma)^2\max\{1-\gamma, C_g\sigma\}\varepsilon^2}\right),$$

with $C_g > 0$ a positive constant related to the geometry of the norm defined in 3.2.1. For classical

Result type	Reference	Distance	*sa*-rectangularity		*s*-rectangularity	
			$0 < \sigma \lesssim 1-\gamma$	$1-\gamma \lesssim \sigma < \sigma_{\max}$	$0 < \tilde{\sigma} \lesssim 1-\gamma$	$1-\gamma \lesssim \tilde{\sigma} < \tilde{\sigma}_{\max}$
Upper bound	Yang et al. (2021)	TV	$\frac{S^2A(2+\sigma)^2}{\sigma^2(1-\gamma)^4\varepsilon^2}$	$\frac{S^2A(2+\sigma)^2}{\sigma^2(1-\gamma)^4\varepsilon^2}$	$\frac{S^2A^2(2+\tilde{\sigma})^2}{\tilde{\sigma}^2(1-\gamma)^4\varepsilon^2}$	$\frac{S^2A^2(2+\tilde{\sigma})^2}{\tilde{\sigma}^2(1-\gamma)^4\varepsilon^2}$
	Panaganti and Kalathil (2022b)	TV	$\frac{S^2A}{(1-\gamma)^4\varepsilon^2}$	$\frac{S^2A}{(1-\gamma)^4\varepsilon^2}$	×	×
	Shi et al. (2023)	TV	$\frac{SA}{(1-\gamma)^3\varepsilon^2}$	$\frac{SA}{\sigma(1-\gamma)^2\varepsilon^2}$	×	×
	Clavier et al. (2023)	L_p	$\frac{SA}{(1-\gamma)^3\varepsilon^2}$	$\frac{SA}{(1-\gamma)^4\varepsilon^2}$	$\frac{SA}{(1-\gamma)^3\varepsilon^2}$	$\frac{SA}{(1-\gamma)^4\varepsilon^2}$
	This paper	L_p	$\frac{SA}{(1-\gamma)^3\varepsilon^2}$	$\frac{SA}{\sigma(1-\gamma)^2\varepsilon^2}$	$\frac{SA}{(1-\gamma)^3\varepsilon^2}$	$\frac{SA}{(1-\gamma)^2\tilde{\sigma}\min_s\|\pi_s\|_*\varepsilon^2}$
	This paper	General L_p [3.2.1]	$\frac{SA}{(1-\gamma)^3\varepsilon^2}$	$\frac{SA}{\sigma(1-\gamma)^2\varepsilon^2}$	$\frac{SA}{(1-\gamma)^3\varepsilon^2}$	$\frac{SA}{(1-\gamma)^2\tilde{\sigma}C_g\min_s\|\pi_s\|_*\varepsilon^2}$
Lower bound	Yang et al. (2021)	TV	$\frac{SA}{(1-\gamma)^3\varepsilon^2}$	$\frac{SA(1-\gamma)}{\sigma^4\varepsilon^2}$	×	×
	Shi et al. (2023)	TV	$\frac{SA}{(1-\gamma)^3\varepsilon^2}$	$\frac{SA}{\sigma(1-\gamma)^2\varepsilon^2}$	×	×
	This paper	L_p	$\frac{SA}{(1-\gamma)^3\varepsilon^2}$	$\frac{SA}{\sigma(1-\gamma)^2\varepsilon^2}$	×	×
	This paper	L_∞	$\frac{SA}{(1-\gamma)^3\varepsilon^2}$	$\frac{SA}{\sigma(1-\gamma)^2\varepsilon^2}$	$\frac{SA}{(1-\gamma)^3\varepsilon^2}$	$\frac{SA}{\tilde{\sigma}(1-\gamma)^2\varepsilon^2}$

Table 3.1: Comparisons with prior results (up to log terms) regarding finding an ε-optimal policy for the distributionally RMDP, where σ is the radius of the uncertainty set and $\sigma_{\max}$ defined in Theorem 3.4.1.

L_P norms, $C_g \geq 1$ so we can directly relax this constant to 1 to obtain the result in table 3.1. In addition, we provide a matching minimax lower bound (cf. Theorem 3.4.2) that confirms the near-optimality of the upper bound for almost full range of the uncertainty level. Our results match the near-optimal sample complexity derived in Shi et al. (2023) for the specific case using TV distance, while holding for broader cases using general L_p norms. The results rely on a new dual optimization form for *sa*-rectangular RMDPs and reveal the relationship between the sample complexity and this new dual form — the infinite span seminorm (controlled in Lemma 7.1), which may be of independent interest.

In the case of *s*-rectangularity, we provide a sample complexity upper bound for solving RMDPs with general smooth L_p norms in the order of

$$\widetilde{O}\left(\frac{SA}{(1-\gamma)^2\max\{1-\gamma, C_g\min_s\|\pi_s\|_*\tilde{\sigma}\}\varepsilon^2}\right).$$

This result improves the prior art $\widetilde{O}\left(\frac{SA}{(1-\gamma)^4\varepsilon^2}\right)$ in Clavier et al. (2023) for classical L_p when $\tilde{\sigma} \lesssim 1-\gamma$ — by at least a factor of $O\left(\frac{1}{1-\gamma}\right)$. Furthermore, we present a lower bound for a representative case with L_∞ norm, which corroborates the tightness of the upper bound. To the best of our knowledge, this is the first lower bound for solving RMDPs with *s*-rectangularity.

• Considering the second question, as illustrated in Figure 3.1, our results highlight that robust RL is at least the same as and sometimes can be more sample-efficient to solve than standard RL for general smooth L_p norms in 3.2.1. This insight is of significant practical importance and serves to provide crucial motivation for the use and study of distributionally robustness in RL. Notably, robust RL does not only reduce the vulnerability of RL policy to estimation errors and sim-to-real gaps, but also leads to better data efficiency. In terms of comparing the statistical implications of *sa*- and *s*- rectangularity, our results show that solving *s*-rectangular RMDPs is not harder than solving *sa*-rectangular RMDPs in terms of sample requirement (See Theorem 3.4.3 and Figure 3.2, Right).

• We highlight the technical contributions as below. For the upper bounds, regarding optimization contribution, we derive new dual optimization problem forms for both *sa*− and *s*− rectangular cases (Lemma 6.3 and 6.4), which is the foundation of the covering number argument in finite-

sample analysis. From a statistical point of view, a new concentration lemma (See Lemma 7.4 for dual forms and two new lemmas to obtain sample complexity lower than classical RL, controlling the infinite span semi norm of the value function, both for $sa-$ and $s-$ rectangular case are derived (See Lemmas 7.1 and 7.2). For the lower bound, the technical contributions are mainly in s-rectangular cases, which involves entire new challenges compared to sa-rectangularity case: the optimal policies can be stochastic and hard to be characterized as a closed form, compared to the deterministic one in sa-rectangular cases. Therefore, we construct new hard instances for s-rectangular cases that is distinct from those used in sa-rectangular cases or standard RL.

3.2 Problem Formulation: Robust Markov Decision Processes

In this section, we formulate distributionally robust Markov decision processes (RMDPs) in the discounted infinite-horizon setting, introduce the sampling mechanism, and describe our goal.

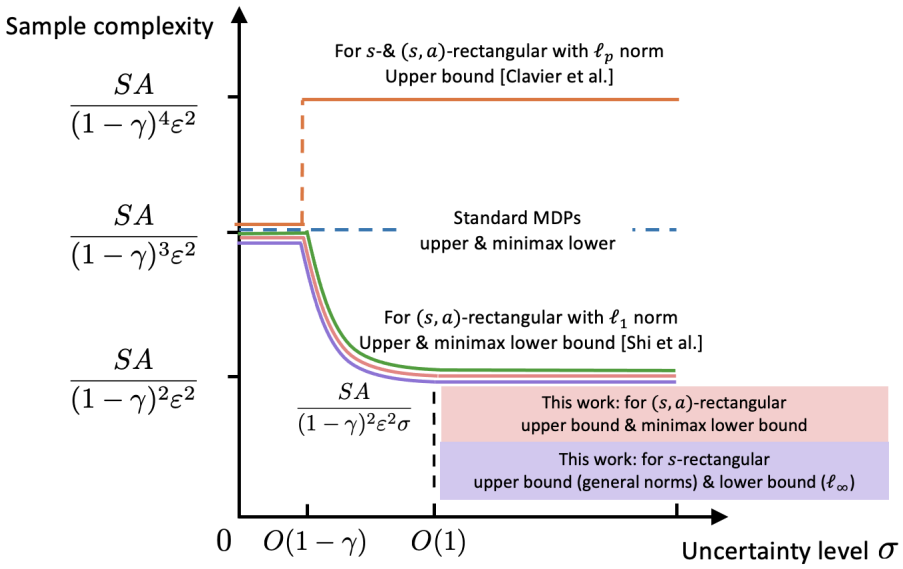


Figure 3.1: Left: Sample complexity results for RMDPs with sa- and s-rectangularity with L_p with comparisons to prior arts (Shi et al. 2023) (for L_1 norm, or called total variation distance) and (Clavier et al. 2023)

Standard Markov decision processes (MDPs). A discounted infinite-horizon MDP is represented by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \gamma, P, r)$, where $\mathcal{S} = \{1, \cdots, S\}$ and $\mathcal{A} = \{1, \cdots, A\}$ are the finite state and action spaces, respectively, $\gamma \in [0, 1)$ is the discounted factor, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ denotes the probability transition kernel, and $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the immediate reward function, which is assumed to be deterministic. Moreover, we assume that the reward function is bounded in $(0, 1)$ without loss of generality of the results due to the variance reward invariance. Finally we denote 1_A or 1_S the unitary vector of respectively dimension A or S. Moreover, e_s is the standard unitary vector supported on s. The policy we are looking for is denoted by $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, which specifies the probability of action selection over the action space in any state. Note that if the policy is deterministic in the sa-rectangular case, we overload the notation and refer to $\pi(s)$ as the action selected by the policy π in state s. Finally, to characterize the cumulative reward,

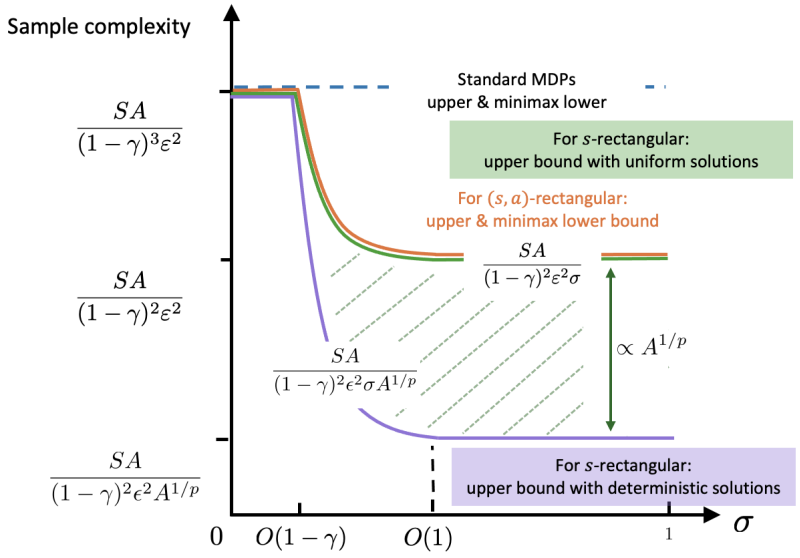


Figure 3.2: The data and instance-dependent sample complexity upper bound of solving s-rectangular dependency RMDPs with L_P norms.

the value function $V^{\pi,P}$ for any policy π under the transition kernel P is defined by $\forall s \in \mathcal{S}$

$$V^{\pi,P}(s) := \mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\,\middle|\, s_0 = s\right]. \tag{3.1}$$

The expectation is taken over the randomness of the trajectory $\{s_t,a_t\}_{t=0}^{\infty}$ generated by executing the policy π under the transition kernel P, such that $a_t \sim \pi(\cdot\,|\,s_t)$ and $s_{t+1} \sim P(\cdot\,|\,s_t,a_t)$ for all $t \geq 0$. In the same way, the Q function $Q^{\pi,P}$ associated with any policy π under the transition kernel P is defined using expectation taken over the randomness of the trajectory under policy π as

$$Q^{\pi,P}(s,a) := \mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\,\middle|\, s_0,a_0 = s,a\right]. \tag{3.2}$$

Distributionally robust MDPs. We consider distributionally robust MDPs (RMDPs) in the discounted infinite-horizon setting, denoted by $\mathcal{M}_{\mathsf{rob}} = \{\mathcal{S},\mathcal{A},\gamma,\mathcal{U}^{\sigma}_{\|.\|}(P^0),r\}$, where $\mathcal{S},\mathcal{A},\gamma,r$ are the same sets and parameters as in standard MDPs. The main difference compared to standard MDPs is that instead of assuming a fixed transition kernel P, it allows the transition kernel to be arbitrarily chosen from a prescribed uncertainty set $\mathcal{U}^{\sigma}_{\|.\|}(P^0)$ centered around a *nominal* kernel $P^0 : \mathcal{S}\times\mathcal{A} \to \Delta(\mathcal{S})$, where the uncertainty set is specified using some called L_p smooth norm denoted $\|.\|$ defined in of radius $\sigma > 0$ defined in 3.2.1.

Definition 3.2.1 (General smooth L_p norms and dual norms)**.** *A norm $\|\cdot\|$ is said to be a general smooth L_p norm if*

- *for all $x \in \mathbb{R}^n$, $\|x\| = \|x\|_{p,w} = (\sum_{k=1}^n w_k(|x_k|)^p)^{1/p}$, where $w \in \mathbb{R}^n_+$, is an arbitrary positive vector,*
- *it is twice continuously differentiable Rudin et al. (1964) with the supremum of the Hessian Matrix over the simple $C_S = \sup_{x\in\Delta_S}\|\nabla^2\|x\|\|_2$, where $\|\|_2$ here is the spectral norm*

Finally, we denote the dual norm of $\|\cdot\|$ *as* $\|\cdot\|_*$ *s.t.* $\|y\|_* = \max_x x^T y : \|x\| \leq 1$. *Moreover, for any metric* $\|.\|$, *we define* C_g *as* $C_g = 1/\min_s \|e_s\|$ *where* $e_s \in \mathbb{R}^S$ *is the standard basis of supported in* s.

Note the quantity C_S exists as the Hessian is continuous for C^2 functional and the simplex is a compact set, so by Extreme Value Theorem Rudin et al. (1964), C_S is finite. Moreover, to give an example, considering L_p, $p \geq 2$, norms, C_S is bounded by $(p-1)S^{1/q}$. (See (A.203)) This definition is general and includes $L_p, p \geq 2$, all rescaled and weighted norms. Moreover, we could extend our result to a larger set than the one of the norms defined in Def. 3.2.1, this is why a complete discussion about the set of norms can be found in Appendix 5. However, it does not include divergences such as KL and χ^2. Not that the case of TV which is not C^2 smooth is treated independently with different arguments in the proof but has the same sample complexity. In particular, given the nominal transition kernel P^0 and some uncertainty level σ, the uncertainty set—with arbitrary smooth L_p norm metric $\|\ \| : \mathbb{R}^{\mathcal{S}} \times \to \mathbb{R}^+$ in *sa* rectangular case or from $\mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ in the *s*-rectangular case, is specified as $\mathcal{U}^{\sigma}_{\|.\|}(P^0) \coloneqq \otimes_{s,a} \mathcal{U}^{\mathsf{sa},\sigma}_{\|.\|}(P^0_{s,a})$

$$\mathcal{U}^{\mathsf{sa},\sigma}_{\|.\|}(P^0_{s,a}) \coloneqq \left\{P_{s,a} \in \Delta(\mathcal{S}) : \left\|P_{s,a} - P^0_{s,a}\right\| \leq \sigma\right\}, \tag{3.3}$$

$$P_{s,a} \coloneqq P(\cdot \,|\, s,a) \in \mathbb{R}^{1\times S}, P^0_{s,a} \coloneqq P^0(\cdot \,|\, s,a) \in \mathbb{R}^{1\times S}. \tag{3.4}$$

where we denote a vector of the transition kernel P or P^0 at state-action pair (s,a). In other words, the uncertainty is imposed in a decoupled manner for each state-action pair, obeying the so-called *sa*-rectangularity (Zhou et al. 2021, Wiesemann et al. 2013). More generally, we define *s*-rectangular MDPs as $\mathcal{U}^{\sigma}_{\|.\|}(P) = \otimes_s \mathcal{U}^{\mathsf{s},\widetilde{\sigma}}_{\|.\|}(P_s)$, for the general smooth L_p norm $\|.\|$. The uncertainty is imposed in a decoupled manner for each state pair, and a fixed budget given a state for all action is defined. To get a similar meaning for the radius of the ball between *sa*-rectangular and *s*-rectangular assumptions, we need to rescale the radius depending on the norm like in Yang et al. (2022). The *s*- uncertainty set is then defined using the rescaled radius $\tilde{\sigma}$ as

$$\mathcal{U}^{\mathsf{s},\widetilde{\sigma}}_{\|.\|}(P_s) \coloneqq \left\{P'_s \in \Delta(\mathcal{S})^{\mathcal{A}} : \|P'_s - P_s\| \leq \tilde{\sigma} = \sigma\|1_A\|\right\}, \tag{3.5}$$

$$P_s \coloneqq P(\cdot,\cdot \,|\, s) \in \mathbb{R}^{1\times SA}, \quad P^0_s \coloneqq P^0(\cdot,\cdot \,|\, s) \in \mathbb{R}^{1\times SA} \tag{3.6}$$

where $1_A \in \mathbb{R}^A$ denotes the unitary vector. For the specific case of respectively L_1,L_p and L_∞ norm, $\tilde{\sigma}$ is equal to $|\sigma\mathcal{A}|, \sigma|\mathcal{A}|^{1/p}$ and σ. Note that this scaling allows for a fair comparison between *sa*- and *s*-rectangular MDPs. In RMDPs, we are interested in the worst-case performance of a policy π over all the possible transition kernels in the uncertainty set. This is measured by the *robust value function* $V^{\pi,\sigma}$ and the *robust Q-function* $Q^{\pi,\sigma}$ in $\mathcal{M}_{\mathsf{rob}}$, defined respectively as $\forall(s,a) \in \mathcal{S}\times\mathcal{A}$

$$V^{\pi,\sigma}(s) \coloneqq \inf_{P\in\mathcal{U}^{\mathsf{sa},\sigma}_{\|.\|}(P^0)} V^{\pi,P}(s), \quad Q^{\pi,\sigma}(s,a) \coloneqq \inf_{P\in\mathcal{U}^{\mathsf{sa},\sigma}_{\|.\|}(P^0)} Q^{\pi,P}(s,a)\ . \tag{3.7}$$

Similarly for *s*-rectangularity, the value function is denoted $V^{\pi,\sigma}_s(s) \coloneqq \inf_{P\in\widetilde{\mathcal{U}^{\mathsf{s},\sigma}_{\|.\|}}(P^0)} V^{\pi,P}(s)$.

Optimal robust policy and robust Bellman operator. As a generalization of properties of standard MDPs in the *sa*-rectangular robust case, it is well-known that there exists at least one deterministic policy that maximizes the robust value function (resp. robust Q-function) simultaneously for all states (resp. state-action pairs) (Iyengar 2005, Nilim and El Ghaoui 2005) but not in the *s*-rectangular case. Therefore, we denote the *optimal robust value function* (resp. *optimal robust Q-function*) as $V^{\star,\sigma}$ (resp. $Q^{\star,\sigma}$), and the optimal robust policy as $\pi^\star$, which satisfy $\forall(s,a) \in \mathcal{S}\times\mathcal{A}$

$$V^{\star,\sigma}(s) \coloneqq V^{\pi^\star,\sigma}(s) = \max_\pi V^{\pi,\sigma}(s), \quad Q^{\star,\sigma}(s,a) \coloneqq Q^{\pi^\star,\sigma}(s,a) = \max_\pi Q^{\pi,\sigma}(s,a). \tag{3.8a}$$

A key concept in RMDPs is a generalization of Bellman's optimality principle, encapsulated in the following *robust Bellman consistency equation* (resp. *robust Bellman optimality equation*):

$$\forall (s,a) \in \mathcal{S} \times \mathcal{A}, \quad Q^{\pi,\sigma}(s,a) = r(s,a) + \gamma \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}_{\|\cdot\|}(P^0_{s,a})} \mathcal{P}V^{\pi,\sigma}, \tag{3.9a}$$

$$\forall (s,a) \in \mathcal{S} \times \mathcal{A} \quad Q^{\star,\sigma}(s,a) = r(s,a) + \gamma \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}_{\|\cdot\|}(P^0_{s,a})} \mathcal{P}V^{\star,\sigma} \ , \tag{3.9b}$$

for the *sa*-rectangular case and same equation replacing $P^0_{s,a}$ by P^0_s and σ by $\tilde{\sigma}$. The robust Bellman operator (Iyengar 2005, Nilim and El Ghaoui 2005) is denoted by $\mathcal{T}^\sigma(\cdot) : \mathbb{R}^{SA} \to \mathbb{R}^{SA}$

$$\mathcal{T}^\sigma(Q^\pi)(s,a) \coloneqq r(s,a) + \gamma \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}_{\|\cdot\|}(P^0_{s,a})} \mathcal{P}V, \quad \text{with} \quad V(s) \coloneqq \max_\pi Q^\pi(s,a) \ , \tag{3.10}$$

for *sa*-rectangular MDPs. Given that $Q^{\star,\sigma}$ is the unique-fixed point of $\mathcal{T}^\sigma$ one can recover the optimal robust value function and Q-function using a procedure termed *distributionally robust value iteration* ($DRVI$). Generalizing the standard value iteration, $DRVI$ starts from some given initialization and recursively applies the robust Bellman operator until convergence. As has been shown previously, this procedure converges rapidly due to the γ-contraction property of $\mathcal{T}^\sigma$ with respect to the L_∞ norm (Iyengar 2005, Nilim and El Ghaoui 2005).

3.3 Distributionally Robust Value Iteration

Generative model-based sampling. Following Zhou et al. (2021), Panaganti and Kalathil (2022b), we assume access to a generative model or a simulator (Kearns and Singh 1999), which allows us to collect N independent samples for each state-action pair generated based on the *nominal* kernel P^0: $\forall (s,a) \in \mathcal{S} \times \mathcal{A}$, $s_{i,s,a} \overset{i.i.d}{\sim} P^0(\cdot \,|\, s,a), \quad i = 1, 2, \cdots, N$. The total sample size is, therefore, NSA. We consider a model-based approach tailored to RMDPs, which first constructs an empirical nominal transition kernel based on the collected samples and then applies distributionally robust value iteration (DRVI) to compute an optimal robust policy. As we decouple the statistical estimation error and the optimization error, we exhibit an algorithm that can achieve arbitrary small error ϵ_{opt} in the empirical MDP defined as an empirical nominal transition kernel $\widehat{P}^0 \in \mathbb{R}^{SA \times S}$ that can be constructed on the basis of the empirical frequency of state transitions, i.e. $\forall (s,a) \in \mathcal{S} \times \mathcal{A}$

$$\widehat{P}^0(s'|s,a) \coloneqq \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\{s_{i,s,a} = s'\}, \tag{3.11}$$

which leads to an empirical RMDP $\widehat{\mathcal{M}}_{\mathsf{rob}} = \{\mathcal{S}, \mathcal{A}, \gamma, \mathcal{U}^\sigma_{\|\cdot\|}(\widehat{P}^0), r\}$. Analogously, we can define the corresponding robust value function (resp. robust Q-function) of policy π in $\widehat{\mathcal{M}}_{\mathsf{rob}}$ as $\widehat{V}^{\pi,\sigma}$ (resp. $\widehat{Q}^{\pi,\sigma}$) (cf. (3.8)). In addition, we denote the corresponding *optimal robust policy* as $\widehat{\pi}^\star$ and the *optimal robust value function* (resp. *optimal robust Q-function*) as $\widehat{V}^{\star,\sigma}$ (resp. $\widehat{Q}^{\star,\sigma}$) (cf. (3.9)), which satisfies the robust Bellman optimality equation $\forall (s,a) \in \mathcal{S} \times \mathcal{A}$:

$$\widehat{Q}^{\star,\sigma}(s,a) = r(s,a) + \gamma \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}_{\|\cdot\|}(\widehat{P}^0_{s,a})} \mathcal{P}\widehat{V}^{\star,\sigma}. \tag{3.12}$$

Equipped with $\widehat{P}^0$, we can define the empirical robust Bellman operator $\widehat{\mathcal{T}}^\sigma$ as $\forall (s,a) \in \mathcal{S} \times \mathcal{A}$

$$\widehat{\mathcal{T}}^\sigma(Q^\pi)(s,a) \coloneqq r(s,a) + \gamma \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}_{\|\cdot\|}(\widehat{P}^0_{s,a})} \mathcal{P}V, \tag{3.13}$$

with $V(s) \coloneqq \max_\pi Q^\pi(s,a)$. The aim of this work is given the collected samples, to learn the robust optimal policy for the RMDP w.r.t. some prescribed uncertainty set $\mathcal{U}^\sigma(P^0)$ around the nominal kernel using as few samples as possible. Specifically, given some target accuracy level $\varepsilon > 0$, the goal is to seek an ε-optimal robust policy $\widehat{\pi}$ obeying

$$\forall s \in \mathcal{S}: \quad V^{\star,\sigma}(s) - V^{\widehat{\pi},\sigma}(s) \leq \varepsilon\,, \tag{3.14}$$

$$\widehat{V}^{\widehat{\pi}^\star,\sigma} - \widehat{V}^{\widehat{\pi},\sigma} \leq \varepsilon_{\mathsf{opt}}. \tag{3.15}$$

This formulation allows plugging any solver of RMDPs in this bound, for instance, the distributionally robust value iteration (DRVI) algorithm detailed in Appendix 10.

3.4 Theoretical guarantees

In this section, we present our main results characterizing the sample complexity of solving RMDPs with *sa*-and *s*-rectangularity. Additionally, we discuss the implications of our results for the comparisons between standard and robust RL, and for comparisons between *sa*- versus *s*-rectangularity.

3.4.1 *sa*-rectangular uncertainty set with general smooth norms

To begin, we consider the RMDPs with *sa*-rectangularity with general norms. We first provide the following sample complexity upper bound for certain oracle planning algorithms, whose proof is postponed to Appendix 7.2. Technically, we derive two new dual forms for RMDPs problems using arbitrary norms in Lemmas 6.3 and 6.4 for respectively *sa*- and *s*-rectangular RMDPS. In these dual forms, a central quantity denoted $\mathrm{sp}(.)_*$, representing the dispersion of the value function, appears and is the dual span semi-norm associated with the considered general L_p norm $\|.\|$ defined in 3.2.1 in the initial primal problem. The main challenge in this analysis is to derive a tight upper bound on this quantity in Lemmas (7.1) and (7.2), leading to the following sample complexity.

Theorem 3.4.1 (Upper bound for *sa*-rectangularity)**.** *Consider the uncertainty set $\mathcal{U}^{\mathsf{sa},\sigma}_{\|\cdot\|}(\cdot)$ associated with arbitrary L_p smooth norm $\|\cdot\|$ defined in 3.2.1. We denote $\sigma_{\max} \coloneqq \max_{p_1,p_2\in\Delta(\mathcal{S})} \|p_1 - p_2\|$ as the accessible maximal uncertainty level. Consider any $\delta \in (0,1)$, discount factor $\gamma \in \left[\frac{1}{4}, 1\right)$, and uncertainty level $\sigma \in (0, \sigma_{\max}]$. Let $\widehat{\pi}$ be the output policy of some oracle planning algorithm with optimization error $\varepsilon_{\mathsf{opt}}$ introduced in (3.15). With introduced in 3.2.1, one has with probability at least $1-\delta$,*

$$\forall s \in \mathcal{S}: \quad V^{\star,\sigma}(s) - V^{\widehat{\pi},\sigma}(s) \leq \varepsilon + \frac{8\varepsilon_{\mathsf{opt}}}{1-\gamma} \tag{3.16}$$

for any $\varepsilon \in (0, \sqrt{1/\max\{1-\gamma, \sigma C_g\}}]$, as long as the total number of samples obeys

$$NSA \gtrsim \frac{c_1 SA}{(1-\gamma)^2 \max\{1-\gamma, C_g\sigma\}\varepsilon^2} + \frac{c_2 SAC_S \|1_S\|_*}{(1-\gamma)^2\epsilon} \tag{3.17}$$

with c_1, c_2, c_3 a universal positive constant. For a sufficiently small level of accuracy $\epsilon \leq (\max\{1-\gamma, C_g\sigma\})/(C_S\|1_S\|)$, the sample complexity is

$$NSA \gtrsim \frac{c_3 SA}{(1-\gamma)^2 \max\{1-\gamma, C_g\sigma\}\varepsilon^2}\,. \tag{3.18}$$

Note that this result is also true for TV without the geometric smooth term depending on C_S. Considering L_p norms, $C_g \geq 1$ and $C_S \leq S^{1/q}(p-1)$. In Theorem 3.4.1, we introduce the following minimax-optimal lower bound to verify the tightness of the above upper bound; a proof is provided in Appendix 8.

Theorem 3.4.2 (Lower bound for *sa*-rectangularity)**.** *Consider the uncertainty set* $\mathcal{U}^{\mathsf{sa},\sigma}_{\|\cdot\|}(\cdot)$ *associated with arbitrary* L_P *norm* $\|\cdot\|$ *defined in* 3.2.1*. We denote* $\sigma_{\max} \coloneqq \max_{p_1,q_1\in\Delta(\mathcal{S})} \|p_1 - p_2\|$ *as the accessible maximal uncertainty level. Consider any tuple* $(S, A, \gamma, \sigma, \varepsilon)$*, where* $\gamma \in \left[\frac{1}{2}, 1\right)$*,* $\sigma \in (0, \sigma_{\max}(1-c_0)]$ *with* $0 < c_0 \leq \frac{1}{8}$ *being any small enough positive constant, and* $\varepsilon \in \left(0, \frac{c_0}{256(1-\gamma)}\right]$*. We can construct two infinite-horizon RMDPs* $\mathcal{M}_0, \mathcal{M}_1$ *such that giving a dataset with* N *independent samples for each state-action pair over the nominal transition kernel (for either* $\mathcal{M}_0$ *or* $\mathcal{M}_1$ *respectively), one has*

$$\inf_{\widehat{\pi}} \max_{\mathcal{M}\in\{\mathcal{M}_0,\mathcal{M}_1\}} \left\{ \mathbb{P}_{\mathcal{M}}\Big(\max_{s\in\mathcal{S}} \left[V^{\star,\sigma}(s) - V^{\widehat{\pi},\sigma}(s)\right] > \varepsilon \Big) \right\} \geq \frac{1}{8} ,$$

where the infimum is taken over all estimators $\widehat{\pi}$*,* $\mathbb{P}_0$ *(resp.* $\mathbb{P}_1$*) are the probability when the RMDP is* $\mathcal{M}_0$ *(resp.* $\mathcal{M}_1$*), as long as, for* c_7 *is a universal positive constant,*

$$NSA \leq \frac{c_7 SA}{(1-\gamma)^2 \max\{1-\gamma, C_g\sigma\}\varepsilon^2} . \tag{3.19}$$

• **Near minimax-optimal sample complexity with general L_p norms.** Recall that Theorem 3.4.1 shows that the sample complexity upper bound of oracle algorithms for RMDPs is in the order of

$$\widetilde{O}\left(\frac{SA}{(1-\gamma)^2 \max\{1-\gamma, C_g\sigma\}\varepsilon^2}\right).$$

Combined with the lower bound in Theorem 3.4.2, we observe that the above sample complexity is near minimax-optimal, in almost the full range of uncertainty.

• **Solving RMDPs with general L_p norms can be easier than solving standard RL.** Recall that the sample complexity of solving standard RL with a generative model (Agarwal et al. 2020, Li et al. 2024, Azar et al. 2013a) is: $\widetilde{O}\left(\frac{SA}{(1-\gamma)^3\varepsilon^2}\right)$. Comparing this with the sample complexity in (3.18), it highlights that solving robust MDPs (cf. (3.18)) using any norm as the divergence function for the uncertainty set is not harder than (and is sometimes easier than) solving standard RL (cf. (3.4.1)). Specifically, when the uncertainty level is small $\sigma \lesssim 1-\gamma$, the sample complexity of solving robust MDPs matches that of standard MDPs. While when the uncertainty level is relatively larger $1-\gamma \lesssim \sigma \leq \sigma_{\max}$, the sample complexity of solving robust MDPs is smaller than that of standard MDPs by a factor or $\frac{\sigma}{1-\gamma}$, which goes to $\frac{1}{1-\gamma}$ when $\sigma = O(1)$.

• **Comparisons with prior arts.** In Figure 3.1, we illustrate the comparisons with two state-of-the-arts (Clavier et al. 2023, Shi et al. 2023) which use some divergence functions belonging to the class of general norms considered in this work. In particular, Shi et al. (2023) achieved the state-of-the-art minimax-optimal sample complexity $\widetilde{O}\left(\frac{SA}{(1-\gamma)^2\max\{1-\gamma,\sigma\}\varepsilon^2}\right)$ for specific L_1 norm (or called total variation distance). In this work, we attain near minimax-optimal sample complexity for any general norm (including L_1) which matches the one in Shi et al. (2023) when narrowing down to L_1 norm. Note that in TV case, $C_g = 1$. This reveals that the finding of robust MDPs can be easier than standard MDPs (Shi et al. 2023) in terms of sample requirement does not only hold for L_1 norm, but for any general norm. In addition, compared to Clavier et al. (2023) which focuses on L_p

norms for any $1 \leq p \leq \infty$: when $1-\gamma \lesssim \sigma \leq \sigma_{\max}$, we improve the sample complexity $\widetilde{O}(\frac{SA}{(1-\gamma)^4\varepsilon^2})$ to $\widetilde{O}(\frac{SA}{(1-\gamma)^2\sigma\varepsilon^2})$ by at least a factor of $\frac{1}{1-\gamma}$; otherwise, we match the results in Clavier et al. (2023).

Burn-in Condition, C_g factor and TV case : In Th. 3.4.1 and 3.4.3 we need a sufficiently small level of accuracy $\epsilon \leq (\max\{1-\gamma, C_g\sigma\})/(C_s\|1_S\|)$, to obtain the sample complexity. This type of condition is usual in MDPS analysis Shi et al. (2022) and is equivalent to burn in term. Moreover, the quantity C_S exists (see 3.2.1) and for example, considering L_p norms, C_S is bounded by $S^{1/q}$. (See (A.203)) and the product $C_S\|1_S\|$ is upper bounded by S for L_2 norm. Moreover, note that our theorem for the smooth norm is also true for TV which is not C^2 and has the same complexity as (Shi et al. (2023). In this case, the burn-in condition is not needed. (See Lemma 7.3.3). Finally, the factor $C_g = 1/\min_s \|e_s\|$ is norm dependent and depends on how big the vector e_{s_0} is in the considered norm. Note for classical L_p this quantity is bigger than 1, which reduces the sample complexity.

3.4.2 s-rectangular uncertainty set with general norms

To continue, we move on to the case when the uncertainty set is constructed under s-rectangularity smooth norm. The following theorem presents the sample complexity upper bound for learning an ϵ-optimal policy for RMDPs with s-rectangularity. A proof is shown in Appendix 7.2.

Theorem 3.4.3 (Upper bound for s-rectangularity)**.** *Consider the uncertainty set $\mathcal{U}^{\mathsf{s},\widetilde{\sigma}}_{\|\cdot\|}(\cdot)$ with s-rectangularity. Consider any discount factor $\gamma \in \left[\frac{1}{4}, 1\right)$, the rescaled uncertainty level $\tilde{\sigma} = \sigma\|1_A\|$, and denote $\tilde{\sigma}_{\max} := \|1_A\|, \max_{p_1,p_2\in\Delta(\mathcal{S})}\|p_1 - p_2\|$ and $\delta \in (0,1)$. Let $\widehat{\pi}$ be the output policy of an arbitrary optimization algorithm with error $\varepsilon_{\mathsf{opt}}$. , with probability at least $1-\delta$, one has for any $\varepsilon \in (0, \sqrt{1/\max\{1-\gamma, C_g \min_s \|\pi_s\|_* \sigma\}}]$,*

$$\forall s \in \mathcal{S}: \quad V^{\star,\widetilde{\sigma}}(s) - V^{\widehat{\pi},\widetilde{\sigma}}(s) \leq \varepsilon + \frac{8\varepsilon_{\mathsf{opt}}}{1-\gamma}$$

as long as the total number of samples obeys

$$NSA \gtrsim \frac{c_4 SA}{(1-\gamma)^2\varepsilon^2} \min\left\{\frac{1}{\max\{1-\gamma, C_g\sigma\}}, \frac{1}{\sigma C_g \min\limits_{s\in\mathcal{S}}\left\{\|\pi^*_s\|_* \|1_A\|, \|\hat{\pi}_s\|_* \|1_A\|\right\}}\right\} + \frac{c_5 SAC_S \|1_S\|_*}{(1-\gamma)^2\epsilon}. \tag{3.20}$$

For a sufficiently small accuracy, $\epsilon \leq (\max\{1-\gamma, C_g\tilde{\sigma}\})/(C_s\|1_S\|)$ the sample complexity is

$$NSA \gtrsim \frac{c_6 SA}{(1-\gamma)^2\varepsilon^2} \min\left\{\frac{1}{\max\{1-\gamma, C_g\sigma\}}, \frac{1}{\sigma C_g \min_{s\in\mathcal{S}}\left\{\|\pi^*_s\|_* \|1_A\|, \|\hat{\pi}_s\|_* \|1_A\|\right\}}\right\} \tag{3.21}$$

where $\hat{\pi}_s \in \Delta_A$ denote the policy of the empirical RMPDs at state s, $\pi^*_s \in \Delta_A$ the optimal policy given s of the true RMPDs, $\|.\|_*$ the dual norm and c_4, c_5, c_6 are universal constant. Note that this result is also true for TV without the term depending on smoothness C_S. In addition, we provide the lower bounds for a representative divergence function L_∞ norm in the following. Note that for classical L_p, $C_S = S^{1/q}(p-1)$ and C_g can be lower bounded by 1. A proof is provided in Appendix 9.

Theorem 3.4.4 (Lower bound for s-rectangularity)**.** *Consider the uncertainty set $\mathcal{U}^{\mathsf{s},\widetilde{\sigma}}_{L_\infty}(\cdot)$ associated with the L_∞ norm. Consider any tuple $(S, A, \gamma, \sigma, \varepsilon)$ and $0 < c_0 \leq \frac{1}{8}$ being any small enough positive constant, where $\gamma \in \left[\frac{1}{2}, 1\right)$, and $\varepsilon \in (0, \frac{c_0}{256(1-\gamma)}]$. Correspondingly, we denote the*

accessible maximal uncertainty level for $\mathcal{U}_{L_\infty}^{\mathsf{s},\widetilde{\sigma}}(\cdot)$ *as* $\sigma_{\max}^{\infty} := \max_{p_1,p_1\in\Delta(\mathcal{S})^A} \|p_1 - p_2\|_\infty = 1$. *Then we can construct a collection of infinite-horizon RMDPs* $\mathcal{M}_{L_\infty}$ *defined by the uncertainty set with* $\mathcal{U}_{L_\infty}^{\mathsf{s},\widetilde{\sigma}}(\cdot)$ *so that for any* $\sigma \in (0, \sigma_{\max}^{\infty}(1-c_0)]$, *and any dataset with in total* N_{all} *independent samples for all state-action pairs over the nominal transition kernel (for any RMDP inside* $\mathcal{M}_{L_\infty}$*), one has*

$$\inf_{\widehat{\pi}} \max_{\mathcal{M}\in\mathcal{M}_{L_\infty}} \left\{ \mathbb{P}_{\mathcal{M}}\big(\max_{s\in\mathcal{S}} \left[V^{\star,\sigma}(s) - V^{\widehat{\pi},\sigma}(s)\right] > \varepsilon \big) \right\} \geq \frac{1}{8}, \tag{3.22}$$

provided that for c_8 *is a universal positive constant,*

$$N_{\mathsf{all}} \leq \frac{c_8 SA}{(1-\gamma)^2 \max\{1-\gamma, \tilde{\sigma}\}\varepsilon^2}. \tag{3.23}$$

with $\mathbb{P}_{\mathcal{M}}$ *the probability when the RMDP is* $\mathcal{M}$, *and the infimum is taken over all estimators* $\widehat{\pi}$.

Now we can present some implications of Theorem 3.4.3 and Theorem 3.4.4.

• **Robust MDPs with *s*-rectangularity are at least as easy as *sa*-rectangularity.** Theorem 3.4.3 shows that the sample complexity of solving RMDPs with s-rectangularity does not exceed the order of $\widetilde{O}\left(\frac{SA}{(1-\gamma)^2\max\{1-\gamma, C_g\sigma\}\varepsilon^2}\right)$. This matches the sample complexity for sa-rectangularity (cf. (3.18)) and indicates that although s-rectangular RMDPs are of a more complicated formulation, solving s-rectangular RMDPs is at least as easy as solving sa-rectangular RMDPs in terms of the sample complexity. In addition to the worst-case sample complexity upper bound, Theorem 3.4.3 also provides a data and instance-dependent sample complexity upper bound for s-rectangular RMDPs (cf. in (3.20)).Taking the divergence function $\|\cdot\| = L_p$ for instance, the data and instance-dependent sample complexity upper bound is

$$\begin{cases} \widetilde{O}\left(\frac{SA}{(1-\gamma)^2\varepsilon^2}\frac{1}{\max\{1-\gamma,\sigma\}}\right) & \text{if } \widehat{\pi}_s(a\,|\,s) = \pi_s^*(a\,|\,s) = \frac{1}{A}, \quad \forall (s,a) \in \mathcal{S}\times\mathcal{A} \\ \widetilde{O}\left(\frac{SA}{(1-\gamma)^2\varepsilon^2}\frac{1}{\max\{1-\gamma,\sigma A^{1/p}\}}\right) & \text{if } \|\widehat{\pi}_s(\cdot\,|\,s)\|_0 = \|\pi_s^*(\cdot\,|\,s)\|_0 = 1, \quad \forall s \in \mathcal{S} \end{cases}$$

where $\|.\|_0$ corresponds to the total number of nonzero elements in a vector.The intuition beyond this theorem is that when the policy becomes proportional to uniform, the uncertainty budget of the s-rectangular MDPs is equally spread into all actions, and we retrieve the sa-rectangular case. When the policy becomes deterministic, all the uncertainty budget concentrates on one action. In this case, most of the actions are not robust except one, and the problem is simpler than classical MDP for this only specific action. An illustration of this result can be found in Fig. 3.2.

• **Comparisons with prior arts.** In Figure 3.1, we illustrate the comparisons with Clavier et al. (2023) which use L_p norms functions belonging to the class of general norms considered in this work. We do not compare in this section to Yang et al. (2021) as it is not anymore state-of-the-art with regard to the work of Clavier et al. (2023). In particular, the latest achieves in the s-rectangular case at sample complexity of $\widetilde{O}\left(\frac{SA}{(1-\gamma)^3\varepsilon^2}\right)$ in the regime where $\tilde{\sigma} \lesssim 1-\gamma$. In this regime, our result is the same but more general but in the regime where $\tilde{\sigma} \gtrsim 1-\gamma$, they achieve sample complexity of $\widetilde{O}\left(\frac{SA}{(1-\gamma)^4\varepsilon^2}\right)$ which is bigger than our result $\widetilde{O}\left(\frac{SA}{(1-\gamma)^2\max\{1-\gamma,\tilde{\sigma}\}\varepsilon^2}\right)$ by a factor at least $\frac{1}{1-\gamma}$.

3.5 Conclusion

This work refined sample complexity bounds to learn robust Markov decision processes when the uncertainty set is characterized by an general L_p metric, assuming the presence of a generative

model. Our findings not only strengthen the current knowledge by improving both the upper and lower bounds, but also highlight that learning s-rectangular MDPs is less challenging in terms of sample complexity compared to classical sa-rectangular MDPs. This work is the first to provide results with a minimax bound, as prior results concerning s-rectangular cases were not minimax optimal. Additionally, we have established the minimax sample complexity for RMDPs using a general L_p norm, demonstrating that it is never larger than that required for learning standard MDPs. Our research identifies potential avenues for future work, such as exploring the characterization of tight sample complexity for RMDPs under a broader family of uncertainty sets, such as those defined by f-divergence. It would be highly desirable for a more unified theoretical foundation, as the distance between probability measures is more natural to define using divergence. Moreover, it would be interesting to focus on the finite-horizon Setting and linear setting, as our current analytical framework opens the door for potential extensions to address finite-horizon RMDPs. Such an extension would contribute to a more comprehensive understanding of tabular cases. Finally, the case of linear MDPs would be interesting to explore.

Part II

Practical Robust Reinforcement Learning

Chapter 4

Robust Reinforcement Learning with Distributional Risk-averse formulation

Contents

Ever tried. Ever failed. No matter. Try again. Fail again. Fail better.
Samuel Beckett, Worstward Ho

4.1 Introducion

The classical Reinforcement Learning (RL)Sutton and Barto (2018) problem using Markov Decision Processes (MDPs) modelization gives a practical framework to solve sequential decision problems under uncertainty of the environment. However, for real-world applications, the final chosen policy can sometimes be very sensitive to sampling errors, inaccuracy of the model parameters, and definition of the reward.

This problem motivates robust Reinforcement Learning, aiming to reduce such sensitivity by taking to account that the transition and/or reward function (P, r) may vary arbitrarily inside a given uncertainty set. The optimal solution can be seen as the solution that maximizes a worst-case problem in this uncertainty set or the result of a dynamic zero-sum game where the agent tries to find the best policy under the most adversarial environment (Abdullah et al. 2019). In general, this problem is NP-hard (Wiesemann et al. 2013) due to the complex max-min

problem, making it challenging to solve in a discrete state action space and to scale to a continuous state action space.

Many algorithms exist for the tabular case for Robust MDPs with Wasserstein constraints over dynamics and reward such as Yang (2017), Petrik and Russel (2019), Grand-Clément and Kroer (2020a;b) or for L_∞constrained S-rectangular Robust MDPs (Behzadian et al. 2021). Here we focus on a *more general continuous state space* $\mathcal{S}$ with a discrete or continuous action space $\mathcal{A}$ and with constraints defined using f-divergence.

Robust RL (Morimoto and Doya 2005) with continuous action space focuses on robustness in the dynamics of the system (changes of P) and has been studied in Abdullah et al. (2019), Singh et al. (2020), Urpí et al. (2021), Eysenbach and Levine (2021) among others. Eysenbach and Levine (2021) tackles the problem of both reward and transition using Max Entropy RL, whereas the problem of robustness in action noise perturbation is presented in Tessler et al. (2019). Here, we tackle the problem of robustness *through dynamics of the system.*.

In this paper, we show that it is possible to tackle a Robust Distributional Reinforcement Learning problem with f-divergence constraints by solving a risk-averse RL problem, using a formulation based on mean standard deviation optimization.

The idea beyond that relies on the argument from Robust Learning theory, stating that Robust Learning under an uncertainty set defined with f-divergence is asymptotically close to Mean-Variance (Gotoh et al. 2018) or Mean-Standard deviation optimization (Duchi et al. 2016, Duchi and Namkoong 2018).

In this work, we focus on the idea that generalization, regularization, and robustness are strongly linked in RL or MDPs as shown in Husain et al. (2021), Derman and Mannor (2020), Derman et al. (2021), Ying et al. (2021), Brekelmans et al. (2022). We show that is it possible to improve the Robustness of RL algorithms with variance/standard deviation regularisation. Moreover, the problem of uncertainty under the distribution of the environment is transformed into a problem with uncertainty over the distribution of the rewards, which makes it tractable.

Note that our work is related to Smirnova et al. (2019b) as they penalise the expectation by the variance of returns. However, their approach differs from ours since they use the variance estimate under a Gaussian assumption of distributions while we use a standard deviation penalization without any distribution assumptions. Moreover, the idea of robustness in the change of dynamics is not demonstrated numerically, and the problem tackled is different since they consider close policy distributions, while we consider dynamic distributions.

The contribution of the work is the following: we motivate the use of standard deviation penalization and derive two algorithms for discrete and continuous action space that are robust to changes in dynamics. These algorithms only require one additional parameter tuning, which is the Mean-Standard Deviation trade-off. Moreover, we show that our formulation using Distributional Reinforcement Learning is robust to changing transition dynamics in environments with both discrete and continuous action spaces both in the Mujoco suite and in stochastic environments derived from Mujoco.

Related topics : Regularised MDPs : Policy Regularisation in RL Geist et al. (2019) has been studied and led to state-of-the-art algorithms such as PPO and SAC (Schulman et al. 2017b, Haarnoja et al. 2018b, Vieillard et al. 2020). In these algorithms, an additional penalisation based on the current policy is added to the classical objective function. The idea is different, as we penalize our mean objective function using the standard deviation of the return distribution. Being pessimistic about the distributional state-value function leads to more stable learning, reduces the variance, and, tends to improve the robustness of systems as demonstrate

(Brekelmans et al. 2022). Recent advances in Robust MDPs have shown a link between this field and Regularised MDPs as in Derman et al. (2021), Kumar et al. (2022).

Distributional RL : Second-order estimation is done using Distributional Reinforcement Learning (Bellemare et al. 2017, Zhang and Weng Zhang and Weng) using a quantile estimate of our distribution to approximate our action value function (Dabney et al. 2017; 2018a) with the QRDQN and IQN algorithms. Distributional state-action function representation is also used to learn an accurate critic for a policy-based algorithm, such as in Kuznetsov et al. (2020), Ma et al. (2021), Nam et al. (2021).

Risk-Averse RL : Risk-averse RL aims at minimizing different objectives than the classical mean optimization e.g. CVaR or other risk measures. For example, Dabney et al. (2018a), Ma et al. (2021) use distributional RL for optimizing different risk measures. Our goal is to show the robustness of using risk-averse solutions to our initial problem. Our formulation is close to mean-variance formulation (Jain et al. 2021, Wang and Zhou 2020) that already exists in risk-averse RL, although not using a distributional framework that shows highly competitive performance in a controlled setting.

Pessimism and Optimism in Distributional RL Moskovitz et al. (2021) describes a way of performing Optimistic / Pessimistic Deep RL using a constructed confidence interval with the variance of rewards. Their work is close to ours in the pessimistic case but the confidence interval is expressed in terms of variance of expectation estimate and not using the variance of the distribution itself. Moreover, they use an adaptative regularizer where we look at the interest of using a fixed parameter.

Notations: Considering a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho)$, where $\mathcal{A}$ is the action space, $\mathcal{S}$ is the state space, $P\left(s' \mid s, a\right)$ is the reward and transition distribution from state s to s' taking action a and $\gamma \in (0,1)$ is the discount factor. Stochastic policy are denoted $\pi(a \mid s) : \mathcal{S} \to \Delta(\mathcal{A})$ and we consider usually the case where action space is continuous and action space is either discrete our continuous.

A rollout or trajectory using π from state s using initial action a is defined as the the random sequence $\tau^{P,\pi|s_0,a_0} = \left(\left(s_0, a_0, r_0(s_0,a_0)\right), \left(s_1, a_1, r_1, r_0(s_1,a_1)\right), \ldots\right)$ with $s_0 = s, a_0 = a, a_t \sim \pi\left(\cdot \mid s_t\right)$ and $s_{t+1} \sim P\left(\cdot \mid s_t, a_t\right)$; we denote the distribution over rollouts by $\mathbb{P}(\tau)$ with $\mathbb{P}(\tau) = \rho\left(s_0\right) \prod_{t=0}^{T} P\left(s_{t+1} \mid s_t, a_t\right) \pi\left(a_t \mid s_t\right) d\tau$ and usually write $\tau \sim \mathbb{P} = (P, \pi)$. Moreover, considering the the distribution of discounted cumulative return $Z^{P,\pi}(s,a) = R(\tau^{P,\pi|s,a})$ with $R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t(s_t, a_t)$, the Q-function $Q^{P,\pi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ of π is its expected discounted cumulative return of the distribution

$$Q^{\pi,P}(s,a) := \mathbb{E}[Z^{\pi,P}(s,a)] = \mathbb{E}_{\tau\sim(\pi,P)}\left[R(\tau) \mid a_t \sim \pi\left(\cdot \mid s_t\right), s_{t+1} \sim P\left(\cdot \mid s_t, a_t\right), s_0 = s, a_0 = a\right].$$

The initial goal of (RL) also called risk-neutral RL, is to find the optimal policy π^* where $Q^{P,\pi^*}(s,a) \geq Q^{P,\pi}(s,a)$ for all π and $s \in \mathcal{S}, a \in \mathcal{A}$. Finally, the Bellman operator $\mathcal{T}^\pi$ and Bellman optimal operator $\mathcal{T}^*$ can be defined as follow :

$$\begin{aligned}
\mathcal{T}^\pi Q(s,a) &:= r(s,a) + \gamma \mathbb{E}_{s'\sim P, a'\sim\pi}\left[Q\left(s', a'\right)\right] \\
\mathcal{T}^* Q(s,a) &:= r(s,a) + \gamma \mathbb{E}_{s'\sim P}\left[\max_{a'} Q\left(s', a'\right)\right].
\end{aligned}$$

Applying either operator from some initial Q^0 lead to fixed point Q^π or Q^* at a geometric rate as both operators are contractive. Simplifying the notation with regards to s, a, π and P, we define the set of greedy policies w.r.t. Q called $\mathcal{G}(Q) = \text{argmax}_{\pi\in\Pi}\langle Q, \pi\rangle$. A classical approach

to estimate an optimal policy is Approximate Modified Policy Iteration (AMPI) Scherrer et al. (2015),

$$\begin{cases} \pi_{k+1} \in \mathcal{G}\left(Q_k\right) \\ Q_{k+1} = \left(T^{\pi_{k+1}}\right)^m Q_k + \epsilon_{k+1} \end{cases}$$

which usually reduces to Approximate Value Iteration (AVI, $m = 1$) and Approximate Policy Iteration (API, $m = \infty$) as special cases. The term ϵ_{k+1} accounts for errors made when applying the Belleman Operator.

4.2 Robust formulation in greedy step of AVI.

In this section, we would like to find policy that are robust to change of environment law P as small variations of P should not affect to much the new policy in the greedy step. In our case we are not looking at classical greedy step $\pi' \in \mathcal{G}(Q) = \text{argmax}_{\pi\in\Pi}\langle Q, \pi\rangle$ but at the following greedy step :

$$\pi' \in \mathcal{G}(Q) = \arg\max_{\pi\in\Pi}\langle \min_P Q^{\pi,P}, \pi\rangle$$

.

With this reformulation, we need to constraint the set of admissible transitions from state-action to the next state P to get a solution of the problem. In general without constraint, the problem is NP-Hard and we have to constrain the problem to distributions that are not too far from the original using distance between distribution such that Wasserstein metric (Abdullah et al. 2019) or other specific distance where the problem can be simplify (Eysenbach and Levine 2021). Moreover, an explicit form for $\min_P Q^{P,\pi}$ given a particular divergence our distance between probability distribution would allow a simplification of the greed step and transforming this max-min problem into a simple one. In fact, a simplification is possible using f-divergence $\mathcal{H}_f$ to constrain the problem with Φ a closed convex function such that $\Phi : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ and $f(z) \geq f(1) = 0$ for all $z \in \mathcal{R}$.

$$\mathcal{H}_f\left(\mathbb{Q} \mid \mathbb{P}\right) = \left\{ \begin{array}{ll} \sum_{i:p_i>0} p_i f\left(\frac{q_i}{p_i}\right) & ; \quad \sum_{i:p_i>0} q_i = 1,\ q_i \geq 0 \\ +\infty \quad \text{otherwise.} & \end{array} \right\}$$

This constraint requires $q_i = 0$ if $p_i = 0$ so the measure $\mathbb{Q}$ absolutely continuous with respect to $\mathbb{P}$. The χ^2-divergence are a particular case of f-divergence with $f(z) = (z-1)^2$. For trajectories τ sampled from distribution $\mathbb{P}^0 = (\pi, P^0)$ and looking at distribution $\mathbb{P}$ closed to $\mathbb{P}^0$ with regards to χ^2-divergence, the minimisation problem reduces to :

$$\min_{\mathbb{P}\in D_{\chi^2}(\mathbb{P}\|\mathbb{P}^0)\leq\alpha} Q^{\pi,P}(s,a) = Q^{P^0,\pi}(s,a) - \alpha^{1/2}\mathbb{V}_{P^0}[Z(s,a)]^{\frac{1}{2}} \ . \tag{4.1}$$

The proof can be found in Appendix 11 for α such that $\alpha \leq \frac{\mathbb{V}[Z^{P_0}]}{\left\|\tilde{Z}^{P_0}\right\|_\infty^2} \leq 1$ with $\tilde{Z}^{P_0} = Z^{P_0} - \mathbb{E}[Z^{P_0}]$ the centered return distribution and $\mathbb{V}[Z^{P_0}]$ the variance of returns. For $\alpha > \frac{\mathbb{V}[Z^{P_0}]}{\left\|\tilde{Z}^{P_0}\right\|_\infty^2}$, the equality becomes an inequality, but we still optimize a lower bound of our initial problem. Defining a new greedy step which is penalized by the standard deviation.

Note that this result can be obtained for any α using the same proof as in Iyengar (2005), Lemma 5, but doing a relaxation of the problem where probabilities of trajectories can be possibly negative. The main difference with classical RMDPs defined in 3.7 in is that this formulation, the minimum operator is taken over the probability of the trajectory τ and not only in the transition kernel of the next state P. Using this formulation, this gives a penalisation (here the the standard deviation) which depend on the distribution of returns starting from state-action space (s, a) which is not the case writing classical RMDPs formulation where penalisation are usually global quantities which does not depend on (s, a). (See Introduction in 3.7. Defining a new standard deviation return penalised greedy step :

$$\pi' \in \mathcal{G}_\alpha(Q) = \arg\max_{\pi\in\Pi} \langle \min_{P\in D_{\chi^2}(P\|P^0)\leq\alpha} Q^{P,\pi}, \pi\rangle = \arg\max_{\pi\in\Pi} \langle Q^{P^0,\pi} - \alpha^{1/2}\mathbb{V}_{P^0}[Z(s,a)]^{\frac{1}{2}}, \pi\rangle,$$

we now look at the the current AMPI to improve robustness :

$$\begin{cases} \pi_{k+1} \in \mathcal{G}_\alpha\left(Q_k\right) \\ Q_{k+1} = \left(T^{\pi_{k+1},P}\right)^m Q_k + \epsilon_{k+1} \end{cases}$$

Approximate identity like 4.1 for a larger class of Φ-divergence and not only χ^2 ca be found in the work of (Duchi et al. 2016).

Robustness is not present in the evaluation step as we use classical Bellman Operator in contrast of the work of (Derman et al. 2021) but only in the greedy step. This idea is very closed to Risk-averse formulation in RL (i.e minimizing risk measure and not only the mean of rewards) but here the idea is approximate a robustness problem in RL. To do so, standard deviation of the distribution of the returns must be estimated. Many ways are possible but we will privilege distributional RL (Bellemare et al. 2017, Dabney et al. 2017; 2018a) which achieve very good performances in many RL applications. Estimating quantiles of the distribution of return, we can simply estimate standard deviation using classical estimator of the standard deviation given the quantiles over an uniform grid $\{q_i(s,a)\}_{1\leq i\leq n}, \forall(s,a) \in \mathcal{S}\times\mathcal{A}$.

$$\mathbb{V}[Z(s,a)]^{\frac{1}{2}} = \sigma(s,a) = \sqrt{\sum_{i=1}^{n} \left(q_i(s,a) - \bar{q}(s,a)\right)^2}$$

where $\bar{q}$ is the classical estimator of the mean. A different interpretation of this formulation could be that taking actions with less variance, we constructing a confidence interval with the standard deviation of the distribution

$$Z^{\pi,P}(s,a) \overset{d}{=} \bar{Z}(s,a) - \alpha\sigma(s,a)\ .$$

This idea is present in classical UCB algorithms (Auer 2002) or pessimism/optimism Deep RL. Here we construct confidence interval using the distribution of the return and note different estimates of the Q function such as in Moskovitz et al. (2021), Bai et al. (2022). In the next section, we derive two algorithms, one for discrete action space and one for continuous action space using this idea. A very interesting way of doing robust Learning is by doing Max entropy RL such as in the SAC algorithm. In Eysenbach and Levine (2021), a demonstration that SAC is a surrogate of Robust RL is demonstrated formally and numerically and we will compare our algorithm to this method.

4.3 Algorithms based on Distributional RL

To derive our algorithms, estimation the second-order moment of the distribution of return must be carried out. For discrete action space a variant of QR-DQN (Dabney et al. 2017) with mean-standard deviation objective is proposed whereas for continuous action space, we propose a mean-standard TQC algorithm (Kuznetsov et al. 2020) based on soft-actor framework as it as already show some robustness as a surrogate of Robust RL (Eysenbach and Levine 2021).

4.3.1 Distributional RL using quantile representation

Distributional RL aims at approximating the return random variable $Z^{\pi,P}(s,a) := \sum_{t=0}^{\infty} \gamma^t R(s_t,a_t)$ with $s_0 = s, a_0 = a$ and $s_{t+1} \sim P(\cdot \mid s_t, a_t), a_t \sim \pi(\cdot \mid s_t)$, as the classical RL framework approximate the expectation of the return or the Q-function, $Q^{\pi,P}(s,a) := \mathbb{E}\left[Z^{\pi,P}(s,a)\right]$. Many algorithms and distributional representation fo the critic exits (Bellemare et al. 2017, Dabney et al. 2017; 2018a) but here we will focus on QR-DQN Dabney et al. (2017) that approximates the distribution of returs $Z^{\pi}(s,a)$ with $Z_\psi(s,a) := \frac{1}{M}\sum_{m=1}^{M} \delta\left(\theta_\psi^m(s,a)\right)$, a mixture of atoms-Dirac delta functions located at $\theta_\psi^1(s,a), \ldots, \theta_\psi^M(s,a)$ given by a parametric model $\theta_\psi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^M$.

Parameters ψ are obtained by minimizing the averaged over the 1-Wasserstein distance between Z_ψ and the temporal difference target distribution $\mathcal{T}^{\pi,P} Z_{\bar{\psi}}$, where $\mathcal{T}_\pi$ is the distributional Bellman operator defined in Bellemare et al. (2017). The control version or optimal operator is denoted $\mathcal{T} Z_{\bar{\psi}}$,

$$\mathcal{T}^{\pi,P} Z(s,a) = R(s,a) + \gamma Z\left(s', a'\right) \text{ with } s' \sim P(\cdot \mid s,a), a' \sim \pi\left(\cdot \mid s'\right)$$

Considering $\mathcal{Z}$ be the space of action-value distributions with finite moments: $\mathcal{Z} = \{Z : \mathcal{X} \times \mathcal{A} \to P(\mathbb{R})\}$ with $\mathbb{E}\left[|Z(x,a)|^p\right] < \infty, \forall (x,a), p \geq 1$ Bellemare et al. (2017) show that :

$$\left\| \mathbb{E}\mathcal{T}^{\pi,P} Z_1 - \mathbb{E}\mathcal{T}^{\pi,P} Z_2 \right\|_\infty \leq \gamma \left\| \mathbb{E}Z_1 - \mathbb{E}Z_2 \right\|_\infty$$

so point wise convergence is exponentially fast for the the mean of the distribution as in the classical case. According to Dabney et al. (2017), the minimization of the 1-Wasserstein loss can be done by learning quantile locations for fractions $\tau_m = \frac{2m-1}{2M}, m \in [1..M]$ via quantile regression loss, defined for a quantile fraction $\tau \in [0,1]$ as :

$$\begin{aligned} \mathcal{L}_{\mathrm{QR}}^{\tau}(\theta) : &= \mathbb{E}_{\tilde{Z} \sim Z}\left[\rho_\tau(\tilde{Z} - \theta)\right], \text{ with} \\ \rho_\tau(u) &= u(\tau - \mathbb{I}(u < 0)), \forall u \in \mathbb{R} . \end{aligned}$$

Finally, to obtain better gradients when u is small, Huber quantile loss (or asymmetric Huber loss) can be used:

$$\rho_\tau^H(u) = |\tau - \mathbb{I}(u < 0)| \mathcal{L}_H^1(u) ,$$

where $\mathcal{L}_H^1(u)$ is a classical Huber loss with parameter 1. The quantile representation has the advantage of not fixing the support of the learned distribution and is used to represent the distribution of return in our algorithm for both discrete and continuous action space.

4.3.2 Mean-standard deviation RL with discrete action space

Once the estimation is done, a phase of policy improvement is done using a Q-learning style algorithm with distributional estimation like QR-DQN (Dabney et al. 2017). The main difference

in our case is that we are not taking the expectation in this phase but mean-standard deviation objective 4.2 in the greedy step estimated using M quantile over a uniform grid on $[0, 1]$. Formally we choose actions with less variance to improve robustness using classical empirical estimator of the variance one the quantiles estimated. However, the estimation step of the algorithm remain the same than in classical QR-DQN algorithm. Parameters ψ of the quantile network are classically updated using a stochastic gradient descent where $\hat{\nabla}$ represent a stochastic estimate of the gradient. Moreover, β controls the learning of the target quantile network parametrised by $\bar{\psi}$.

$$a^* = \arg\max_{a \in \mathcal{A}} \xi_\alpha Z^{\pi,P}(s,a) = \arg\max_{a \in \mathcal{A}} \mathbb{E}[Z^{\pi,P}(s,a)] - \sqrt{\alpha \mathbb{V}[Z^{\pi,P}(s,a)]} \tag{4.2}$$

Algorithm 4: QR-DQN with Standard Deviation penalisation

Initial critics $Z_\psi, Z_{\bar{\psi}}$
for each iteration **do**
  for each step of the environment **do**
    collect (s_t, a_t, r_t, s_{t+1}) according to $\pi(a_t|s_t) = \arg\max_a \xi_\alpha Z^{\pi,P}(s_t, a_t)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, r_t, s_{t+1})\}$
  end for
  for each gradient steps **do**
    Sample batch (s, a, r, s') of $\mathcal{D}$
    Take $a^* = \arg\max_{a'} \xi_\alpha Z^{\pi,P}(s', a')$
    $y_i(s,a) \leftarrow r + \gamma \theta^i_\psi(s', a^*), i \in [1..M]$
    $J_Z(\psi) = \mathbb{E}_{\mathcal{D},\pi} \sum_{i,j=1}^m \rho^H_{\tau_j} \left(y_i(s,a) - \theta^j_\psi(s,a) \right)$
    $\psi \leftarrow \psi - \lambda_Z \hat{\nabla}_\psi J_Z(\psi)$,
    $\bar{\psi} \leftarrow (1-\beta)\bar{\psi} + \beta\psi$
  end for
end for
return critic $Z_\psi, Z_{\bar{\psi}}$.

4.3.3 Mean-standard deviation Maximum Entropy RL for continuous action space

We use a Distributional Maximum Entropy framework for continuous action space which is closed to the TQC algorithm Kuznetsov et al. (2020) which uses an actor-critic framework with a distributional truncated critic to ovoid overestimation in the estimation with the max operator. This algorithm is based on a soft-policy iteration where we penalize the target using the entropy of the distribution. More formally, to compute the target, the principle is to train N approximate estimate $Z_{\psi_1}, \dots Z_{\psi_C}$ of the distribution of returns Z^π where Z_{ψ_c} maps each (s,a) to $Z_{\psi_c}(s,a) := \frac{1}{M}\sum_{m=1}^{M} \delta\left(\theta^m_{\psi_n}(s,a)\right)$, which is supported on atoms $\theta^1_{\psi_c}(s,a), \dots, \theta^M_{\psi_c}(s,a)$. Then approximations $Z_{\psi_1}, \dots Z_{\psi_N}$ are trained on the temporal difference target distribution denoted $Y(s,a)$ contructed as follow. First atoms are pooled into a distributions $Z_{\psi_1}(s',a'), \dots, Z_{\psi_C}(s',a')$ into $\mathcal{Z}(s',a') := \left\{\theta^m_{\psi_c}(s',a') \mid c \in [1..C], m \in [1..M]\right\}$ and denote elements of $\mathcal{Z}(s',a')$ sorted in ascending order by $z_{(i)}(s',a')$, with $i \in [1..MC]$. Then we only keep the kC smallest elements of $\mathcal{Z}(s',a')$. We remove outliers of distribution to avoir overestimation of the value function. Finally the atoms of the target distribution $Y(s,a) := \frac{1}{kC}\sum_{i=1}^{kC} \delta\left(y_i(s,a)\right)$ are computed according to a soft policy gradient method where we penalised with the log of the policy :

$$y_i(s,a) := r(s,a) + \gamma\left[z_{(i)}(s',a') - \eta \log \pi_\phi\left(a' \mid s'\right)\right]. \tag{4.3}$$

As in QR-DQN, the 1-Wasserstein distance between each of $Z_{\psi_n}(s,a), n \in [1..N]$ and the temporal difference target distribution $Y(s,a)$ is minimized learning the locations for quantile fractions $\tau_m = \frac{2m-1}{2M}, m \in [1..M]$. Similary, we minimize the loss :

$$J_Z(\psi_c) = \mathbb{E}_{\mathcal{D},\pi}\left[\mathcal{L}^k(s_t,a_t;\psi_c)\right] = \mathbb{E}_{\mathcal{D},\pi}\left[\frac{1}{MkC}\sum_{j=1}^{M}\sum_{i=1}^{kC}\rho^H_{\tau_j}\left(y_i(s,a) - \theta^j_{\psi_c}(s,a)\right)\right] \tag{4.4}$$

over the parameters ψ_n, for each critic. The learning of all quantiles $\theta^m_{\psi_n}(s,a)$ is with this formulation dependent on all atoms of the truncated mixture of target distributions. To optimize the actor, the following loss based on KL-divergence denoted $\mathrm{D_{KL}}$ is used for soft policy improvement, where η can be seen as a temperature and needs to be tuned:

$$J_{\pi,\alpha}(\phi) = \mathbb{E}_{\mathcal{D}}\left[\mathrm{D_{KL}}\left(\pi_\phi(\cdot \mid s) \,\|\, \frac{\exp\left(\frac{1}{\eta}\xi_\alpha(\theta_\psi(s,\cdot))\right)}{D}\right)\right]$$

where D is a constant of normalisation. This expression simplify into :

$$J_{\pi,\alpha}(\phi) = \mathbb{E}_{\mathcal{D},\pi}\left[\eta \log \pi_\phi(a \mid s) - \frac{1}{C}\sum_{c=1}^{C}\xi_\alpha(\theta_{\psi_c}(s,a))\right] \tag{4.5}$$

where $s \sim \mathcal{D}, a \sim \pi_\phi(\cdot \mid s)$. Nontruncated estimate of the Q-value are used for policy optimization to avoid a double truncation, in fact the Z-functions approximate already truncated future distribution. Finally, η is the entropy temperature coefficient is dynamically adjusted by taking a gradient step with respect to the loss like in Haarnoja et al. (2018b) :

$$J(\eta) = \mathbb{E}_{\mathcal{D},\pi_\phi}\left[\left(-\log \pi_\phi(a_t \mid s_t) - \mathcal{H}_\eta\right)\eta\right]$$

at every time the π_ϕ changes. Temperature η decrease if the policy entropy, $-\log \pi_\phi(a_t \mid s_t)$, is higher than $\mathcal{H}_\eta$ and increases η otherwise. The algorithm is summarized as follow :

Algorithm 5: TQC with Standard Deviation penalisation

Initialize policy π_ϕ, critics $Z_{\psi_c}, Z_{\bar{\psi}_c}$ for $c \in [1..C]$
for each iteration **do**
  for each step of the environment **do**
    collect (s_t, a_t, r_t, s_{t+1}) with policy π_ϕ
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, r_t, s_{t+1})\}$
  end for
  for each gradient steps **do**
    Sample batch (s, a, s', r) of $\mathcal{D}$
    $y_i(s,a) \leftarrow r(s,a) + \gamma \left[z_{(i)}\left(s', a'\right) - \eta \log \pi_\phi\left(a' \mid s'\right) \right]$
    $\eta \leftarrow \eta - \lambda_\eta \hat{\nabla}_\eta J(\eta)$
    $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_{\pi,\alpha}(\phi)$
    $\psi_c \leftarrow \psi_c - \lambda_Z \hat{\nabla}_{\psi_c} J_Z\left(\psi_n\right), c \in [1..C]$
    $\bar{\psi}_c \leftarrow \beta \psi_c + (1-\beta)\bar{\psi}_c, n \in [1..C]$
  end for
end for
return π_ϕ, critics $Z_{\psi_c}, n \in [1..C]$.

Our algorithm is based SAC framework but with many distributional critics to improve estimation of Q-functions while using mean-standard deviation objective in the policy loss to improve robustness.

4.4 Experiments

We try different experiments on continuous and discrete action space to demonstrate the interest of our algorithms for robustness using $\xi : Z \to \mathbb{E}[Z] - \alpha^{1/2}\mathbb{V}[Z]^{\frac{1}{2}}$ instead of the mean. The choice of α is crucial as it determines the degree of penalty in the objective. The more the environment is penalized, the more a pessimistic action is chosen.

4.4.1 Results on continuous action spaces

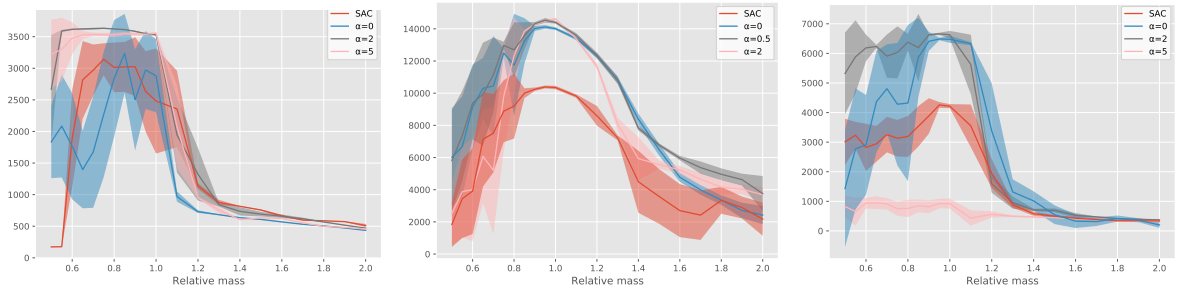
For continuous action space, we compare our algorithm with SAC which achieves state-of-the-art robust control (Eysenbach and Levine 2021) on the Mujoco environment such as Hopper-v3, Walker-v3 or HalfCheetah-v3. We use a version where the entropy coefficient is adjusted during learning for both SAC and our algorithm, as it requires less parameter tuning. Moreover, we show the influence of a distributional critic without a mean-standard deviation greedy step using $\alpha = 0$ to demonstrate the advantage of using a distributional critic against the classical SAC algorithm. We also compare our results to TQC algorithm, varying the penalty α to show that for the tested environment, there exists a value of α such that prediction are more robust to change of dynamics.

The interest of our algorithm is best shown in stochastic environments, since it involves the distributions of returns which are varying in stochastic environments. The only source of stochasticity in the Mujoco subject is the initial point, so in order to make its environments stochastic we have noised environments at each step by adding a noise in $[-1e^{-2}, 1e^{-2}]$ to each action. Since we also compare our algorithm in non-stochastic environments, we differentiate the two cases by denoting noisy environments by **(N)** and environments without noise **(wN)**. In these simulations, variations of dynamics are carried out by moving the relative mass, which is an

influential physical parameter in all environments. All algorithms are trained with a relative mass of 1 and then tested on new environment where the mass varies from 0.5 to 2. Two phenomena can be observed for the 3 environments.

First, for all environments in Fig 4.1,4.2, and Fig A12.2 in annex, where performance is normalized by the maximum of the performance for every curve to highlight robustness and not only mean-performance. We see that we can find a value of α where the robustness is clearly improved without deteriorating the average performance. In fact, if a penalty is applied too strongly, the average performance can be reduced, as in the HalfCheetah-v3 environment. For Hopper-v3, a α calibrated at 5 gives very good robustness performances, while for Walker2d-v3, the value is closer to 2. This phenomenon was expected and was in agreement with our formulation. Moreover, our algorithm outperforms the SAC algorithm for Robustness tasks in all environments. Tuning of α must be chosen carefully, for example, α is chosen in $\{0, 1, ..., 5\}$ for Hopper-v3 and Walker2d-v3 whereas values of α are chosen smaller in $\{0, 0.1, 0.5.1, 1.5, 2\}$ and not in a bigger interval. As a rule of thumb for choosing α, we can look at the empirical mean and variance at the end of the trajectories to see if the environment has rewards that fluctuate a lot. The smaller the mean/variance ratio, the more likely we are to penalise our environment. For HalfCheetah, the mean/variance ratio is about approximately 100, so we will favour smaller penalties than for Walker2d where the mean/variance ratio is about 50 or 10 for Hopper.

The second surprising observation is that penalizing our objective also improves performance in terms of stability during training and in terms of average performance, especially for Hopper and Walker2d in Fig 4.4 or sometimes in Fig 4.3. Similar results are present in the work of (Moskovitz et al. 2021), which gives an interpretation in terms of optimism and pessimism for environments. This phenomenon is not yet explained, but it is present in environments that are particularly unstable and have a lot of variance. The variance of the return is a consequence of the stochasticity of the environment or of the policy. Intuitively, the most favorable settings are thus the one with the most stochasticity. We have, however, observed that our method remains interesting in low-stochasticity or non-stochasticity environments even if the policy is not stochastic. A possible explanation is a better exploration thanks to the pessimistic approach.

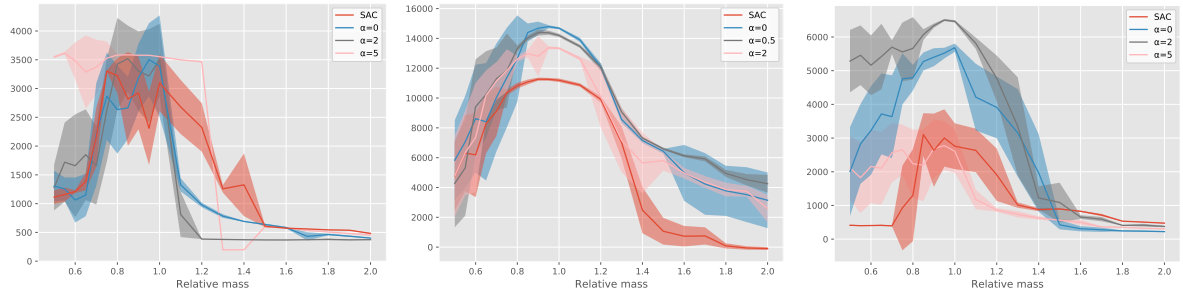


(a) Hopper-v3 **(N)** **(b)** HalfCheetah-v3 **(N)** **(c)** Walker-v3 **(N)**

Figure 4.1: y-axis : normalised mean ± standard deviation over 20 trajectories. x-axis : relative mass.

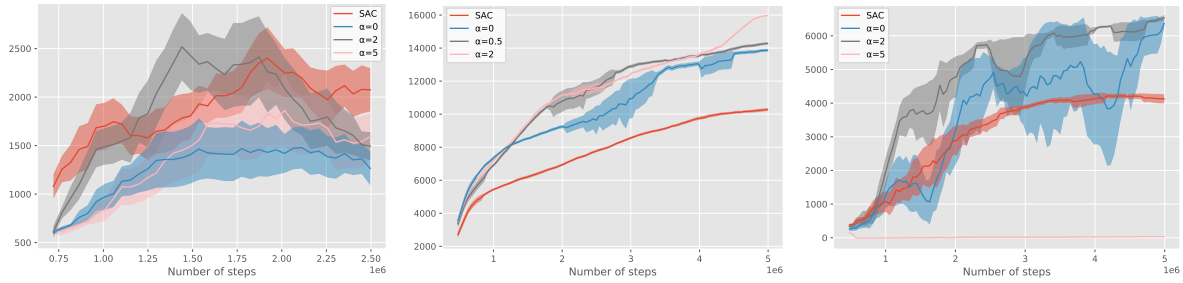
4.4.2 Results on discrete action spaces

We test our QRDQN algorithm with standard deviation penalization on discrete action space, varying the length of the pole in Cartpole-v1 and Acrobot-v1 environments. We observe similar results for the discrete environment in terms of robustness. Training is done for a length of the pole equal to the x-axis of the black star on the graph, and then for testing, the length of the pole is increased or decreased. We show that robustness is increased when we penalised our distributional critic. We have compared our algorithm to PPO which has shown relatively good



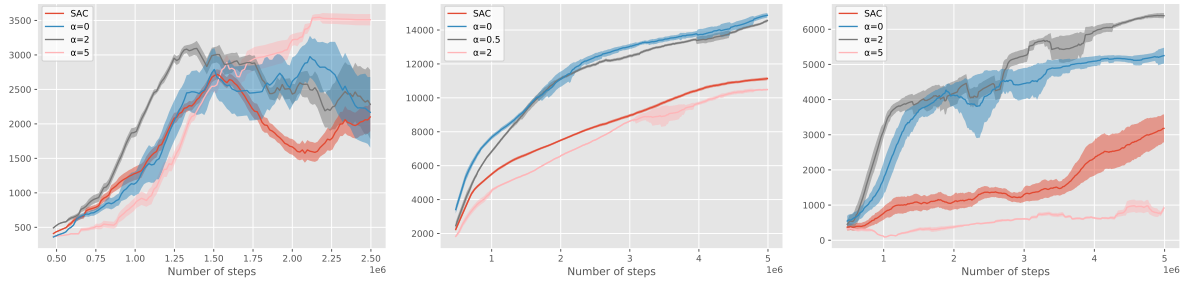
(a) Hopper-v3 **(wN)** **(b)** HalfCheetah-v3 **(wN)** **(c)** Walker-v3 **(wN)**

Figure 4.2: y-axis : mean ± standard deviation over 20 test trajectories. x-axis: relative mass.



(a) Hopper-v3 **(N)** **(b)** HalfCheetah-v3 **(N)** **(c)** Walker-v3 **(N)**

Figure 4.3: y-axis : mean over 20 trajectories ± standard deviation in function of timesteps.



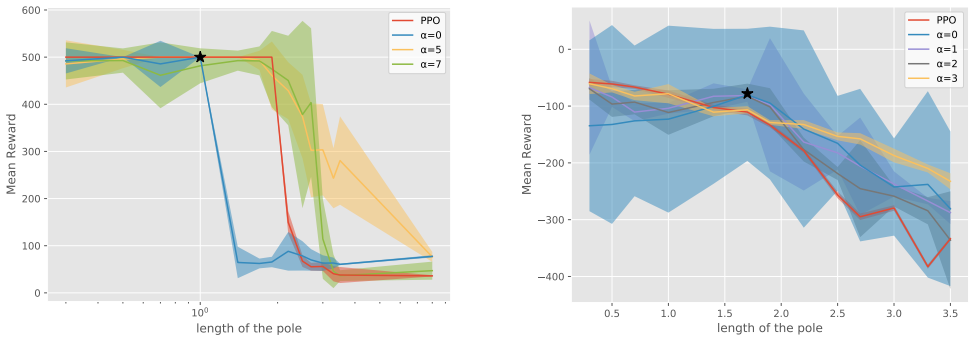
(a) Hopper-v3 **(wN)** **(b)** HalfCheetah-v3 **(wN)** **(c)** Walker-v3 **(wN)**

Figure 4.4: y-axis : mean over 20 trajectories ± standard deviation in function of timesteps.

results in terms of robustness for discrete action space in (Abdullah et al. 2019) as SAC does not apply to discrete action space. The same phenomenon is observed in terms of robustness as for continuous environments. However, the improvement in terms of mean performance on Hopper and Walker2d environments is not observed. This is partly explained by the fact that the maximum reward is reached in Cartpole and Acrobot quickly. An ablation study can be found in annex C where we study the impact of penalization on our behavior policy during testing and on the policy used during learning. It is shown that both are needed in the algorithm.

4.5 Conclusion of Chapter 4

In this Chapter, we have tried to show that by using a mean-standard deviation formulation to choose our actions pessimistically, we can increase the robustness of our environment for continuous and discrete environments without adding too much the complexity. A single fixed α parameter must be tuned to obtain good performance without penalizing the average performance



(a) CartPole-v1 **(penal A)** **(b)** Acrobot-v1 **(penal A)**

Figure 4.5: Mean over 20 trajectories varying length's pole, trained on the x-axis of the black star.

too much. Moreover, for some environments, it is relevant to penalize to increase the average performance as well when there is many variability in the environment.

About limitations of this work : Convergence of the algorithm to a fix point is not shown using only for mean-standard deviation penalisation in the greedy step. In fact there is no policy improvement theorem with this formulation. Moreover it may be difficult to tune α in practice. In the next Chapter, we will try to deal with these problems, deriving a Deep Robust formulation with theoretical guarantees, avoiding the problem of estimation of a penalisation and trying to find more interpretable uncertainty parameter α which can be easily tuned.

Chapter 5

Boostraping Expectile in Reinforcement Learning

Contents

Pessimism is a desirable concept in many Reinforcement Learning (RL) algorithms to stabilize the learning and get an accurate estimation of the value function. This idea is developed in Double Q-learning (Hasselt 2010), an RL technique designed to address the issue of overestimation bias in value estimation, a common challenge in Q-learning and related algorithms. Overestimation bias occurs when the estimated values of actions are higher than their true values, potentially leading to a suboptimal policy. By maintaining two sets of Q-values and decoupling action selection from value estimation, Double Q-learning provides a more accurate and less optimistic estimate of the true values of actions. In general, Double Q-learning enhances the stability of the learning process and these principles can be extended to deep RL known as Double Deep Q-Networks (DDQN), a successful approach in various applications (Van Hasselt et al. 2016). Pessimism also appears in the twin critic approach, the equivalent of Double Q-learning for continuous action spaces, which requires training two critics to select the most pessimistic one. Many state-of-the-art RL algorithms are based on this method, such as TD3 (Fujimoto et al. 2018) that uses this method to improve on DDPG (Lillicrap et al. 2015) and SAC (Haarnoja et al. 2018a) that uses this trick to stabilize the learning of Q-functions and policies.

The idea of pessimism is also central in Robust RL (Moos et al. 2022), where the agent tries to find the best policy under the worst transition kernel in a certain uncertainty space. It has been introduced first theoretically in the context of Robust MDPs (Iyengar 2005, Nilim and El Ghaoui 2005) (RMDPs) where the transition probability varies in an uncertainty (or ambiguity) set.

Hence, the solution of robust MDPs is less sensitive to model estimation errors with a properly chosen uncertainty set, as RMDPs are formulated as a max-min problem, where the objective is to find the policy that maximizes the value function for the worst possible model that lies within an uncertainty set around a nominal model. Fortunately, many structural properties of MDPs are preserved in RMDPs (Iyengar 2005), and methods such as robust value iteration, robust modified policy iteration, or partial robust policy iteration (Ho et al. 2021) can be used to solve them. It is also known that the uncertainty in the reward can be easily tackled while handling uncertainty in the transition kernel is much more difficult (Kumar et al. 2022, Derman et al. 2021). Finally, the sample complexity of RMDPs has been studied theoretically (Yang et al. 2021, Shi and Chi 2022, Clavier et al. 2023, Shi et al. 2023). However, these works usually assume having access to a generative model.

Robust RL (Moos et al. 2022) tries to bridge a gap with real-life problems, classifying its algorithms into two distinct groups. The first group engages solely with the nominal kernel or the center of the uncertainty set. To enhance robustness, these algorithms often adopt an equivalent risk-averse formulation to instill pessimism. For instance, Clavier et al. (2022) employ mean-standard deviation optimization through Distributional Learning to bolster robustness. Another strategy involves introducing perturbations on actions during the learning process, as demonstrated by Tessler et al. (2019), aiming to fortify robustness during testing. Another method, known as adversarial kernel robust RL (Wang et al. 2023), exclusively samples from the nominal kernel and employs resampling techniques to simulate the adversarial kernel. While this approach introduces a novel paradigm, it also leads to challenges associated with poor sample complexity due to resampling and requiring access to a generative model. Despite this drawback, the adversarial kernel robust RL paradigm offers an intriguing avenue for exploration and development in the realm of robust RL. Finally, policy gradient (Kumar et al. 2023, Li et al. 2023) in the case of Robust MDPs is also an alternative. A practical algorithm using robust policy gradient with Wasserstein metric is proposed by Abdullah et al. (2019), but this approach requires having access to model parameters which are usually not available in a model-free setting. The second category of algorithms engages with samples within the uncertainty set, leveraging available information to enhance the robustness and generalization of policies to diverse environments. Algorithms within this category, such as IWOCS (Zouitine et al. 2023), M2TD3 (Tanabe et al. 2022b), M3DDPG (Li et al. 2019a), and RARL (Pinto et al. 2017) actively interact with various close environments to fortify robustness in the context of RL.

In all these settings, the idea of pessimism is central. We propose here a new simple form of pessimism based on expectile estimates that can be plugged into any RL algorithm. For a given algorithm, the only modification relies on the critic loss in an actor-critic framework or in the Q-learning loss for Q-function based algorithms. Given a target $y(r, s') = r + \gamma Q_{\phi,\text{targ}}(s', \pi(s'))$ with reward r, policy π, we propose to minimize

$$L(\phi, \mathcal{D}) = \mathop{\mathbb{E}}_{(s,a,r,s')\sim\mathcal{D}} \left[L_2^\alpha \left(Q_\phi(s,a) - y(r,s') \right) \right],$$

where L_2^τ is the expectile loss defined in Section 5.2.3. For $\alpha = 1/2$, the expectile coincides with the classical mean, and we recover the classical L_2 loss of most RL algorithms. We denote this modification as `ExpectRL`. In many RL algorithms, we are bootstrapping the expectation of the Q-function over the next state, by definition of the classical Bellman equation.

Our method `ExpectRL` is equivalent to bootstrapping the expectile and not the expectation of the Q value. Bootstraping expectiles still leads to an algorithm with the contraction mapping property for the associated Expectile Bellman Operator, but adds pessimism by giving more weight to the pessimistic next state compared to a classical expectation (see Section 5.2.3).

The `ExpectRL` modification is relevant in the context of the twin critic approach as when employing this method, the challenge arises in effectively regulating the level of pessimism

through the application of the twin critic method, which remains heuristic for continuous action spaces, although it has been studied in the discrete case by Hasselt (2010). Furthermore, the acquisition of imprecise Q functions has the potential to yield detrimental outcomes in practical applications, introducing the risk of catastrophic consequences. Using the `ExpectRL` method, the degree of pessimism in learning the value or Q function is controlled through the parameter α, and our first question is:

Can we replace the learning of two critics in the twin critic method, using only a simple expectile bootstrapping?

In the Robust RL setting, `ExpectRL` can also be beneficial as by nature expectiles are a coherent, convex risk measure, that can be written as a minimum of an expectation over probability measure on a close convex set (Delbaen 2002). So implicitly bootstrapping an expectile instead of an average leads to a robust RL algorithm. Compared to many Robust RL algorithms, our method is simple in the sense that the α-expectile is more interpretable and easy to choose than a penalization or trade-off parameter in mean-standard deviation optimization (Clavier et al. 2022). `ExpectRL` has the advantage of being computationally simple compared to other methods, as it uses all samples, compared to the work of Wang et al. (2023), that needs resampling to induce robustness. Finally, our method is simple and can be adapted to practical algorithms, compared to robust policy gradient methods such as Kumar et al. (2023), Li et al. (2023). Moreover, while these algorithms can be considered more mathematically grounded and less heuristic, the second group with IWOCS, M2DTD2, RARL (Zouitine et al. 2023, Tanabe et al. 2022b, Li et al. 2019a, Pinto et al. 2017) tends to rely on heuristic approaches that exhibit practical efficacy on real-world benchmarks. This dichotomy prompts the question:

Can we leverage `ExpectRL` method as a surrogate for Robust RL and formulate robust RL algorithms that are both mathematically founded and requiring minimal parameter tuning?

By extending expectile bootstrapping (`ExpectRL`) with sampling from the entire uncertainty set using domain randomization (DR), our approach bolsters robustness, positioning itself competitively against the best-performing algorithms. Notably, our algorithm incurs low computational costs relatively to other algorithms and requires minimal or no hyperparameter tuning. Our contributions are the following.

Our **first contribution**, is to introduce `ExpectRL`, and demonstrate the efficacy of that method as a viable alternative to the twin critic trick with L_2 loss across diverse environments. This substitution helps empirically control of the overestimation in the Q-function, thereby reducing the computational burden associated with the conventional application of the twin trick, which entails learning two critics.

The **second contribution** of our work lies in establishing that expectile bootstrapping or `ExpectRL` facilitates the development of straightforward Deep Robust RL approaches. These approaches exhibit enhanced robustness compared to classical RL algorithms. The effectiveness of our approach combining `ExpectRL` with DR is demonstrated on various benchmarks and results in an algorithm that closely approaches the state of the art in robust RL, offering advantages such as lower computational costs and minimal hyperparameters to fine-tune.

Our **third contribution** introduces an algorithm, `AutoExpectRL` that leverages an automatic mechanism for selecting the expectile or determining the degree of pessimism. Leveraging bandit algorithms, this approach provides an automated and adaptive way to fine-tune the expectile

parameter, contributing to the overall efficiency and effectiveness of the algorithm.

5.1 Related Work

TD3 and twin critics. To tackle the problem of over-estimation of the value function, TD3 algorithm (Fujimoto et al. 2018) algorithm uses two critics. Defining the target y_{min} as $y_{min}(r, s') = r + \gamma \min_{i=1,2} Q_{\phi_{i,\text{targ}}}(s', \pi(s'))$, both critics are learned by regressing to this target, such that, for $i \in \{1, 2\}$,

$$L(\phi_i, \mathcal{D}) = \mathop{\mathbb{E}}_{(s,a,r,s',d)\sim\mathcal{D}} \left(Q_{\phi_i}(s, a) - y_{min}(r, s')\right)^2 .$$

Our approach is different as we do not consider the classic L_2 loss and only use one critic. We will compare `ExpectRL` to the classic TD3 algorithm both with twin critics and one critic to understand the influence of our method.

Expectiles in Distributional RL. Expectiles have found application within the domain of Distributional RL (RL), as evidenced by studies such as (Rowland et al. 2019, Dabney et al. 2018b, Jullien et al. 2023). It is crucial to note a distinction in our approach, where we specifically focus on learning a single expectile to substitute the conventional L_2 norm. This diverges from the methodology adopted in these referenced papers, where the entire distribution is learned using different expectiles. Moreover, they do not consider expectile statistics on the same random variable as they consider expectiles of the full return.

Expectile in Offline RL and the IQL algorithm . Implicit Q-learning (IQL) (Kostrikov et al. 2021) in the context of offline RL endeavors to enhance policies without the necessity of evaluating actions that have not been encountered. Like our method, IQL employs a distinctive approach by treating the state value function as a random variable associated with the action, but achieves an estimation of the optimal action values for a state by utilizing a state conditional upper expectile. In `ExpectRL`, we employ lower expectiles to instill pessimism on the next state and approximate a minimum function, contrasting with the conventional use of upper expectiles for approximating the maximum in the Bellman optimality equation.

Risk-Averse RL. Risk-averse RL, as explored in studies like Pan et al. (2019), diverges from the traditional risk-neutral RL paradigm. Its objective is to optimize a risk measure associated with the return random variable, rather than focusing solely on its expectation. Within this framework, Mean-Variance Policy Iteration has been considered for optimization, as evidenced by Zhang et al. (2021), and Conditional Value at Risk (CVaR), as studied by Greenberg et al. (2022). The link between Robust and Risk averse MDPS has been highlighted by Chow et al. (2015) and Zhang et al. (2023) who provide a mathematical foundation for risk-averse RL methodologies, emphasizing the significance of coherent risk measures in achieving robust and reliable policies. Our method lies in risk-averse RL as expectiles are a coherent risk measure (Zhang et al. 2023), but to the best of our knowledge, the expectile statistic has never been considered before for tackling robust RL problems.

Regularisation and robustness in RL. Regularization plays a pivotal role in the context of Markov Decision Processes (MDPs), as underscored by Derman et al. (2021) or Eysenbach and Levine (2021), who have elucidated the pronounced connection between robust MDPs and their regularized counterparts. Specifically, they have illustrated that a regularised policy during

interaction with a given MDP exhibits robustness within an uncertainty set surrounding the MDP in question. In this work, we focus on the idea that generalization, regularization, and robustness are strongly linked in RL or MDPs as shown by Husain et al. (2021), Derman and Mannor (2020), Derman et al. (2021), Ying et al. (2021), Brekelmans et al. (2022). The main drawback of this method is that it requires tuning the introduced penalization to improve robustness, which is not easy in practice as it is very task-dependent. The magnitude of the penalization is not always interpretable compared to α, the value of the expectile.

5.2 Background

5.2.1 Markov Decision Processes

We first define Robust Markov Decision Processes (MDPs) as $\mathcal{M}_\Omega = \{\mathcal{M}_\omega\}_{\omega\in\Omega}$, with $\mathcal{M}_\omega = \{\mathcal{S}, \mathcal{A}, P_\omega, P^0_\omega, r_\omega, \gamma\}$ the MDP with specific uncertainty parameter $\omega \in \Omega$. The chosen state space S and action space A are subsets of real-valued vector spaces in our setting. The transition probability density $P_\omega : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, the initial state probability density $P^0_\omega : \mathcal{S} \to \mathbb{R}$, and the immediate reward $r_\omega : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ depend on ω. Moreover, we define $P_{s,a,w}$ the vector of $P_\omega(s, a, .)$. The discount factor is denoted by $\gamma \in (0, 1)$. Let $\pi_\theta : S \to A$ be a policy parameterized by $\theta \in \Theta$ and π^* the optimal policy. Given an uncertainty parameter $\omega \in \Omega$, the initial state follows $s_0 \sim P^0_\omega$. At each time step $t \geqslant 0$, the agent observes state s_t, selects action $a_t = \pi_\theta(s_t)$, interacts with the environment, and observes the next state $s_{t+1} \sim P_\omega(\cdot \mid s_t, a_t)$, and the immediate reward $r_t = r_\omega(s_t, a_t)$. The discounted return of the trajectory starting from time step t is $R_t = \sum_{k\geqslant 0} \gamma^k r_{t+k}$. The action value function $Q^{\pi_\theta}(s, a, \omega)$ and optimal action value $Q^*(s, a, \omega)$ under ω is the expectation of R_t starting with $s_t = s$ and $a_t = a$ under ω; that is,

$$Q^{\pi_\theta,P}(s, a, \omega) = \mathbb{E}_{P_\omega,\pi_\theta}[R_t \mid s_t = s, a_t = a], \quad Q^{*,P}(s, a, \omega) = \mathbb{E}_{P_\omega,\pi^*}[R_t \mid s_t = s, a_t = a],$$

where $\mathbb{E}$ is the expectation. Note that we introduce ω to the argument to explain the Q-value dependence on ω. Lastly, we define the value function as

$$V^{\pi_\theta,P}(s, \omega) = \mathbb{E}_{P_\omega,\pi_\theta}[R_t \mid s_t = s], \quad V^{*,P}(s, \omega) = \mathbb{E}_{P_\omega,\pi^*}[R_t \mid s_t = s].$$

In the following, we will drop the ω subscript for simplicity and define the expectile (optimal) value function, that follows the recursive Bellman equation

$$V^{\pi,P}(s) = v^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}[\underbrace{r(s, a) + \gamma\mathbb{E}_{P_{sa}}[V^{\pi,P}]}_{\triangleq Q^{\pi,P}s,a)}], \tag{5.1}$$

$$V^{*,P}(s) = \max_{a\in\mathcal{A}}(\underbrace{r(s, a) + \gamma\mathbb{E}_{P_{sa}}[V^{*,P}]}_{\triangleq Q^{*,P}(s,a)}). \tag{5.2}$$

Finally, we define the classical Bellman Operator and optimal Bellman Operator that are γ-contractions, so iteration of these operators leads $V^{\pi,P}$ or $V^{*,P}$:

$$\mathcal{T}^{\pi,P}V(s) := \sum_a \pi(a|s)(r(s, a) + \gamma\mathbb{E}_{P_{sa}}[V]) \tag{5.3}$$

$$\mathcal{T}^{*,P}V(s) := \mathcal{T}^{\pi^*,P}V(s) = \max_a(r(s, a) + \gamma\mathbb{E}_{P_{sa}}[V]). \tag{5.4}$$

5.2.2 Robust MDPs

Once classical MDPs are defined, we can define robust (optimal) Bellman operators $\mathcal{T}^\pi_\mathcal{U}$ and $\mathcal{T}^*_\mathcal{U}$,

$$\mathcal{T}^\pi_\mathcal{U}V(s) := \min_{P\in\mathcal{U}} \mathcal{T}^{\pi,P}V(s), \mathcal{T}^*_\mathcal{U}V(s) := \max_{\pi\in\Delta(A)} \min_{P\in\mathcal{U}} \mathcal{T}^{\pi,P}V(s), \tag{5.5}$$

where P belongs to the uncertainty set $\mathcal{U}$. The optimal robust Bellman operator $T^*_{\mathcal{U}}$ and robust Bellman operator $T^\pi_{\mathcal{U}}$ are γ-contraction maps for any policy π (Iyengar 2005, Thm. 3.2) if the uncertainty set $\mathcal{U}$ is a subset of $\Delta(S)$ where $\Delta(S)$ it the simplex of $|S|$ elements so that the transition kernel is valid. Finally, for any initial values V^π_0, V^*_0, sequences defined as $V^\pi_{n+1} := T^\pi_{\mathcal{U}} V^\pi_n$ and $V^*_{n+1} := T^*_{\mathcal{U}} V^*_n$ converge linearly to their respective fixed points, that is $V^\pi_n \to V^\pi_{\mathcal{U}}$ and $V^*_n \to V^*_{\mathcal{U}}$.

5.2.3 Expectiles

Let's first define expectiles. For $\alpha \in (0,1)$ and X a random variable, the α-expectile is defined as $m_\alpha(X) = \arg\min_m \mathbb{E}_x[L^\alpha_2(x-m)]$ with

$$L^\alpha_2(u) = |\alpha - \mathbb{1}_{\{u<0\}}|u^2 = \alpha u^2_+ + (1-\alpha)u^2_-,$$

where $u_+ = \max(u,0)$ and $u_- = \max(-u,0)$. We can recover the classical mean with $m_{\frac{1}{2}}(X) = \mathbb{E}[X]$ as $L^{1/2}_2(u) = u^2$ .Expectiles are gaining interest in statistics and finance as they induce the only law-invariant, coherent (Artzner et al. 1999) and elicitable (Gneiting 2011) risk measure. Using the coherent property representation (Delbaen 2000), one has that $\rho : L^\infty \to \mathbb{R}$ is a coherent risk measure if and only if there exists a closed convex set $\mathcal{P}$ of P-absolutely continuous probability measures such that $\rho(X) = \inf_{Q\in\mathcal{P}} \mathbb{E}_Q[X], \forall X \in L^\infty$. with L^∞ the vector space of essentially bounded measurable functions with the essential supremum norm. The uncertainty set induced by expectiles as been described by Delbaen (2013) as $m_\alpha(X) = \min_{Q\in\mathcal{E}} \mathbb{E}_Q[X]$ such as

$$\mathcal{E} = \left\{ Q \in \mathcal{P} \mid \ \exists \eta > 0, \eta\sqrt{\frac{\alpha}{1-\alpha}} \le \frac{dQ}{dP} \le \sqrt{\frac{(1-\alpha)}{\alpha}}\eta \right\} \tag{5.6}$$

where we define $\frac{dQ}{dP}$ as the Radon-Nikodym derivative of Q with respect to P. Here, the uncertainty set corresponds thus to a lower and upper bound on $\frac{dQ}{dP}$ with a quantity depending on the degree of uncertainty. For $\alpha = 1/2$, the uncertainty set becomes the null set and we retrieve the classical mean. This variational form of the expectile will be useful to link risk-sensitive and robust MDPs formulation in the next section.

5.3 `ExpectRL` method

First, we introduce the Expectile Bellman Operator and then we will explain our proposed method `ExpectRL` and `AutoExpectRL` that work both in classic and robust cases.

5.3.1 Expectile Bellman Operator

In this section, we derive the loss and explain our approach. Recall that for $\alpha \in (0,1)$ and X a random variable taking value x and following a probability law P, the α-expectile is denoted $m_\alpha(X)$ or $m_\alpha(P,x)$ in the following. Writing the classical Bellman operator for q function

$$\mathcal{T}^{\pi,P}Q(s,a) = r(s,a) + \gamma\langle P_{sa}, v\rangle = r(s,a) + \gamma\mathbb{E}_{s'\sim P_{sa}(\cdot)}[V(s')].$$

and denoting $\mathbf{V}_{sa}$ the random variable which is equal to $V(s')$ with probability $P_{sa}(s')$, it holds that:

$$\mathcal{T}^{\pi,P}Q(s,a) = r(s,a) + \gamma m_{\frac{1}{2}}(\mathbf{V}_{sa}) = r(s,a) + \gamma m_{\frac{1}{2}}(P_{sa}, V).$$

Our method consists instead in considering the following Expectile Bellman operator

$$\mathcal{T}_\alpha Q(s,a) = r(s,a) + \gamma m_\alpha(\mathbf{V}_{sa}). \tag{5.7}$$

With $\alpha < \frac{1}{2}$, Eq. (5.7) allows to learn a robust policy, in the sense that it is a pessimistic estimate about the value we bootstrap according to the value sampled according to the nominal kernel. Next, we define the expectile value of a given policy and the optimal expectile value as:

$$V_\alpha^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}[\underbrace{r(s,a) + \gamma m_\alpha(P_{sa}, V_\alpha^\pi)}_{\triangleq q_\alpha^\pi(s,a)}], \tag{5.8}$$

$$V_\alpha^*(s) = \max_{a\in\mathcal{A}}(\underbrace{r(s,a) + \gamma m_\alpha(P_{sa}, V_\alpha^*)}_{\triangleq q_\alpha^*(s,a)}). \tag{5.9}$$

With $\alpha = \frac{1}{2}$, we retrieve the standard Bellman equations but we consider $\alpha < \frac{1}{2}$ for the robust case. Finally, we define (optimal) expectile Bellman Operator as:

$$T_\alpha^\pi V(s) = \sum_a \pi(a|s)(r(s,a) + \gamma m_\alpha(P_{sa}, V)).$$
$$T_\alpha^* V(s) = \max_a (r(s,a) + \gamma m_\alpha(P_{sa}, V)).$$

Theorem 5.3.1. *The (optimal) Expectile Bellman Operators are γ-contractions for the sup norm. (proof in Appx. 16).*

So as T_α^π and T_α^* are γ-contractions, it justifies the definition of fixed point V_α^π and V_α^*. The central idea to show that expectile bootstrapping or `ExpectRL` is implicitly equivalent to Robust RL comes (Zhang et al. 2023) where we try to estimate the optimal robust value function $V_\mathcal{E}^* = \max_\pi \min_{Q\in\mathcal{E}} V^{\pi,Q}$.

Theorem 5.3.2. *The (optimal) Expectile value function is equal to the (optimal) robust value function*

$$V_\alpha^*(s) = V_\mathcal{E}^\pi := \max_\pi \min_{Q\in\mathcal{E}} V^{\pi,Q}, \quad V_\alpha^\pi(s) = V_\mathcal{E}^\pi := \min_{Q\in\mathcal{E}} V^{\pi,Q} \tag{5.10}$$

where $\mathcal{E}$ is defined in 5.2.3. Proof can be found in 16.2. Note that his formulation does not converge to the expectile of the value distribution but to $V_\mathcal{E}^*$ the robust value function. Moreoever, for $\alpha > 1/2$, Now that expectile operators are defined, we will define the related loss.

5.3.2 The `ExpecRL` Loss

In this section, we present the method more from a computational and practical point of view. As stated before, this method can be plugged into any RL algorithm where a Q-function is estimated, which included any Q-function-based algorithm or some actor-critic framework during the critic learning. For a given algorithm, the only modification relies on modifying the L_2 loss in the Q-value step by the Expectile loss. Given a target $y(r,s') = r + \gamma Q_{\phi,\text{targ}}(s', \pi(s'))$ with reward r, policy π, we propose to minimize

$$L(\phi, \mathcal{D}) = \mathop{\mathbb{E}}_{(s,a,r,s')\sim\mathcal{D}} \left[L_2^\alpha\left(Q_\phi(s,a) - y(r,s')\right)\right], \tag{5.11}$$

where L_2^α is the expectile loss defined in Section 5.2.3. For $\alpha = 1/2$, the expectile coincides with the classical mean, and we retrieve the classical L_2 loss present in most RL algorithms. We will use TD3 as a baseline and replace the learning of the critic with this loss. The actor loss remains the same in the learning process. With `ExpectRL`, only one critic is needed, replacing the double critic present in this algorithm. We will compare our method with the classical TD3 algorithm using the twin critic trick and TD3 with one critic to see the influence of our method.

5.3.3 `ExpecRL` method with Domain randomisation

From a practical point of view, many Robust RL algorithms such as M2TD3 (Tanabe et al. 2022b), M3DDPG (Li et al. 2019a), and RARL (Pinto et al. 2017) not only interact with the nominal environment but also with environments that belong to the uncertainty set $\mathcal{U}$. Sampling trajectories from the entire uncertainty set allows algorithms to get knowledge from dangerous trajectories and allows algorithms to generalize better than algorithms that only sample from the nominal. Receiving information about all environments that need to be robust during the training phase, the algorithm tends to obtain better performance on minimum performance over these environments on testing. With the same idea of generalization, Domain Randomisation (DR) (Tobin et al. 2017) focuses not on the worst case under the uncertainty set but on the expectation. Given a point of the uncertainty set $P_\omega \in \mathcal{U}$, the DR objective is: $\pi^*_{\text{DR}} = \arg\max_\pi \mathbb{E}_{\omega\in\Omega, s\sim P^0_\omega}[V^\pi(s,w)]$. In other words, DR tries to find the best policy on average over all environments in the uncertainty set. The approach we propose to be competitive on a robust benchmark is to find the best policy using `ExpectRL` under domain randomization or

$$\pi^*_{\text{DR},\alpha} = \arg\max_\pi \mathbb{E}_{\omega\in\Omega, s\sim P^0_\omega}[V^\pi_\alpha(s,\omega)] = \arg\max_\pi \mathbb{E}_{\omega\in\Omega, s\sim P^0_\omega}[\min_{P_\omega\in\mathcal{E}} V^{\pi,P}(s,\omega)], \tag{5.12}$$

where $V^\pi_\alpha(s,\omega)$ is the expectile value function under uncertainty kernel P_ω and $\mathcal{E}$ defined in Section 5.2.3. Using this approach, we hope to get sufficient information from all the environments using DR and improve robustness and worse-case performance using `ExpectRL`. The advantage of the approach is that any algorithm can be used for learning the policy, sampling from the entire uncertainty set uniformly and replacing the critic loss of this algorithm learning with `ExpecRl` loss. The effectiveness of this algorithm on a Robust benchmark will be conducted in Section 5.3. Getting an algorithm that is mathematically founded and which tries to get the worst-case performance, the last question is how to choose the degree of pessimism or $\alpha \in (0, 1/2)$ in practice. The following section tries to answer this question using a bandit algorithm to auto-tune α.

5.3.4 Auto-tuning of the expectile α using bandit

In the context of varying levels of uncertainty across environments, the selection of an appropriate expectile α becomes contingent on the specific characteristics of each environment. To automate the process of choosing the optimal expectile, we employ a bandit algorithm, specifically the Exponentially Weighted Average Forecasting algorithm (Cesa-Bianchi and Lugosi 2006). We denote this method as `AutoExpectRL`. This formulation adopts the multi-armed bandit problem, where each bandit arm corresponds to a distinct value of α. We consider a set of D expectiles making predictions from a discrete set of values $\{\alpha_d\}_{d=1}^D$. At each episode m, a cumulative reward R_m is sampled, and a distribution over arms $\mathbf{p}_m \in \Delta_D$ is formed, where $\mathbf{p}_m(d) \propto \exp(w_m(d))$. The feedback signal $f_m \in \mathbb{R}$ is determined based on the arm selection as the improvement in performance, specifically $f_m = R_m - R_{m-1}$, where R_m denotes the cumulative reward obtained in the episode m. Then, w_{m+1} is obtained from w_m by modifying only the d_m according to $w_{m+1}(d_m) = w_m(d_m) + \eta\frac{f_m}{\mathbf{p}_m(d)}$ where $\eta > 0$ is a step size parameter. The exponential weights distribution over α values at episode m is denoted as $\mathbf{p}^\alpha_m$. This approach can be seen as a form of model selection akin to the methodology presented by Pacchiano et al. (2020). Notably, instead of training distinct critics and actors for each α choice, our approach updates one single neural network for the critic and one single neural network for the actor. In both critic and actor, neural networks are composed of one common body and different heads for every value of α, in our case 4 values for $\{\alpha_d\}_{d=1}^D = \{0.2, 0.3, 0.4, 0.5\}$. The critic's heads correspond to the 4 expectile losses for different values of α. The actor's neural network is trained using 4 classical TD3 losses, evaluated with action chosen by one specific head of the actor. Then in both critic and actor, the 4 losses are summed, allowing an update of all heads at each iteration. Finally, the sampling of new trajectories

	TD3 Twin Critic	TD3 1 critic	`ExpectRL` best Expectile	`AutoExpectRL`
$Ant(\times 10^3)$	3.65 ± 0.33	1.90 ± 0.07	$\mathbf{4.46 \pm 0.12}$	4.27 ± 0.25
$HalfCheetah(\times 10^3)$	$\mathbf{10.91} \pm 0.14$	10.36 ± 0.54	10.42 ± 0.13	10.40 ± 0.09
$Hopper(\times 10^3)$	2.88 ± 0.10	2.022 ± 0.09	$\mathbf{3.10 \pm 0.05}$	3.03 ± 0.11
$Walker(\times 10^3)$	2.95 ± 0.12	2.35 ± 0.25	$\mathbf{3.22 \pm 0.11}$	3.02 ± 0.09
$HumanoidStandup(\times 10^5)$	1.101 ± 0.09	1.087 ± 0.09	$\mathbf{1.197 \pm 0.05}$	1.143 ± 0.010

Table 5.1: Expectile vs Twin-critic, Mean performance $\pm$ standard error, on 10 train seed

is done using the chosen head of the actor, proposed by the bandit algorithm. More details about implementation can be found in Appendix 17. Intuitively, when the agent receives a higher reward compared to the previous trajectory, the probability of choosing this arm is increased to encourage this arm to be picked again. Note that the use of a bandit algorithm to automatically select hyperparameters in an RL algorithm has been proposed in other contexts, such as Moskovitz et al. (2021), Badia et al. (2020). The `AutoExpectRL` method allows picking automatically expectile α and reduces hyperparameter tuning. Practical details can be found in Appendix 17 where we expose the neural network architecture of this problem and associated losses. Note that this approach does not work in the DR setting as uncertainty parameters change between trajectories in DR. It is difficult for the algorithm to know if high or low rewards on trajectories come because the uncertainty parameter leads to small rewards, or if it is due to bad expectile picked at this iteration.

5.4 Empirical Result on Mujoco

The Mujoco benchmark is employed in this experiment due to its significance for evaluating robustness in the context of continuous environments, where physical parameters may vary. In contrast, the Atari benchmark very deterministic with discrete action space without physical parameters cannot change during the testing period. In this section, we compare the performance of the TD3 algorithm using the twin critic method during learning, only one critic, and finally our method `ExpectRL`. The different values of α are $\{\alpha_d\}_{d=1}^{D} = \{0.2, 0.3, 0.4, 0.5\}$. We can notice that `ExpectRL` with $\alpha = 0.5$ is exactly TD3 with one critic. Here, we only interact with the nominal and there is no notion of robustness. The mean and standard deviation are reported in Table 5.1, where we use 10 seeds of 3M steps for training, each evaluated on 30 trajectories. The last column is our last algorithm, `AutoExpectRL`. In all environments except HalfCheetah, `ExpectRL` with fine-tuning of α has the best score and `AutoExpectRL` has generally close results. The scores for every expectiles can be found in Appendix19. In Halcheetah 5.1 environment, it seems that no pessimism about Q-function is needed and our method `ExpectRL` is outperformed by TD3 with twin critic. Similar observations have been observed in Moskovitz et al. (2021) on this environment. Moreover, results for $\alpha = 0.5$ and $\alpha = 0.4$ are very close in Appendix 19 while the variance is reduced using $\alpha = 0.4$. Results of Table 5.1 show that it is possible to replace the twin critic approach with only one critic with the relevant value of pessimism or expectile. Moreover, one can remark in Appendix 19 that in Hopper, Walker, and Ant environment, high pessimism is needed to get an accurate Q function and better results, with a value of $\alpha = 0.2$ or $\alpha = 0.3$ whereas less pessimism with $\alpha = 0.4$ is needed for HumanoidStandup and HalfCheetah. Note that the value of $\alpha = 0.5$ is never chosen and leads to generally the worst performance as reported in column TD3 with

	TD3 mean	ExpectRL mean	Auto mean	TD3 worst	ExpectRL worst	Auto worst
*Ant*1	2.76 ± 0.5	3.55 ± 0.65	$\mathbf{3.55 \pm 0.51}$	2.22 ± 0.5	2.65 ± 0.57	$\mathbf{2.71 \pm 0.43}$
*Ant*2	2.28 ± 0.09	$\mathbf{2.50 \pm 0.89}$	2.41 ± 0.77	1.59 ± 0.08	$\mathbf{2.49 \pm 0.94}$	2.42 ± 0.51
*Ant*3	0.31 ± 1.13	$\mathbf{0.54 \pm 0.08}$	0.53 ± 0.69	-0.99 ± 1.13	-0.94 ± 0.21	$\mathbf{-0.88 \pm 0.34}$
*Half*1	2.79 ± 0.22	$\mathbf{3.05 \pm 0.48}$	2.98 ± 0.19	-0.34 ± 0.04	$\mathbf{-0.27 \pm 0.19}$	-0.27 ± 0.21
*Half*2	$\mathbf{2.63 \pm 0.20}$	2.51 ± 0.41	2.58 ± 0.32	-0.53 ± 0.06	$\mathbf{-0.223 \pm 0.16}$	-0.23 ± 0.10
*Half*3	$\mathbf{2.47 \pm 0.18}$	2.45 ± 0.42	2.39 ± 0.15	-0.61 ± 0.08	$\mathbf{-0.557 \pm 0.27}$	-0.58 ± 0.09
*Hopper*1	2.39 ± 0.14	$\mathbf{2.76 \pm 0.04}$	2.52 ± 0.11	0.4 ± 0.02	0.44 ± 0.01	$\mathbf{0.449 \pm 0.15}$
*Hopper*2	1.54 ± 0.17	$\mathbf{2.06 \pm 0.01}$	1.87 ± 0.02	0.21 ± 0.04	$\mathbf{0.32 \pm 0.03}$	0.32 ± 0.03
*Hopper*3	1.15 ± 0.14	$\mathbf{1.43 \pm 0.02}$	1.433 ± 0.09	0.14 ± 0.03	$\mathbf{0.25 \pm 0.22}$	2.42 ± 0.19
*Walker*1	3.12 ± 0.2	$\mathbf{3.66 \pm 0.68}$	3.58 ± 0.27	0.68 ± 0.12	$\mathbf{2.77 \pm 0.15}$	1.99 ± 0.13
*Walker*2	2.70 ± 0.2	$\mathbf{3.98} \pm 0.58$	3.88 ± 0.61	0.28 ± 0.07	$\mathbf{1.36 \pm 0.82}$	1.11 ± 0.15
*Walker*3	2.60 ± 0.18	$\mathbf{3.84 \pm 0.45}$	3.58 ± 0.15	0.17 ± 0.06	0.65 ± 0.12	$\mathbf{0.87 \pm 0.09}$
*Humanoid*1	1.03 ± 0.4	1.12 ± 0.25	$\mathbf{1.13 \pm 0.26}$	0.85 ± 0.07	$\mathbf{0.97 \pm 0.23}$	0.98 ± 0.24
*Humanoid*2	1.03 ± 0.3	$\mathbf{1.13 \pm 0.15}$	1.11 ± 0.12	0.73 ± 0.07	$\mathbf{0.83 \pm 0.23}$	0.80 ± 0.18
*Humanoid*3	1.01 ± 0.3	$\mathbf{1.06 \pm 0.13}$	1.05 ± 0.18	0.57 ± 0.04	$\mathbf{0.71 \pm 0.21}$	0.68 ± 0.09

Table 5.2: Result on Robust Benchmark for TD3 `ExpectRL` and `AutoExpectRL`. Results are $\times 10^3$ bigger for all environments except for Humanoid where results are $\times 10^5$ bigger.

one critic which coincides with $\alpha = 0.5$. Finally, the variance is also decreased using our method compared to TD3 with twin critics or TD3 with one critic. Finally, our method `AutoExectRL` allows choosing automatically the expectile almost without loss of performance and outperforming TD3, except on the environment HalfCheetah. Learning curves can be found in Appendix 19.

5.5 Empirical Results on Robust Benchmark

This section presents an assessment of the worst-case and average performance and generalization capabilities of the proposed algorithm. The experimental validation was conducted on optimal control problems utilizing the MuJoCo simulation environments (Todorov et al. 2012). The performance of the algorithm was systematically benchmarked against state-of-the-art robust RL M2TD3 as it is state of the art compared to other algorithms methodologies, M3DDPG, and RARL. Furthermore, a comparative analysis was undertaken with Domain Randomization (DR) as introduced by Tobin et al. (2017) for a comprehensive evaluation. To assess the worst-case performance of the policy π under varying uncertainty parameters $\omega \in \Omega$, following the benchmark of Tanabe et al. (2022b), 30 evaluations of the cumulative reward were conducted for each uncertainty parameter value $\omega_1, \ldots, \omega_K \in \Omega$. Specifically, $R_k(\pi)$ denotes the cumulative reward on ω_k, averaged over 30 trials. Subsequently, $R_{\text{worst}}(\pi) = \min_{1 \leqslant k \leqslant K} R_k(\pi)$ (denoted (w) in Table 5.2 and 5.3) was computed as an estimate of the worst-case performance of π on Ω. Additionally, the average performance was computed as $R_{\text{average}}(\pi) = \frac{1}{K} \sum_{k=1}^{K} R_k(\pi)$ (denoted (m) in Table 5.2 and 5.3). For the evaluation process, K uncertainty parameters $\omega_1, \ldots, \omega_K$ were chosen according to the dimensionality of ω: for 1D ω, $K = 10$ equally spaced points on the 1D interval Ω; for 2D ω, 10 equally spaced points were chosen in each dimension of Ω, resulting in $K = 100$ points; and for 3D ω, 10 equally spaced points were selected in each dimension of Ω, resulting in $K = 1000$ points or different environments. Each approach underwent policy training 10 times in each environment. The training time steps $T_{\max}$ were configured as 2M, 4M, and 5M for scenarios with 1D, 2D, and 3D uncertainty parameters respectively, following Tanabe

	DR+ExpectRL(m)	M2TD3(m)	DR(m)	DR+ExpectRL(w)	M2TD3(w)	DR(w)
$Ant1$	4.84 ± 0.43	4.51 ± 0.08	$\mathbf{5.25 \pm 0.1}$	3.36 ± 0.55	$\mathbf{3.84 \pm 0.1}$	3.51 ± 0.08
$Ant2$	5.63 ± 0.43	5.44 ± 0.05	$\mathbf{6.32 \pm 0.09}$	2.72 ± 0.42	$\mathbf{4.13 \pm 0.11}$	1.64 ± 0.13
$Ant3$	2.86 ± 1.03	2.66 ± 0.22	$\mathbf{3.62 \pm 0.11}$	$\mathbf{0.28 \pm 0.35}$	0.10 ± 0.10	-0.32 ± 0.03
$Half1$	5.3 ± 0.59	3.89 ± 0.06	$\mathbf{5.93 \pm 0.18}$	2.86 ± 0.99	3.14 ± 0.10	$\mathbf{3.19 \pm 0.08}$
$Half2$	5.25 ± 0.32	4.35 ± 0.05	$\mathbf{5.79 \pm 0.15}$	1.77 ± 0.31	$\mathbf{2.61 \pm 0.16}$	2.12 ± 0.13
$Half3$	4.52 ± 0.24	3.79 ± 0.09	$\mathbf{5.54 \pm 0.16}$	1.02 ± 0.24	0.93 ± 0.21	$\mathbf{1.09 \pm 0.06}$
$Hopper1$	2.58 ± 0.23	$\mathbf{2.68 \pm 0.11}$	2.57 ± 0.15	$\mathbf{0.64 \pm 0.20}$	0.62 ± 0.45	0.53 ± 0.26
$Hopper2$	$\mathbf{2.53 \pm 0.22}$	2.51 ± 0.07	1.89 ± 0.08	$\mathbf{0.55 \pm 0.07}$	0.53 ± 0.28	0.47 ± 0.02
$Hopper3$	$\mathbf{2.21 \pm 0.33}$	0.85 ± 0.07	1.5 ± 0.07	$\mathbf{0.39 \pm 0.07}$	0.28 ± 0.25	0.21 ± 0.03
$Walker1$	$\mathbf{3.77 \pm 0.89}$	3.70 ± 0.31	3.59 ± 0.26	$\mathbf{3.41 \pm 0.05}$	2.83 ± 0.39	2.19 ± 0.42
$Walker2$	$\mathbf{4.75 \pm 0.57}$	4.72 ± 0.12	4.54 ± 0.31	2.74 ± 0.61	$\mathbf{3.14 \pm 0.39}$	2.31 ± 0.51
$Walker3$	4.39 ± 0.37	4.27 ± 0.21	$\mathbf{4.48 \pm 0.16}$	1.14 ± 0.79	$\mathbf{1.34 \pm 0.43}$	1.32 ± 0.34
$Humanoid1$	$\mathbf{1.21 \pm 0.23}$	1.08 ± 0.04	1.12 ± 0.05	$\mathbf{1.04 \pm 0.86}$	0.93 ± 0.07	0.96 ± 0.06
$Humanoid2$	$\mathbf{1.23 \pm 0.22}$	0.97 ± 0.04	1.06 ± 0.04	$\mathbf{0.86 \pm 0.28}$	0.65 ± 0.07	0.73 ± 0.78
$Humanoid3$	$\mathbf{1.12 \pm 0.35}$	1.09 ± 0.06	1.04 ± 0.07	$\mathbf{0.84 \pm 0.26}$	0.62 ± 0.06	0.54 ± 0.34

Table 5.3: Result on Robust Benchmark for `ExpectRL` + DR , M2TD3 and DR. Results are $\times 10^3$ bigger for all environments except for Humanoid results are $\times 10^5$ bigger. The mean performance is denoted (m) and worst case (w).

et al. (2022b). Table 9.7 summarizes the different changes of parameters in the environments. The final policies obtained from training were then evaluated for their worst-case performances and average performance over all uncertainty parameters. The results are the following.

We first demonstrate that our method `ExpectRl` is more robust than the classical RL algorithm. To do so, we conduct the benchmark task presented previously on TD3 algorithm (with twin critic trick) as a baseline and our method `ExpectRl`. As exposed in Table 5.2, our method outperforms TD3 in all environments on worst-case performance, which was expected as TD3 is not designed by nature to be robust and to maximize a worst-case performance. Moreover, `AutoExpectRL` has good and similar performance compared to the best expectile like in Table 5.1. As TD3 has sometimes very bad performance, our method also performs better on average over all environments except HalfCheetah 2 and HalfCheetah 3. These two environments required more exploration, and pessimism is in general not a good thing for these tasks. Moreover, robustness is not needed in HalfCheetah environments that are already quite stable compared to other tasks in Mujoco. However, `ExpectRL` needs to be compared with algorithms designed to be robust, such as M2TD3 which has state-of-the-art performance on this benchmark.

If performance of `ExpectRL` in Table 5.2 and the performance of M2TD3 in Table 5.3 are compared, we can observe a large difference on many tasks where M2TD3 outperforms, in general, our method. This is because sampling trajectories from the entire uncertainty set allows M2TD3 to get knowledge from dangerous trajectories and allows the algorithm to generalize better than our method, which only samples from the nominal. The comparison between methods is then not fair for `ExpectRl` which has only access to samples from the nominal and this is why the method `ExpectRL + DR` was introduced. Receiving information about all environments that need to be robust during the training phase, the algorithm tends to obtain better performance on minimum performance over these environments on testing. Table 5.3 shows the result on average and on worst-case performance between our second method `ExpectRL` + DR with tuning of α against

M2TD3 and DR approach. Recall that `AutoExpectRL` cannot be used with DR as mentioned at the end of Section 5.3.4.

In terms of worst-case performance, our method outperforms 9 times M2TD3 (8 times in bold and one time when DR is better in general for HalfCheetaht3) and has a worse performance on 6 tasks compared to M2TD3. Our method is therefore competitive with the state of the art in robust algorithms such as M2TD3, which already outperformed M3DDPG and RARL on worst-case performance. Except on Hopper1, our method outperforms M2TD3 on average, results which show that M2TD3 is very pessimistic compared to our method. However, in terms of average results, we can see that DR, which is designed to be good on average across all environments, generally performs better than our method and M2TD3 expect on Hopper, Walker1 and 2, and HumanoidStandup which are not stable and need to be robustified to avoid catastrophic performance that affect too much the mean performance over all environment. Moreover, compared to M2TD3, our method `ExpecRL`, even without auto fine-tuning of α, has the advantage of having fewer parameter tuning compared to the M2TD3 algorithm.

5.6 Conclusion and perspectives

We propose a simple method, `ExpectRL` to replace twin critic in practice, only replacing the classic L_2 loss of the critic with an expectile loss. Moreover, we show that it can also lead to a Robust RL algorithm and demonstrate the effectiveness of our method combined with DR on a robust RL Benchmark. The limitations of our method are that `AutoExpectRL` allows fine-tuning of α only without combining with DR. About future perspectives, we demonstrate the effectiveness of our method using as baselines TD3, but our method can be easily adapted to any algorithm using a Q-function such as classical DQN, SAC, and other algorithms both with discrete or continuous action space. Finally, theoretically, it would be interesting to study for example sample complexity of this method compared to the classical RL algorithm. Finally, in this Chapter, the agent try to find the best policy against an adversary that can easily pick the worst kernel in the uncertainty set without any continuity between chosen adversarial transition kernel. In the following Chapter we will try to relax this assumptions to reduce the influence of the adversary and get better results, which allow a tradoff between robustness and performances in RL

Chapter 6

Time-Constrained Robust MDPs

Contents

6.1 Introduction

Robust MDPs capture the problem of finding a control policy for a dynamical system whose transition kernel is only known to belong to a defined uncertainty set. The most common framework for analyzing and deriving algorithms for robust MDPs is that of *sa*-rectangularity (Iyengar 2005, Nilim and El Ghaoui 2005), where probability measures on outcome states are picked independently in different source states and actions (in formal notation, $\mathbb{P}(s'|s,a)$ and $\mathbb{P}(s'|\bar{s},\bar{a})$ are independent of each other). This provides an appreciable decoupling of worst transition kernel search across time steps and enables sound algorithms like robust value iteration (RVI). But policies obtained for such *sa*-rectangular MDPs are by nature very conservative (Goyal and Grand-Clement 2018, Li et al. 2023), as they enable drastic changes in environment properties from one time step to the next, and the algorithms derived from RVI tend to yield very conservative policies even when applied to non-*sa*-rectangular robust MDP problems.

In this paper, we depart from the rectangularity assumption and turn towards a family of robust MDPs whose transition kernels are parameterized by a vector ψ. This parameter vector couples together the outcome probabilities in different (s,a) pairs, hence breaking the independence assumption that is problematic, especially in large dimension Goyal and Grand-Clement (2018). This enables accounting for the notion of transition model consistency across states and actions: outcome probabilities are not picked independently anymore but are rather set across the state and action spaces by drawing a parameter vector. In turn, we examine algorithms for solving such parameter-based robust MDPs when the parameter is constrained to follow a bounded evolution throughout time steps. Our contributions are the following.

1. We introduce a formal definition for parametric robust MDPs and time-constrained robust MDPs, discuss their properties and derive a generic algorithmic framework (Sec. 6.2).

2. We propose three algorithmic variants for solving time-constrained MDPs, named vanilla `TC` , `Stacked-TC` and `Oracle-TC` (Sec. 6.4), which use different levels of information in the state space, and come with theoretical guaranties (Sec. 6.6).

3. These algorithms are extensively evaluated in MuJoCo (Todorov et al. 2012) benchmarks, demonstrating they lead to non-conservative and robust policies (Sec. 6.5).

6.2 Problem statement

(Robust) MDPs. A Markov Decision Process (MDP) (Puterman 2014) is a model of a discrete-time, sequential decision making task. At each time step, from a state $s_t \in S$ of the MDP, an action $a_t \in A$ is taken and the state changes to s_{t+1} according to a stationary Markov transition kernel $P(s_{t+1}|s_t, a_t)$, while concurrently receiving a reward $r(s_t, a_t)$. S and A are measurable sets and we write $\Delta(S)$ and $\Delta(A)$ the set of corresponding probability distributions. A stationary policy $\pi(\cdot|s)$ is a mapping from states to distributions over actions, prescribing which action should be taken in s. The value function $V^{\pi,P}$ of policy π maps state s to the expected discounted sum of rewards $\mathbb{E}_{P,\pi}[\sum_t \gamma^t r_t]$ when applying π from s for an infinite number of steps. An optimal policy for an MDP is one whose value function is maximal in any state. In a Robust MDP (RMDP) (Iyengar 2005, Nilim and El Ghaoui 2005), the transition kernel P is not set exactly and can be picked in an adversarial manner at each time step, from an uncertainty set $\mathcal{P}$. Then, the pessimistic value function of a policy is $V^{\pi}_{\mathcal{P}}(s) = \min_{P\in\mathcal{P}} V^{\pi,P}(s)$. An optimal robust policy is one that has the largest possible pessimistic value function $V^*_{\mathcal{P}}$ in any state, hence yielding an adversarial $\max_\pi \min_P$ optimization problem. Robust Value Iteration (RVI) (Iyengar 2005, Wiesemann et al. 2013) solves this problem by iteratively computing the one-step lookahead best pessimistic value:

$$V_{n+1}(s) = T^*_{\mathcal{P}} V_n(s) := \max_{\pi(s)\in\Delta(A)} \min_{P\in\mathcal{P}} \mathbb{E}_{a\sim\pi(s)}[r(s,a) + \mathbb{E}_P[V_n(s')]].$$

The $T^*_{\mathcal{P}}$ operator is called the robust Bellman operator and the sequence of v_n functions converges to the robust value function $v^*_{\mathcal{P}}$ as long as the adversarial transition kernel belongs to the simplex of $\Delta(S)$.

Zero-sum Markov Games. Robust MDPs can be cast as zero-sum two-players Markov games (Littman 1994, Tessler et al. 2019) where B is the action set of the adversarial player. Writing $\bar{\pi} : \mathcal{S} \times \mathcal{A} \to \Delta_B$ the policy of this adversary, the robust MDP problem turns to $\max_\pi \min_{\bar{\pi}} V^{\pi,\bar{\pi}}$, where $v^{\pi,\bar{\pi}}(s)$ is the expected sum of discounted rewards obtained when playing π (agent actions) against $\bar{\pi}$ (transition models) at each time step from s. This enables introducing the robust value iteration sequence of functions

$$V_{n+1}(s) := \mathcal{T}^{**} V_n(s) := \max_{\pi(s)\in\Delta(A)} \min_{\bar{\pi}(s,a)\in\Delta(S)} \mathcal{T}^{\pi,\bar{\pi}} V_n(s)$$

where $\mathcal{T}^{\pi,\bar{\pi}} := \mathbb{E}_{a\sim\pi(s)}[r(s,a) + \gamma \mathbb{E}_{s'\sim\bar{\pi}(s,a)} V_n(s')]$ is a zero-sum Markov game operator. These operators are also $\gamma-$contractions and converge to their respective fixed point $V^{\pi,\bar{\pi}}$ and $V^{**} = V^*_{\mathcal{P}}$ Tessler et al. (2019). This formulation will be useful to derive a practical algorithm in Section 6.4.

Often, this convergence is analyzed under the assumption of sa-rectangularity, stating that the uncertainty set $\mathcal{P}$ is a set product of independent subsets of $\Delta(S)$ in each s, a pair. Quoting Iyengar (2005), rectangularity is a sort of independence assumption and is a minimal requirement for most theoretical results to hold. Within robust value iteration, rectangularity enables picking $\bar{\pi}(s_t, a_t)$ completely independently of $\bar{\pi}(s_{t-1}, a_{t-1})$. To set ideas, let us consider the robust MDP of a pendulum, described by its mass and rod length. Varying this mass and rod length spans the

uncertainty set of transition models. The rectangularity assumption induces that $\bar{\pi}(s_t, a_t)$ can pick a measure in $\Delta(S)$ corresponding to a mass and a length that are completely independent from the ones picked in the previous time step. While this might be a good representation in some cases, in general it yields policies that are very conservative as they optimize for adversarial configurations which might not occur in practice.

We first step away from the rectangularity assumption and define a parametric robust MDP as an RMDP whose transition kernels are spanned by varying a parameter vector ψ (typically the mass and rod length in the previous example). Choosing such a vector couples together the probability measures on successor states from two distinct (s, a) and (s', a') pairs. The main current robust deep RL algorithms actually optimize policies for such parametric robust MDPs but still allow the parameter value at each time step to be picked independently of the previous time step.

Parametric MDPs. A parametric RMDP is given by the tuple $(\mathcal{S}, \mathcal{A}, \Psi, P_\psi, r)$ where the transition kernel $P_\psi(s, a) \in \Delta(S)$ is parameterized by ψ, and Ψ is the set of values ψ can take, equipped with an appropriate metric. This yields the robust value iteration update :

$$V_{n+1}(s) = \max_{\pi(s)\in\Delta(A)} \min_{\psi\in\Psi} \mathcal{T}^\pi_\psi V_n(s) := \max_{\pi(s)\in\Delta(A)} \min_{\psi\in\Psi} \mathbb{E}_{a\sim\pi(s)}[r(s, a) + \gamma \mathbb{E}_{s'\sim P_\psi(s,a)} V_n(s')].$$

A parametric RMDP remains a Markov game and the Bellman operator remains a contraction mapping as long as P_ψ can reach only elements in the simplex of $\Delta(S)$, where the adversary's action set is the set of parameters instead of a (possibly *sa*-rectangular) set of transition kernels.

Time-constrained RMDPs (TC-RMDPs). We introduce TC-RMDPs as the family of parametric RMDPs whose parameter's evolution is constrained to be Lipschitz with respect to time. More formally a TC-RMDP is given by the tuple $(\mathcal{S}, \mathcal{A}, \Psi, P_\psi, r, L)$, where $\|\psi_{t+1} - \psi_t\| \leq L$, that is the parameter change is bounded through time. In the previous pendulum example, this might represent the wear of the rod which might lose mass or stretch length. Similarly, and for a larger scale illustration, TC-RMDPs enable representing the possible evolutions of traffic conditions in a path planning problem through a busy town. Starting from an initial parameter value ψ_{-1}, the pessimistic value function of a policy π is non-stationary, as ψ_0 is constrained to lay at most L-far away from ψ_{-1}, ψ_1 from ψ_0, and so on. Generally, this yields non-stationary value functions as the uncertainty set at each time step depends on the previous uncertainty parameter. To regain stationarity without changing the TC-RMDP definition, we first change the definition of the adversary's action set. The adversary picks its actions in the constant set $\mathcal{B} = \mathtt{B}(0_\Psi, L)$, which is the ball of radius L centered in the null element in Ψ. In turn, the state of the Markov game becomes the pair s, ψ and the Markov game itself is given by the tuple $((S \times \Psi), \mathcal{A}, \mathcal{B}, P_\psi, r)$, where the Lipschitz constant L is included in B. Thus, given an action $b_t \in B$ and a previous parameter value ψ_{t-1}, the parameter value at time t is $\psi_t = \psi_{t-1} + b_t$. Then, we define the pessimistic value function of a policy as a function of both the state s and parameter ψ:

$$V^\pi_{\mathcal{B}}(s, \psi) := \min_{\substack{(b_t)_{t\in\mathbb{N}},\\ b_t\in\mathcal{B}}} \mathbb{E}\Big[\sum \gamma^t r_t | \psi_{-1} = \psi, s_0 = s, b_t \in \mathcal{B}, \psi_t = \psi_{t-1} + b_t, a \sim \pi, s_t \sim P_{\psi_t}\Big],$$

$$V^*_{\mathcal{B}}(s, \psi) = \max_{\pi(s,\psi)\in\Delta(A)} V^\pi_{\mathcal{B}}(s, \psi).$$

In turn, an optimal robust policy is a function of s and ψ and the TC robust Bellman operators are:

$$\begin{aligned} V_{n+1}(s, \psi) &:= \mathcal{T}^*_B v_n(s, \psi) := \max_{\pi(s,\psi)\in\Delta_A} \mathcal{T}^\pi_B V_n(s, \psi), \\ &:= \max_{\pi(s,\psi)\in\Delta_A} \min_{b\in\mathcal{B}} \mathbb{E}_{a\sim\pi(s)}[r(s, a) + \gamma \mathbb{E}_{s'\sim P_{\psi+b}(s,a)} V_n(s', \psi + b)]. \end{aligned}$$

This iteration scheme converges to a fixed point according to Th. 6.2.1.

Theorem 6.2.1. *The time-constrained (TC) Bellman operators $\mathcal{T}_{\mathcal{B}}^{\pi}$ and $\mathcal{T}_{B}^{*}$ are contraction mappings. Thus the sequences $V_{n+1} = \mathcal{T}_{\mathcal{B}}^{\pi} V_n$ and $V_{n+1} = \mathcal{T}_{\mathcal{B}}^{*} v_n$, converge to their respective fixed points $V_{\mathcal{B}}^{\pi}$ and $V_{\mathcal{B}}^{*}$.*

Proof of Th. 6.2.1 can be found in Appendix 25. We refer to this formulation as algorithm `Oracle-TC` (see Section 6.4 for implementation details) since an oracle makes the current parameter ψ visible to the agent. Therefore, it is possible to derive optimal policies for TC-RMDPs by iterated application of this TC Bellman operator. These policies have the form $\pi(s, \psi)$. In the remainder of this paper, we extend state-of-the-art robust deep RL algorithms to the TC-RMDP framework. In particular, we compare their performance and robustness properties with respect to classical robust MDP formulations, we also discuss their relation with the $\pi(s)$ robust policies of classical robust MDPs.

If the agent is unable to observe the state variable ψ, it is not possible to guarantee the existence of a stationary optimal policy of the form $\pi(s)$. Similarly, there is no guarantee of convergence of value functions to a fixed point. Nonetheless, this scenario, in which access to the ψ parameter is not available, is more realistic in practice. It turns the two-player Markov game into a partially observable Markov game, where one can still apply the TC Bellman operator but without these guarantees of convergence. We call vanilla `TC` the repeated application of the TC Bellman operator in this partially observable case. Vanilla `TC` will be tested in practice, and some theoretical properties of the objective function will be derived using the Lipschitz properties (Sec 6.6).

6.3 Related works

Since our method is a non-rectangular, Deep Robust RL algorithm, (possibly non-stationary for `Stacked-TC` and `TC`), we discuss the following related work.

Non-stationary MDPs. First, non-stationarity has been studied in the Bandits setting in Garivier and Moulines (2008). Then, for episodic, non-stationary MDPs Even-Dar et al. (2004), Abbasi Yadkori et al. (2013), Lecarpentier and Rachelson (2019) have explored and provided regret bounds for algorithms that use oracle access to the current reward and transition functions. More recently Gajane et al. (2018), Cheung et al. (2019) have facilitated oracle access by performing a count-based estimation of the reward and transition functions based on the recent history of interactions. Finally, for tabular MDPs, past data from a non-stationary MDP can be used to construct a full Bayesian model Jong and Stone (2005) or a maximum likelihood model Ornik and Topcu (2019) of the transition dynamics. We focus on the setting not restricted to tabular representations.

Non-rectangular RMDPs. While rectangularity in practice is very conservative, it can be demonstrated that, in an asymptotic sense, non-rectangular ellipsoidal uncertainty sets around the maximum likelihood estimator of the transition kernel constitute the smallest possible confidence sets for the ground truth transition kernel, as implied by classical Cramér-Rao bounds. This is in accordance with the findings presented in § 5 and Appendix A of Wiesemann et al. (2013). More recently, Goyal and Grand-Clement (2018) extends the rectangular assumptions using a factored uncertainty model, where all transition probabilities depend on a small number of underlying factors denoted $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_r \in \mathbb{R}^{\mathsf{S}}$, such that each transition probability P_{sa} for every (s, a) is a linear (convex) combination of these r factors. Finally, Li et al. (2023) use policy gradient algorithms for non-rectangular robust MDPs. While this work presents nice theoretical guarantees of convergence, there is no practical Deep RL algorithms for learning optimal robust policies.

Deep Robust RL Methods. Many Deep Robust algorithms exist such as M2TD3 Tanabe et al. (2022a), M3DDPG Li et al. (2019a), or RARL Pinto et al. (2017), which are all based on the two player zero-sum game presented in 6.2. We will compare our method against these algorithms, except Li et al. (2019a) which is outperformed by Tanabe et al. (2022a) in general. We also compare our algorithm to Domain randomization (DR) Tobin et al. (2017) that learns a value function $V(s) = \max_\pi \mathbb{E}_{p\sim\mathcal{U}(\mathcal{P})} V_p^\pi(s)$ which maximizes the expected return on average across a fixed (generally uniform) distribution on $\mathcal{P}$. As such, DR approaches do not optimize the worst-case performance but still have good performance on average. Nonetheless, DR has been used convincingly in applications Mehta et al. (2020b), Akkaya et al. (2019). Finally, the zero-sum game formulation has lead to the introduction of action robustness Tessler et al. (2019) which is a specific case of rectangular MDPs, in scenarios where the adversary shares the same action space as the agent and interferes with the agent's actions. Several strategies based on this idea have been proposed. One approach, the Game-theoretic Response Approach for Adversarial Defense (GRAD) (Liang et al. 2023) builds on the Probabilistic Action Robust MDP (PR-MDP) (Tessler et al. 2019). This method introduces time-constrained perturbations in both the action and state spaces and employs a game-theoretic approach with a population of adversaries. In contrast to GRAD, where temporal disturbances affect the transition kernel around a nominal kernel, our method is part of a broader setting in which the transition kernel is included in a larger uncertainty set. Robustness via Adversary Populations (RAP) (Vinitsky et al. 2020) introduces a population of adversaries. This approach ensures that the agent develops robustness against a wide range of potential perturbations, rather than just a single one, which helps prevent convergence to suboptimal stationary points. Similarly, State Adversarial MDPs (Zhang et al. 2020; 2021, Stanton et al. 2021, Liang et al. 2023) address adversarial attacks on state observations, effectively creating a partially observable MDP. Finally, using rectangularity assumptions, (Abdullah et al. 2019, Clavier et al. 2022) use Wasserstein and χ^2 balls respectively for the uncertainty set in Robust RL.

6.4 Time-constrained robust MDP algorithms

The TC-RMDP framework addresses the limitations of traditional robust reinforcement learning by considering multifactorial, correlated, and time-dependent disturbances. Traditional robust reinforcement learning often relies on rectangularity assumptions, which are rarely met in real-world scenarios, leading to overly conservative policies. The TC-RMDP framework provides a more accurate reflection of real-world dynamics, moving beyond the conventional rectangularity paradigm.

We cast the TC-RMDP problem as a two-player zero-sum game, where the agent interacts with the environment, and the adversary (nature) changes the MDP parameters ψ. Our approach is generic and can be derived within any robust value iteration scheme, performing $\max_{\pi(s)\in\Delta(A)} \min_{\psi\in\Psi} \mathbb{E}_{a\sim\pi(s)}[r(s,a) + \gamma \mathbb{E}_{s'\sim P_\psi(s,a)} v_n(s')]$ updates, by modifying the adversary's action space and potentially the agent's state space to obtain updates of the form $\max_{\pi(s,\psi)\in\Delta_A} \min_{b\in\mathcal{B}} \mathbb{E}_{a\sim\pi(s)}[r(s,a) + \gamma \mathbb{E}_{s'\sim P_{\psi+b}(s,a)} V_n(s')]$. In Section 6.5, we will introduce time constraints within two specific robust value iteration algorithms, namely RARL Pinto et al. (2017) and M2TD3 Tanabe et al. (2022a) by simply limiting the search space for worst-case ψ at each step. This specific implementation extends the original actor-critic algorithms. For the sake of conciseness, we refer the reader to Appendix 28.1 for details regarding the loss functions and algorithmic details.

Three variations of the algorithm are provided (illustrated in Figure 7.2) but all fall within the training loop of Algorithm 6.

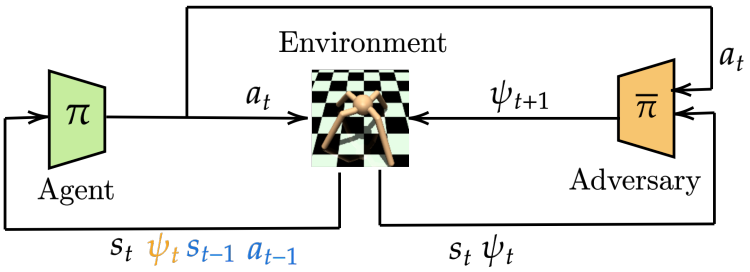


Figure 6.1: TC-RMDP training involves a temporally-constrained adversary aiming to maximize the effect of temporally-coupled perturbations. Conversely, the agent aims to optimize its performance against this time-constrained adversary. In orange, the oracle observation, and in blue the stacked observation.

Algorithm 6: Time-constrained robust training

Input: Time-constrained MDP: $(\mathcal{S}, \mathcal{A}, \Psi, P_\psi, r, L)$, Agent π, Adversary $\bar{\pi}$

for *each interaction time step* t **do**

- $a_t \sim \pi_t(s_t, \psi_t)$ // Sample an action with Oracle-TC
- or $a_t \sim \pi_t(s_t, a_{t-1}, s_{t-1})$ // Sample an action with Stacked-TC
- or $a_t \sim \pi_t(s_t)$ // Sample an action with TC
- $\psi_{t+1} \sim \bar{\pi}_t(s_t, a_t, \psi_t)$ // Sample the worst TC parameter
- $s_{t+1} \sim P_{\psi_{t+1}}(s_t, a_t)$ // Sample a transition
- $\mathcal{B} \leftarrow \mathcal{B} \cup \{(s_t, a_t, r(s_t, a_t), \psi_t, \psi_{t+1}, s_{t+1})\}$ // Add transition to replay buffer
- $\{s_i, a_i, r(s_i, a_i), \psi_i, \psi_{i+1}, s_{i+1}\}_{i \in [1,N]} \sim \mathcal{B}$ // Sample a mini-batch of transitions
- $\pi_{t+1} \leftarrow UpdatePolicy(\pi_t)$ // Update Agent
- $\bar{\pi}_{t+1} \leftarrow UpdatePolicy(\bar{\pi}_t)$ // Update Adversary

Oracle-TC . As discussed in Section 6.2, the Oracle-TC version includes the MDP state and parameter value as input, $\pi : \mathcal{S} \times \Psi \to \mathcal{A}$. This method assumes that the agent has access to the true parameters of the environment, allowing it to make the most informed decisions and possibly reach the true robust value function. However, these parameters ψ are sometimes non-observable in practical scenarios, making this method not always feasible.

Stacked-TC . Since ψ might not be observable but may be approximately identified by the last transitions, the Stacked-TC policy uses the previous state and action as additional inputs in an attempt to replace ψ, $\pi : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathcal{A}$. This approach leverages the information in the transitions, even though it might be insufficient for a perfect estimate of ψ. It aims to retain (approximately) the convergence properties of the Oracle-TC algorithm.

Vanilla TC . Finally, the vanilla TC version takes only the state, $\pi : \mathcal{S} \to \mathcal{A}$, as input, similar to standard robust MDP policies. This method does not attempt to infer the environmental parameters or the transition dynamics explicitly. Instead, it relies on the current state information to guide the agent's actions. While this version is the most straightforward and computationally efficient, it may not perform as robustly as the Oracle-TC or Stacked-TC versions in environments with significant temporal disturbances, since it attempts to solve a partially observable Markov game, for which there may not exist a stationary optimal policy based only on the observation. Despite this, it remains a viable option in scenarios where computational simplicity and quick decision-making are prioritized.

6.5 Results

Experimental settings. This section evaluates the robust time-constrained algorithm's performance under severe time constraints and in the static settings. Experimental validation was conducted in continuous control scenarios using the MuJoCo simulation environments (Todorov et al. 2012). The approach was categorized into three variants. The `Oracle-TC` , where the agent accessed environmental parameters $\pi(s_t, \psi)$; the `Stacked-TC` , where the agent took in input $\pi(s_t, s_{t-1}, a_{t-1})$; and the vanilla `TC` , which did not receive any additional inputs $\pi(s)$. For each variant of the time-constrained algorithms, we applied them to RARL (Pinto et al. 2017), and M2TD3 Tanabe et al. (2022a), renaming them TC-RARL and TC-M2TD3, respectively. The algorithms were tested against two state-of-the-art robust reinforcement learning algorithms, M2TD3 and RARL. Additionally, the Oracle versions of M2TD3 and RARL, where the agent's policy included ψ in the input $\pi : \mathcal{S} \times \Psi \to \mathcal{A}$, were evaluated for a more comprehensive assessment. Comparisons were also made with Domain Randomization (DR) (Tobin et al. 2017) and vanilla TD3. (Fujimoto et al. 2018) to ensure a thorough analysis. A 3D uncertainty set is defined in each environment $\mathcal{P}$ normalized between $[0, 1]^3$. Appendix 30 provides detailed descriptions of uncertainty parameters. Performance metrics were gathered after five million steps to ensure a fair comparison. All baselines were constructed using TD3, and a consistent architecture was maintained across all TD3 variants. The results presented below were obtained by averaging over ten distinct random seeds. Appendices 37.3, 37.2, 37.1, and 35 discuss further details on hyperparameters, network architectures, and implementation choices, including training curves for our methods and baseline comparisons. In the following tables 6.1, 7.6, 7.7, the best performances are shown in bold. Oracle methods, with access to optimal information, are shown in black. Items in bold and green represent the best performances with limited information on ψ, making them more easily usable in many scenarios. When there is only one element in bold and green, this implies that the best overall method is a non-oracle method.

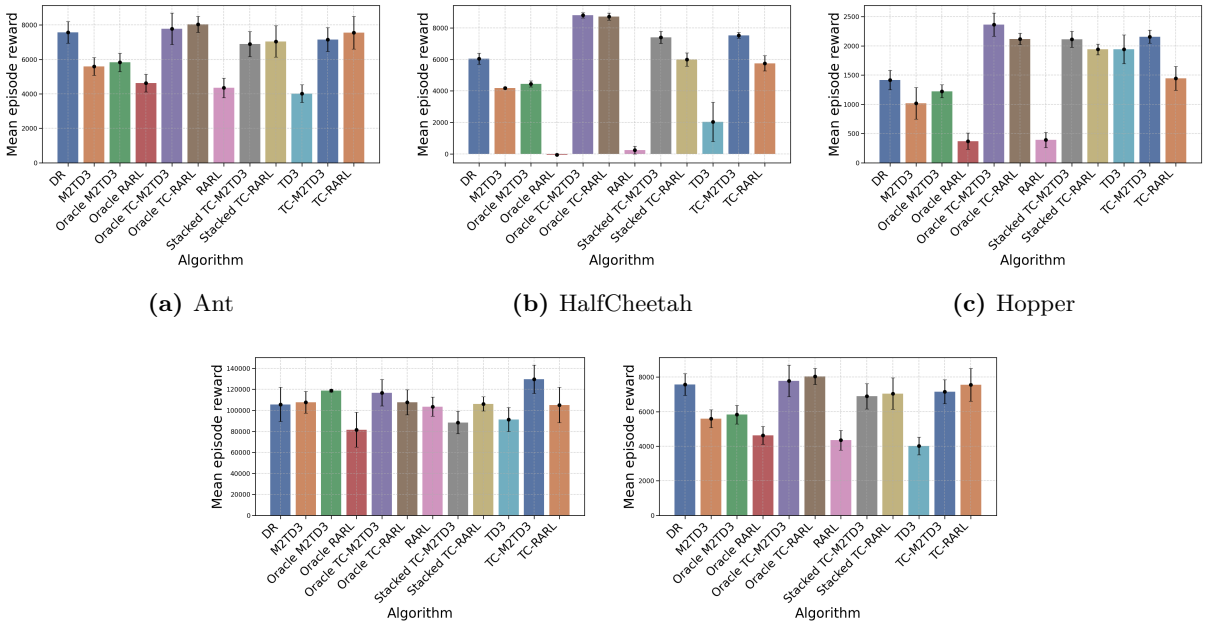
	Ant	HalfCheetah	Hopper	Humanoid	Walker	Agg.
Oracle M2TD3	1.11 ± 0.07	0.95 ± 0.1	1.51 ± 0.84	2.07 ± 0.19	1.31 ± 0.36	1.39 ± 0.31
Oracle RARL	0.72 ± 0.18	-0.71 ± 0.05	-1.3 ± 0.28	-2.8 ± 1.62	-0.19 ± 0.2	-0.86 ± 0.47
`Oracle-TC` -M2TD3	1.61 ± 0.32	$\mathbf{2.76 \pm 0.16}$	$\mathbf{7.79 \pm 1.0}$	1.69 ± 2.14	1.49 ± 0.41	$\mathbf{3.07 \pm 0.81}$
`Oracle-TC` -RARL	$\mathbf{1.66 \pm 0.32}$	2.63 ± 0.12	6.86 ± 1.46	0.19 ± 1.68	1.34 ± 0.11	2.54 ± 0.74
`Stacked-TC` -M2TD3	1.33 ± 0.21	$\mathbf{2.4 \pm 0.19}$	$\mathbf{6.51 \pm 0.59}$	-1.42 ± 1.44	$\mathbf{1.69 \pm 0.33}$	2.1 ± 0.55
`Stacked-TC` -RARL	1.48 ± 0.22	1.76 ± 0.08	3.28 ± 0.27	1.39 ± 0.57	1.01 ± 0.21	1.78 ± 0.27
`TC` -M2TD3	1.52 ± 0.2	$\mathbf{2.42 \pm 0.1}$	5.16 ± 0.2	$\mathbf{4.02 \pm 1.23}$	1.38 ± 0.25	$\mathbf{2.9 \pm 0.4}$
`TC` -RARL	1.57 ± 0.26	1.54 ± 0.15	2.04 ± 0.49	1.25 ± 1.91	0.89 ± 0.2	1.46 ± 0.6
TD3	0.0 ± 0.19	0.0 ± 0.27	0.0 ± 1.27	0.0 ± 1.18	0.0 ± 0.23	0.0 ± 0.63
DR	$\mathbf{1.58 \pm 0.2}$	1.59 ± 0.12	2.28 ± 0.42	0.87 ± 1.79	1.03 ± 0.19	1.47 ± 0.54
M2TD3	1.0 ± 0.19	1.0 ± 0.14	1.0 ± 0.96	1.0 ± 1.31	1.0 ± 0.31	1.0 ± 0.58
RARL	0.63 ± 0.2	-0.61 ± 0.18	-1.5 ± 0.33	0.8 ± 0.88	0.27 ± 0.25	-0.08 ± 0.37

Table 6.1: Avg. of normalized time-coupled worst-case performance over 10 seeds for each method

Performance of TCRMDPs in worst-case time-constrained. Table 6.1 reports the worst-case time-constrained perturbation. To address the worst-case time-constrained perturbations for each trained agent π^*, we utilized a time-constrained adversary using TD3 algorithm $\bar{\pi}^* = \min_{b\in\mathcal{B}} \mathbb{E}_{a\sim\pi^*(s),b\sim\bar{\pi}(s,a,\psi)}[r(s,a) + \gamma\mathbb{E}_{s'\sim P_{\psi+b}(s,a)}v_n(s')]$ within a perturbation radius of $L = 0.001$ for a total of 5 million steps. The sum of episode rewards was averaged over 10 episodes. To compare metrics across different environments, each method's score v was standardized relative to the reference score of TD3. TD3 was trained on the environment using default transition function parameters, with its score denoted as v_{TD3}. The M2TD3 score, v_{M2TD3}, was used as the comparison target. The formula applied was $(v - v_{TD3})/(|v_{M2TD3} - v_{TD3}|)$. This positioned v_{TD3} as the minimal baseline and v_{M2TD3} as the target score. This standardisation provides a metric that quantifies the improvement of each method over TD3 in relation to the improvement of M2TD3 over TD3. In each evaluation environment, agents trained with the time-constrained framework (indicated by TC in the method name) demonstrated significantly superior performance compared to those trained using alternative robust reinforcement learning approaches, including M2TD3 and RARL. Furthermore, they outperformed those trained through domain randomisation (DR). Notably, even without directly conditioning the policy with ψ, the time-constrained trained policies excelled against all baselines, achieving up to a 2.9-fold improvement. The non-normalized scores are reported in Appendix 31. Additionally, when policies were directly conditioned by ψ and trained within the robust reinforcement learning framework, they tended to be overly conservative in the time-constrained framework. This is depicted in Table 6.1, comparing the performances of Oracle RARL, Oracle M2TD3, Oracle TC-RARL, and Oracle TC-M2TD3. Both policies also observe ψ. The only difference is that Oracle RARL and Oracle M2TD3 were trained in the robust reinforcement learning framework, while Oracle TC-RARL and Oracle TC-M2TD3 were trained in the time-constrained framework. The performance differences under worst-case time-coupled perturbation are as follows: for Oracle RARL (resp. M2TD3) and Oracle TC-RARL (resp. M2TD3), the values are -0.86 (1.39) vs. 2.54 (3.07). This observation highlights the need for a balance between robust training and flexibility in dynamic conditions. A natural question arises regarding the worst-case time-constrained perturbation. Was the adversary in the loop adequately trained, or might its suboptimal performance lead to overestimating the trained agent's reward against the worst-case perturbation? The adversary's performance was monitored during its training against all fixed-trained agents. The results in Appendix 29 show that our adversary converged.

Robust Time-Constrained Training under various time fixed adversaries. The method was evaluated against various fixed adversaries, focusing on the random fixed adversary shown in Figure 6.2. This evaluation shows that robustly trained agents can handle dynamic and unpredictable conditions. The random fixed adversary simulates stochastic changes by selecting a parameter ψ_t at each timestep within a radius of $L = 0.1$. This radius is 100 times larger than in our training methods. At the start of each episode, ψ_0 is uniformly sampled from the uncertainty set $\psi_0 \sim \mathcal{U}(\mathcal{P})$. This tests the agents' adaptability to unexpected changes. Figures 6.2a through 6.2e show our agents' performance. Agents trained with our robust framework consistently outperformed those trained with standard methods. The policy was also assessed against five other fixed adversaries: cosine, exponential, linear, and logarithmic. Detailed results are provided in the Appendix. 31.1.

Performance of Robust Time-Constrained MDPs in the static setting. In static environments, the Robust Time-Constrained algorithms were evaluated for worst-case and average performance metrics, shown in Tables 7.6 and 7.7. A fixed uncertainty set $\mathcal{P}$ was used, dividing each dimension of Ψ into ten segments, creating a grid of 1000 points (10^3). Each agent ran five episodes at each grid point, and the rewards were averaged. The scores were normalized as described for the time-constrained adversary analysis in Table 6.1. The raw data is provided in Appendix 9.27 and 9.28. Performance scores were adjusted relative to the baseline v_{TD3} and



(a) Ant **(b)** HalfCheetah **(c)** Hopper

(d) Humanoid **(e)** Walker

Figure 6.2: Evaluation against a random fixed adversary, with a radius $L = 0.1$

v_{M2TD3}. As a result, normalized results reveal distinct trends among agent configurations within the TC-RMDP framework. The Oracle TC-M2TD3 variant achieved an average score of 3.12 7.7, while the Stacked TC-M2TD3 scored 2.23, indicating its resilience. Furthermore, in the worst-case scenario, the TC-RARL and Stacked TC-RARL variants demonstrated adaptability, with TC-RARL scoring 0.92 and TC-M2TD3 scoring 1.02 7.6. This performance highlights its reliability in challenging static environments.

6.6 Some Theoretical properties of TC-MDPS

6.6.1 On the optimal policy of TC

Following Lemma 3.3 of (Iyengar 2005), it is known that in the rectangular case, there exists an optimal policy of the adversary that is stationary, provided that the actor policy is stationary. The TC-RMDP definition enforces a limitation on the temporal variation of the transition kernel. Consequently, all stationary adversarial policies are constrained by this stipulation. In turn, this guarantees that (under the hypothesis of *sa*-rectangularity) there always exists a solution to the TC-RMDP that is also a solution to the original RMDP. In other words: optimizing policies for TC-RMDPs do not exclude optimal solutions to the underlying RMDP. This sheds an interesting light on the search for robust optimal policies, since TC-RMDPs shrink the search space of optimal adversarial policies. In practice, this is confirmed by the previous experimental results (Figure 7.6) where the optimal agent policy found by either `Oracle-TC` , `Stacked-TC` , or vanilla `TC` actually outperforms the one found by M2TD3 or RARL in the non time-constrained setting.

6.6.2 Some Lipchitz-properties for non-stationary TC-RMPDS

In this subsection we slightly depart from the framework defined in Section 6.2 and study the smoothness of the robust objective for vanilla `TC` or `Stacked-TC` . Th. 6.2.1 is no longer

	Ant	HalfCheetah	Hopper	Humanoid	Walker	Agg
Oracle M2TD3	$\mathbf{1.02 \pm 0.19}$	0.34 ± 0.23	0.97 ± 0.55	$\mathbf{3.9 \pm 3.65}$	0.3 ± 0.45	$\mathbf{1.31 \pm 1.01}$
Oracle RARL	0.62 ± 0.32	0.1 ± 0.02	0.48 ± 0.19	-2.59 ± 2.18	0.16 ± 0.21	-0.25 ± 0.58
Oracle-TC -M2TD3	0.1 ± 0.25	$\mathbf{1.87 \pm 0.1}$	0.49 ± 1.07	-0.8 ± 3.05	0.28 ± 0.38	0.39 ± 0.97
Oracle-TC -RARL	0.59 ± 0.36	1.55 ± 0.35	0.4 ± 0.16	1.19 ± 1.24	0.56 ± 0.39	0.86 ± 0.5
Stacked-TC -M2TD3	-0.05 ± 0.09	$\mathbf{1.56 \pm 0.16}$	1.08 ± 0.89	-0.83 ± 2.62	1.12 ± 0.5	0.58 ± 0.85
Stacked-TC -RARL	0.07 ± 0.13	0.76 ± 0.34	$\mathbf{1.35 \pm 0.93}$	$\mathbf{1.75 \pm 2.48}$	0.67 ± 0.32	0.92 ± 0.84
TC -M2TD3	-0.06 ± 0.08	1.49 ± 0.23	1.29 ± 0.29	1.21 ± 2.44	$\mathbf{1.19 \pm 0.34}$	$\mathbf{1.02 \pm 0.68}$
TC -RARL	0.14 ± 0.24	0.89 ± 0.3	1.5 ± 0.76	1.4 ± 4.57	0.67 ± 0.59	0.92 ± 1.29
TD3	0.0 ± 0.34	0.0 ± 0.06	0.0 ± 0.21	0.0 ± 2.27	0.0 ± 0.1	0.0 ± 0.6
DR	0.06 ± 0.16	1.07 ± 0.36	0.86 ± 0.82	0.04 ± 4.1	0.57 ± 0.37	0.52 ± 1.16
M2TD3	$\mathbf{1.0 \pm 0.27}$	1.0 ± 0.16	1.0 ± 0.65	1.0 ± 3.32	1.0 ± 0.63	1.0 ± 1.01
RARL	0.44 ± 0.3	0.13 ± 0.08	0.5 ± 0.22	0.44 ± 2.94	0.12 ± 0.09	0.33 ± 0.73

Table 6.2: Avg. of normalized static worst-case performance over 10 seeds for each method

	Ant	HalfCheetah	Hopper	Humanoid	Walker	Agg
Oracle M2TD3	1.13 ± 0.08	1.56 ± 0.24	1.12 ± 0.46	1.96 ± 1.53	1.23 ± 0.3	1.4 ± 0.52
Oracle RARL	0.7 ± 0.22	-1.4 ± 0.13	-0.77 ± 0.24	-2.6 ± 2.88	-1.13 ± 0.84	-1.04 ± 0.86
Oracle-TC -M2TD3	1.73 ± 0.09	$\mathbf{4.35 \pm 0.26}$	$\mathbf{5.54 \pm 0.13}$	2.12 ± 1.4	1.84 ± 0.37	$\mathbf{3.12 \pm 0.45}$
Oracle-TC -RARL	$\mathbf{1.78 \pm 0.02}$	4.32 ± 0.21	5.08 ± 0.48	0.42 ± 2.9	1.68 ± 0.24	2.66 ± 0.77
Stacked-TC -M2TD3	1.45 ± 0.38	$\mathbf{3.78 \pm 0.29}$	$\mathbf{5.2 \pm 0.29}$	-1.38 ± 1.67	$\mathbf{2.11 \pm 0.52}$	2.23 ± 0.63
Stacked-TC -RARL	1.52 ± 0.11	2.29 ± 0.23	2.91 ± 0.67	1.14 ± 2.19	1.21 ± 0.46	1.81 ± 0.73
TC -M2TD3	1.6 ± 0.06	3.71 ± 0.24	4.4 ± 0.6	$\mathbf{3.28 \pm 2.52}$	1.56 ± 0.23	$\mathbf{2.91 \pm 0.73}$
TC -RARL	$\mathbf{1.67 \pm 0.07}$	2.27 ± 0.22	1.79 ± 0.53	0.89 ± 2.19	1.01 ± 0.21	1.53 ± 0.64
TD3	0.0 ± 0.49	0.0 ± 0.22	0.0 ± 0.83	0.0 ± 1.36	0.0 ± 0.51	0.0 ± 0.68
DR	1.65 ± 0.05	2.31 ± 0.27	2.08 ± 0.49	1.15 ± 2.47	1.22 ± 0.34	1.68 ± 0.72
M2TD3	1.0 ± 0.11	1.0 ± 0.19	1.0 ± 0.55	1.0 ± 1.43	1.0 ± 0.65	1.0 ± 0.59
RARL	0.69 ± 0.13	-1.3 ± 0.54	-0.99 ± 0.11	0.47 ± 1.92	-0.35 ± 0.83	-0.3 ± 0.71

Table 6.3: Avg. of normalized static average case performance over 10 seeds for each method

applicable as ψ is not observed. However, we can still give smoothness of the objective starting from Lipchichz conditions on the evolution of the parameter that leads to smoothness on reward and transition kernel in the following definition 6.6.1.

Definition 6.6.1 (Reward/Kernel Lipchitz TC-RMDPs (Lecarpentier and Rachelson 2019))**.** *We say that a parametric RDMPs is time constrained if the parameter change is bounded through time ie.* $\|\psi_{t+1} - \psi_t\| \leq L$. *Moreover, we assume that this variation in parameter implies a variation in the reward and transition kernel of*

$$\forall s \in \mathcal{S}, \forall a \in \mathcal{A}, \|P_t(\cdot \mid s,a) - P_{t+1}(\cdot \mid s,a)\|_1 \leq L_P \quad ; |r_t(s,a) - r_{t+1}(s,a)| \leq L_r \ .$$

From a theoretical point of view, a TC-RMDP can be seen as a sequence of stationary MDPs with time indexed reward and transition kernel r_t, P_t that have continuity. More formally for $M_t = (\mathcal{S}, \mathcal{A}, \Psi, P_{\psi_t}, r_t, L = (L_p, L_r))$, we can then define the sequence of stationary MDPs with Lipchitz variation :

$$\mathcal{M}_t^L = \Big\{ \{M_t\}_{t'=t_0}^t ; \exists L_r \in \mathbb{R} \forall s \in \mathcal{S}, \forall a \in \mathcal{A}, \left\| P_{\psi_{t'}}(\cdot \mid s,a) - P_{\psi_{t'+1}}(\cdot \mid s,a) \right\|_1 \leq L_P \quad ; \\ |r_{t'}(s,a) - r_{t'+1}(s,a)| \leq L_r \Big\} \ . \tag{6.1}$$

Defining r_t^k as the random variable corresponding to the reward function at time step t for stationary MDPs, but iterating with index k, the stationary rollout return at time t is $G(\pi, M_t) = \sum_{k \geq 0} \gamma^k r_t^k$. Assuming that at a fixed t the reward and transition kernel r_t, P_t are fixed, the robust objective function is:

$$J^R(\pi, t) := \min_{m = \{m'_t\}_{t'=t_0}^t \in \mathcal{M}_t^L} \mathbb{E}\left[G\left(\pi, m\right)\right] \ .$$

This leads to the following guarantee for vanilla `TC` and `Stacked-TC` algorithms.

Theorem 6.6.1. *Assume TC-RMPDS with* $L = (L_r, L_P)$ *smoothness. Then* $\forall t \in \mathbb{N}, r_t \in [0,1]$,

$$\forall t \in \mathbb{N}^+, \forall t_0 \in \mathbb{N}^+, \quad |J^R(\pi, t_0) - J^R(\pi, t_0 + t)| \leq L' t \ , \tag{6.2}$$

with $L' := \left(\frac{\gamma}{(1-\gamma)^2} L_P + \frac{1}{1-\gamma} L_r \right)$.

This theorem states that a small variation of the Kernel and reward function will not affect too much the robust objective. In other terms, despite the fact that the TC Bellman operator may not admit a fixed point and yield a non-stationary sequence of value functions, variations of the expected return remain bounded. Proof of the Th. 6.6.1 can be found in Appendix 26.

6.7 Conclusion

This paper presents a novel framework for robust reinforcement learning, which addresses the limitations of traditional methods that rely on rectangularity assumptions. These assumptions often result in overly conservative policies, which are not suitable for real-world applications where environmental disturbances are multifactorial, correlated, and time-constrained. In order to overcome these challenges, we proposed a new formulation, the Time-Constrained Robust Markov Decision Process (TC-RMDP). The TC-RMDP framework is capable of accurately capturing the dynamics of real-world environments, due to its consideration of the temporal continuity and correlation of disturbances. This approach resulted in the development of three

algorithms: The three algorithms, `Oracle-TC` , `Stacked-TC` , vanilla `TC` which differ in the extent to which environmental information is incorporated into the decision-making process. A comprehensive evaluation of continuous control benchmarks using MuJoCo environments has demonstrated that the proposed TC-RMDP algorithms outperform traditional robust RL methods and domain randomization techniques. These algorithms achieved a superior balance between performance and robustness in both time-constrained and static settings. The results confirmed the effectiveness of the TC-RMDP framework in reducing the conservatism of policies while maintaining robustness. Moreover, we provided theoretical guaranties for `Oracle-TC` in Th. 6.2.1 and for `Stacked-TC` and vanilla `TC` in Th. 6.6.1. This study contributes to the field of robust reinforcement learning by introducing a time-constrained framework that more accurately reflects the dynamics observed in real-world settings. The proposed algorithms and theoretical contributions offer new avenues for the development of more effective and practical RL applications in environments with complex, time-constrained uncertainties. In the next Chapter, we will provide a new Robust RL benchmark based on Mujoco to evaluate Robustness of RL algorithm and improve reproducibility of Robust RL algorithm.

Chapter 7

RRLS: Robust Reinforcement Learning Suite

Contents

7.1 Introduction

Reinforcement learning (RL) algorithms frequently encounter difficulties in maintaining performance when confronted with dynamic uncertainties and varying environmental conditions. This lack of robustness significantly limits their applicability in the real world. Robust reinforcement learning addresses this issue by focusing on learning policies that ensure optimal worst-case performance across a range of adversarial conditions. For instance, an aircraft control policy should be capable of effectively managing various configurations and atmospheric conditions without requiring retraining. This is critical for applications where safety and reliability are paramount to avoid a drastic decrease in performance Morimoto and Doya (2005), Tessler et al. (2019).

The concept of robustness, as opposed to resilience, places greater emphasis on maintaining performance without further training. In robust reinforcement learning (RL), the objective is to optimize policies for the worst-case scenarios, ensuring that the learned policies can handle the most challenging conditions. This framework is formalized through robust Markov decision processes (MDPs), where the transition dynamics are subject to uncertainties. Despite significant advancements in robust RL algorithms, the field lacks standardized benchmarks for evaluating these methods. This hampers reproducibility and comparability of experimental results (Moos et al. 2022). To address this gap, we introduce the Robust Reinforcement Learning Suite, a comprehensive benchmark suite designed to facilitate rigorous evaluation of robust RL algorithms.

The Robust Reinforcement Learning Suite (RRLS) provides six continuous control tasks based on Mujoco Todorov et al. (2012) environments, each with distinct uncertainty sets for training and evaluation. By standardizing these tasks, RRLS enables reproducible and comparable experiments, promoting progress in robust RL research. The suite includes four compatible baselines with the RRLS benchmark, which are evaluated in static environments to demonstrate their efficacy. In summary, our contributions are the following :

- Our first contribution aims to establish a standardized benchmark for robust RL, addressing the critical need for reproducibility and comparability in the field (Moos et al. 2022). The RRLS benchmark suite represents a significant step towards achieving this goal, providing a robust framework for evaluating state-of-the-art robust RL algorithms.
- Our second contribution is a comparison and evaluation of different Deep Robust RL algorithms in Section 7.5 on our benchmark, showing the pros and cons of different methods.

7.2 Problem statement

Reinforcement learning. Reinforcement Learning (RL) (Sutton and Barto 2018) addresses the challenge of developing a decision-making policy for an agent interacting with a dynamic environment over multiple time steps. This problem is modeled as a Markov Decision Process (MDP) (Puterman 2014) represented by the tuple $(\mathcal{S}, \mathcal{A}, P, r)$, which includes states S, actions A, a transition kernel $P(s_{t+1}|s_t, a_t)$, and a reward function $r(s_t, a_t)$. For simplicity, we assume a unique initial state s_0, though the results generalize to an initial state distribution $p_0(s)$. A stationary policy $\pi(s) \in \Delta(A)$ maps states to distributions over actions. The objective is to find a policy π that maximizes the expected discounted return

$$J^\pi = \mathbb{E}_{s_0\sim\rho}[V^{\pi,P}(s_0)] = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)|a_t \sim \pi, s_{t+1} \sim P, s_0 \sim \rho\Big], \tag{7.1}$$

where $V^{\pi,P}$ is the value function of π, $\gamma \in [0, 1)$ is the discount factor, and s_0 is drawn from the initial distribution ρ. The value function $V^{\pi,P}$ of policy π assigns to each state s the expected discounted sum of rewards when following π starting from s and following transition kernel p. An optimal policy π^* maximizes the value function in all states. To converge to the (optimal) value function, the value iteration (VI) algorithm can be applied, which consists in repeated application of the (optimal) Bellman operator $\mathcal{T}^{*,P}$ to value functions:

$$V_{n+1}(s) = \mathcal{T}^* V_n(s) := \max_{\pi(s)\in\Delta(A)} \mathbb{E}_{a\sim\pi(s)}[r(s, a) + \mathbb{E}_P[V_n(s')]]. \tag{7.2}$$

Finally, the Q function is also defined similarly to Equation (7.1) but starting from specific state/action (s, a) as $\forall(s, a) \in S \times A$:

$$Q^{\pi,P}(s, a) = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)|a_t \sim \pi, s_{t+1} \sim P, s_0 = s, a_0 = a\Big]. \tag{7.3}$$

Robust reinforcement learning. In a Robust MDP (RMDP) Iyengar (2005), Nilim and El Ghaoui (2005), the transition kernel p is not fixed and can be chosen adversarially from an uncertainty set $\mathcal{P}$ at each time step. The pessimistic value function of a policy π is defined as $V^\pi_{\mathcal{P}}(s) = \min_{p\in\mathcal{P}} v^\pi_p(s)$. An optimal robust policy maximizes the pessimistic value function $V_{\mathcal{P}}$ in any state, leading to a $\max_\pi \min_p$ optimization problem. This is known as the static model of transition kernel uncertainty, as π is evaluated against a static transition model π. Robust Value Iteration (RVI) (Iyengar 2005, Wiesemann et al. 2013) addresses this problem by iteratively computing the one-step lookahead best pessimistic value:

$$V_{n+1}(s) = \mathcal{T}^*_{\mathcal{P}} V_n(s) := \max_{\pi(s)\in\Delta(A)} \min_{P\in\mathcal{P}} \mathbb{E}_{a\sim\pi(s)}[r(s, a) + \mathbb{E}_P[V_n(s')]]. \tag{7.4}$$

This dynamic programming formulation is called the dynamic model of transition kernel uncertainty, as the adversary picks the next state distribution only for the current state-action

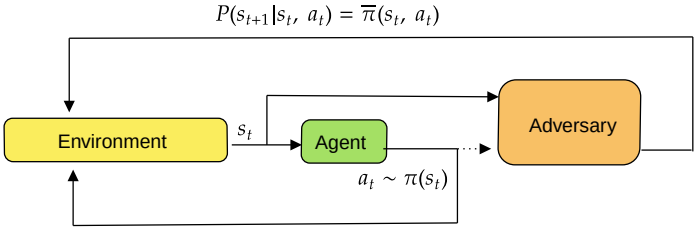


Figure 7.1: Relation between Robust RL and Zero-sum Markov Game

pair, after observing the current state and the agent's action at each time step (and not a full transition kernel). The $\mathcal{T}^*_{\mathcal{P}}$ operator, known as the robust Bellman operator, ensures that the sequence of V_n functions converges to the robust value function $V^*_{\mathcal{P}}$, provided the adversarial transition kernel belongs to the simplex of $\Delta(S)$ and that the static and dynamic cases have the same solutions for stationary agent policies Iyengar (2022).

Robust reinforcement learning as a two-player game. Robust MDPs can be represented as zero-sum two-player Markov games (Littman 1994, Tessler et al. 2019) where $\bar{S}, \bar{A}$ are respectively the state and action set of the adversarial player. In a zero-sum Markov game, the adversary tries to minimize the reward or maximize $-r$. Writing $\bar{\pi} : \bar{S} \to \bar{A} := \Delta(S)$ the policy of this adversary, the robust MDP problem turns to $\max_\pi \min_{\bar{\pi}} V^{\pi,\bar{\pi}}$, where $V^{\pi,\bar{\pi}}(s)$ is the expected sum of discounted rewards obtained when playing π (agent actions) against $\bar{\pi}$ (transition models) at each time step from s. In the specific case of robust RL as a two player-game, $\bar{S} = S \times A$. This enables introducing the robust value iteration sequence of functions

$$V_{n+1}(s) := \mathcal{T}^{**}V_n(s) := \max_{\pi(s)\in\Delta(A)} \min_{\bar{\pi}(s,a)\in\Delta(S)} \mathcal{T}^{\pi,\bar{\pi}}V_n(s) \tag{7.5}$$

where $\mathcal{T}^{\pi,\bar{\pi}} := \mathbb{E}_{a\sim\pi(s)}[r(s,a) + \gamma\mathbb{E}_{s'\sim\bar{\pi}(s,a)}V_n(s')]$ is a zero-sum Markov game operator. These operators are also $\gamma-$contractions and converge to their respective fixed point $V^{\pi,\bar{\pi}}$ and $V^{**} = V^*_{\mathcal{P}}$ Tessler et al. (2019). This two-player game formulation will be used in the evaluation of the RRLS in Section 7.5.

7.3 Related works

7.3.1 Reinforcement learning benchmark

The landscape of reinforcement learning (RL) benchmarks has evolved significantly, enabling the accelerated development of RL algorithms. Prominent among these benchmarks are the Atari Arcade Learning Environment (ALE) Bellemare et al. (2012), OpenAI Gym Brockman et al. (2016), more recently Gymnasium Towers et al. (2023), and the DeepMind Control Suite (DMC) Tassa et al. (2018). The aforementioned benchmarks have established standardized environments for the evaluation of RL agents across discrete and continuous action spaces, thereby fostering the reproducibility and comparability of experimental results. The ALE has been particularly influential, offering a diverse set of Atari games that have become a standard testbed for discrete control tasks Bellemare et al. (2012). Moreover, the OpenAI Gym extended this approach by providing a more flexible and extensive suite of environments for various RL tasks, including discrete and continuous control Brockman et al. (2016). Similarly, the DMC Suite has been essential for benchmarking continuous control algorithms, offering a set of challenging tasks that facilitate evaluating algorithm performance Tassa et al. (2018). In addition to these

general-purpose benchmarks, specialized benchmarks have been developed to address specific research needs. For instance, the DeepMind Lab focuses on 3D navigation tasks from pixel inputs Beattie et al. (2016), while ProcGen Cobbe et al. (2019) offers procedurally generated environments to evaluate the generalization capabilities of RL agents. The D4RL benchmark targets offline RL methods by providing datasets and tasks specifically designed for offline learning scenarios Fu et al. (2021), and RL Unplugged Gulcehre et al. (2020) offers a comprehensive suite of benchmarks for evaluating offline RL algorithms. RL benchmarks such as Meta-World Yu et al. (2021) have been developed to evaluate the ability of RL agents to transfer knowledge across multiple tasks. Meta-World provides a suite of robotic manipulation tasks designed to test RL algorithms' adaptability and generalization in multitask learning scenarios. Similarly, RLBench James et al. (2020) offers a variety of tasks for robotic learning, focusing on the performance of RL agents in multi-task settings. Recent contributions such as the Unsupervised Reinforcement Learning Benchmark (URLB) Lee et al. (2021) have further expanded the scope of RL benchmarks by targeting unsupervised learning methods. URLB aims to accelerate progress in unsupervised RL by providing a suite of environments and baseline implementations, promoting algorithm development that does not rely on labeled data for training. Additionally, the CoinRun benchmark Cobbe et al. (2020) and Sonic Benchmark Nichol et al. (2018) focus on evaluating generalization and transfer learning in RL through procedurally generated levels and video game environments, respectively. Finally, benchmarks like the Behavior Suite (bsuite) Osband et al. (2019) have been designed to test specific capabilities of RL agents, such as memory, exploration, and generalization. Closer to our work, safety in RL is another critical area where benchmarks like SafetyGym Achiam and Amodei (2019) have been instrumental. SafetyGym evaluates how well RL agents can perform tasks while adhering to safety constraints, which is crucial for real-world applications where safety cannot be compromised. Despite the progress in benchmarking RL algorithms, there has been a notable gap in benchmarks specifically designed for robust RL, which aims to learn policies that perform optimally in the worst-case scenario against adversarial environments. This gap highlights the need for standardized benchmarks (Moos et al. 2022) that facilitate reproducible and comparable experiments in robust RL. In the next section, we introduce existing robust RL algorithms.

Finally, a competing work Gu et al. (2024) published after ours, and which cites our research, has many similarities as it is also a benchmark for robust RL. The differences between our work are as follows. Their work includes a larger number of environments, which in a sense makes it more comprehensive than ours. Our benchmark has been tested on robust RL algorithms such as RARL, M2TD3, demonstrating its utility, whereas the competing work has not yet been evaluated in this way in all tasks. Our benchmark differs in that it goes beyond simply adding noise to the transition kernel; it provides a rigorous evaluation framework by varying hyperparameters on a relevant grid or uncertainty set during the evaluation phase.

7.3.2 Robust Reinforcement Learning algorithms

Two principal classes of practical, robust reinforcement learning algorithms exist, those that can interact solely with a nominal transition kernel (or center of the uncertainty set), and those that can sample from the entire uncertainty ball. While the former is more mathematically founded, it is unable to exploit transitions that are not sampled from the nominal kernel and consequently exhibits lower performance. In this benchmark, only the Deep Robust RL as two-player games that use samples from the entire uncertainty set are implemented.

Nominal-based Robust/risk-averse algorithms. The idea of this class of algorithms is to approximate the inner minimum operator present robust Bellman operator in Equation (7.4). Previous work has typically employed a dual approach to the minimum problem, whereby

the transition probability is constrained to remain within a specified ball around the nominal transition kernel. Practically, robustness is equivalent to regularization (Derman et al. 2021) and for example the SAC algorithm Haarnoja et al. (2018a) has been shown to be robust due to entropic regularization. In this line of work, (Kumar et al. 2022) derived approximate algorithm for RMPDS with L_p balls, (Clavier et al. 2022) for χ^2 constrain and (Liu et al. 2022) for KL divergence. Finally, Wang et al. (2023) proposes a novel online approach to solve RMDP. Unlike previous works that regularize the policy or value updates, Wang et al. (2023) achieves robustness by simulating the worst kernel scenarios for the agent while using any classical RL algorithm in the learning process. These Robust RL approaches have received recent theoretical attention, from a statistical point of view (sample complexity) (Yang et al. 2022, Panaganti and Kalathil 2022a, Clavier et al. 2023, Shi et al. 2024) as well as from an optimization point of view (Grand-Clément and Kroer 2021), but generally do not directly translate to algorithms that scale up to complex evaluation benchmarks.

Deep Robust RL as two-player games. A common approach to solving robust RL problems is cast the optimization process as a two-player game, as formalized by Morimoto and Doya (2005), described in Section 7.2, and summarized in Figure 7.1. In this framework, an adversary, denoted by $\bar{\pi} : \mathcal{S} \times \mathcal{A} \to \mathcal{P}$, is introduced, and the game is formulated as

$$\max_{\pi} \min_{\bar{\pi}} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1}) | s_0, a_t \sim \pi(s_t), P_t = \bar{\pi}(s_t, a_t), s_{t+1} \sim P_t(\cdot|s_t, a_t)\right].$$

Most methods differ in how they constrain $\bar{\pi}$'s action space within the uncertainty set. A first family of methods define $\bar{\pi}(s_t) = P_{ref} + \Delta(s_t)$, where p_{ref} denotes the reference (nominal) transition function. Among this family, Robust Adversarial Reinforcement Learning (RARL) (Pinto et al. 2017) applies external forces at each time step t to disturb the reference dynamics. For instance, the agent controls a planar monopod robot, while the adversary applies a 2D force on the foot. In noisy action robust MDPs (NR-MDP) (Tessler et al. 2019) the adversary shares the same action space as the agent and disturbs the agent's action $\pi(s)$. Such gradient-based approaches incur the risk of finding stationary points for π and $\bar{\pi}$ which do not correspond to saddle points of the robust MDP problem. To prevent this, Mixed-NE (Kamalaruban et al. 2020) defines mixed strategies and uses stochastic gradient Langevin dynamics. Similarly, Robustness via Adversary Populations (RAP) (Vinitsky et al. 2020) introduces a population of adversaries, compelling the agent to exhibit robustness against a diverse range of potential perturbations rather than a single one, which also helps prevent finding stationary points that are not saddle points.

Aside from this first family, State Adversarial MDPs (Zhang et al. 2020; 2021, Stanton et al. 2021) involve adversarial attacks on state observations, which implicitly define a partially observable MDP. This case aims not to address robustness to the worst-case transition function but rather against noisy, adversarial observations.

A third family of methods considers the general case of $\bar{\pi}(s_t, a_t) = P_t$ or $\bar{\pi}(s_t) = p_t$, where $P_t \in \mathcal{P}$. Minimax Multi-Agent Deep Deterministic Policy Gradient (M3DDPG) (Li et al. 2019b) is designed to enhance robustness in multi-agent reinforcement learning settings but boils down to standard robust RL in the two-agents case. Max-min TD3 (M2TD3) (Tanabe et al. 2022a) considers a policy π, defines a value function $Q(s, a, p)$ which approximates $Q^{\pi,P}(s, a) = \mathbb{E}_{s' \sim P}[r(s, a, s') + \gamma V^{\pi,P}(s')]$, updates an adversary $\bar{\pi}$ so as to minimize $Q(s, \pi(s), \bar{\pi}(s))$ by taking a gradient step with respect to $\bar{\pi}$'s parameters, and updates the policy π using a TD3 gradient update in the direction maximizing $Q(s, \pi(s), \bar{\pi}(s))$. As such, M2TD3 remains a robust value iteration method that solves the dynamic problem by alternating updates on π and $\bar{\pi}$, but since it approximates $Q^{\pi,P}$, it is also closely related to the method we introduce in the next section.

Domain randomization. Domain randomization (DR) (Tobin et al. 2017) learns a value function $V(s) = \max_{\pi} \mathbb{E}_{p \sim \mathcal{U}(\mathcal{P})} V_p^{\pi}(s)$ which maximizes the expected return *on average* across a

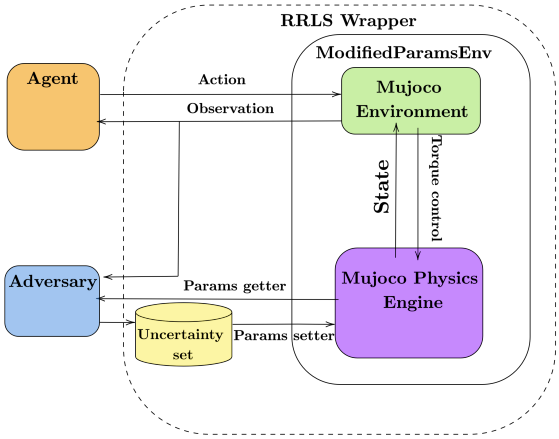


Figure 7.2: RRLS architecture

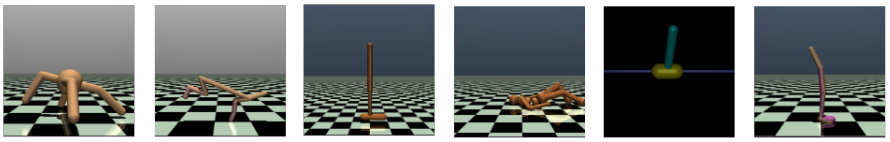


Figure 7.3: Visual representation of various reinforcement learning environments including **Ant**, **HalfCheetah**, **Hopper**, **Humanoid Stand Up**, **Inverted Pendulum**, and **Walker**.

fixed distribution on $\mathcal{P}$. As such, DR approaches do not optimize the worst-case performance. Nonetheless, DR has been used convincingly in applications (Mehta et al. 2020a, OpenAI et al. 2019). Similar approaches also aim to refine a base DR policy for application to a sequence of real-world cases (Lin et al. 2020, Dennis et al. 2020, Yu et al. 2018). For a more complete survey of recent works in robust RL, we refer the reader to the work of Moos et al. (2022).

7.4 RRLS: Benchmark environments for Robust RL

This section introduces the Robust Reinforcement Learning Suite, which extends the Gymnasium Towers et al. (2023) API with two additional methods: `set_params` and `get_params`. These methods are integral to the `ModifiedParamsEnv` interface, facilitating environment parameter modifications within the benchmark environment. Typically, these methods are used within a wrapper to simplify parameter modifications during evaluation. In the RRLS architecture (Figure 7.2), the adversary begins by retrieving parameters from the uncertainty set and setting them in the environment using the `ModifiedParamsEnv` interface. The agent then acts based on the current state of the environment, and the Mujoco Physics Engine updates the state accordingly. The agent observes this updated state, completing the interaction loop. Multiple MuJoCo environments are provided (Figure 7.3), each with a two default uncertainty sets, inspired respectively by those used in the experiments of RARL (Pinto et al. 2017) (Table 9.11) and M2TD3 (Tanabe et al. 2022a) (Table 7.2). This variety allows for a comprehensive evaluation of robust RL algorithms, ensuring that the benchmarks encompass a wide range of scenarios.

Several MuJoCo environments are proposed, each with distinct action and observation spaces. Figure 7.3 shows a visual representation of all provided environments. In all environments, the observation space corresponds to the positional values of various body parts followed by their

velocities, with all positions listed before all velocities. The environments are as follows:

- **Ant**: A 3D robot with one torso and four legs, each with two segments. The goal is to move forward by coordinating the legs and applying torques on the eight hinges. The action dimension is 8, and the observation dimension is 27.
- **HalfCheetah**: A 2D robot with nine body parts and eight joints, including two paws. The goal is to run forward quickly by applying torque to the joints. Positive rewards are given for forward movement, and negative rewards for moving backward. The action dimension is 6, and the observation dimension is 17.
- **Hopper**: A 2D one-legged figure with four main parts: torso, thigh, leg, and foot. The goal is to hop forward by applying torques on the three hinges. The action dimension is 3, and the observation dimension is 11.
- **Humanoid Stand Up**: A 3D bipedal robot resembling a human, with a torso, legs, and arms, each with two segments. The environment starts with the humanoid lying on the ground. The goal is to stand up and remain standing by applying torques to the various hinges. The action dimension is 17, and the observation dimension is 376.
- **Inverted Pendulum**: A cart that can move linearly, with a pole fixed at one end. The goal is to balance the pole by applying forces to the cart. The action dimension is 1, and the observation dimension is 4.
- **Walker**: A 2D two-legged figure with seven main parts: torso, thighs, legs, and feet. The goal is to walk forward by applying torques on the six hinges. The action dimension is 6, and the observation dimension is 17.

The RRLS architecture enables parameter modifications and adversarial interactions using the `gymnasium` Towers et al. (2023) interface. The `set_params` and `get_params` methods in the `ModifiedParamsEnv` interface directly access and modify parameters in the Mujoco Physics Engine. All modifiable parameters are listed in Appendix 34 and lie in the uncertainty set described below.

Uncertainty Sets. Non-rectangular uncertainty sets (opposed to rectangular ones as defined in (Iyengar 2005)) are proposed based on MuJoCo environments, detailed in Table 9.11. These sets, based on previous work evaluating M2TD3 Tanabe et al. (2022a) and RARL Pinto et al. (2017), ensure thorough testing of robust RL algorithms under diverse conditions. For instance, the uncertainty range for the torso mass in the HumanoidStandUp 2 and 3 environments spans from 0.1 to 16.0 (Table 9.11), ensuring challenging evaluation of RL methods. Three uncertainty sets—1D, 2D, and 3D—are provided for each environment, ranging from simple to challenging.

RRLS also directly provides the uncertainty sets from the RARL (Pinto et al. 2017) paper. These sets apply destabilizing forces at specific points in the system, encouraging the agent to learn robust control policies.

Wrappers. We introduce environment wrappers to facilitate the implementation of various deep robust RL baselines such as M2TD3 Tanabe et al. (2022a), RARL Pinto et al. (2017), Domain Randomization Tobin et al. (2017), NR-MDP Tessler et al. (2019) and all algorithms deriving from Robust Value Iteration, ensuring researchers can easily apply and compare different methods within a standardized framework. The wrappers are described as follows:

- The `ModifiedParamsEnv` interface includes methods `set_params` and `get_params`, which are crucial for modifying and retrieving environment parameters. This interface allows dynamic adjustment of the environment during training or evaluation.

Table 7.1: List of parameters uncertainty sets based on M2TD3 in RRLS

Environment	Uncertainty set $\mathcal{P}$	Reference values	Uncertainty parameters
Ant 1	$[0.1, 3.0]$	0.33	torsomass
Ant 2	$[0.1, 3.0] \times [0.01, 3.0]$	$(0.33, 0.04)$	torso mass; front left leg mass
Ant 3	$[0.1, 3.0] \times [0.01, 3.0] \times [0.01, 3.0]$	$(0.33, 0.04, 0.06)$	torso mass; front left leg mass; front right leg mass
HalfCheetah 1	$[0.1, 3.0]$	0.4	world friction
HalfCheetah 2	$[0.1, 4.0] \times [0.1, 7.0]$	$(0.4, 6.36)$	world friction; torso mass
HalfCheetah 3	$[0.1, 4.0] \times [0.1, 7.0] \times [0.1, 3.0]$	$(0.4, 6.36, 1.53)$	world friction; torso mass; back thigh mass
Hopper 1	$[0.1, 3.0]$	1.00	world friction
Hopper 2	$[0.1, 3.0] \times [0.1, 3.0]$	$(1.00, 3.53)$	world friction; torso mass
Hopper 3	$[0.1, 3.0] \times [0.1, 3.0] \times [0.1, 4.0]$	$(1.00, 3.53, 3.93)$	world friction; torso mass; thigh mass
HumanoidStandup 1	$[0.1, 16.0]$	8.32	torsomass
HumanoidStandup 2	$[0.1, 16.0] \times [0.1, 8.0]$	$(8.32, 1.77)$	torso mass; right foot mass
HumanoidStandup 3	$[0.1, 16.0] \times [0.1, 5.0] \times [0.1, 8.0]$	$(8.32, 1.77, 4.53)$	torso mass; right foot mass; left thigh mass
InvertedPendulum 1	$[1.0, 31.0]$	4.90	polemass
InvertedPendulum 2	$[1.0, 31.0] \times [1.0, 11.0]$	$(4.90, 9.42)$	pole mass; cart mass
Walker 1	$[0.1, 4.0]$	0.7	world friction
Walker 2	$[0.1, 4.0] \times [0.1, 5.0]$	$(0.7, 3.53)$	world friction; torso mass
Walker 3	$[0.1, 4.0] \times [0.1, 5.0] \times [0.1, 6.0]$	$(0.7, 3.53, 3.93)$	world friction; torso mass; thigh mass

- The `DomainRandomization` wrapper enables domain randomization by sampling environment parameters from the uncertainty set between episodes. It wraps an environment following the `ModifiedParamsEnv` interface and uses a randomization function to draw new parameter sets. If no function is set, the parameter is sampled uniformly. Parameters reset at the beginning of each episode, ensuring diverse training conditions.
- The `Adversarial` wrapper converts an environment into a robust reinforcement learning problem modeled as a zero-sum Markov game. It takes an uncertainty set and the `ModifiedParamsEnv` as input. This wrapper extends the action space to include adversarial actions, allowing for modifications of transition kernel parameters within a specified uncertainty set. It is suitable for reproducing robust reinforcement learning approaches based on adversarial perturbation in the transition kernel, such as RARL.
- The `ProbabilisticActionRobust` wrapper defines the adversary's action space as the same action space as the agent. The final action applied in the environment is a convex sum between the agent's action and the adversary's action: $a_{pr} = \alpha a + (1-\alpha)\bar{a}$. The adversarial action's effect is bounded by the environment's action space, allowing the implementation of robust reinforcement learning methods around a reference transition kernel, such as NR-MDP or RAP.

Evaluation Procedure. Evaluating Robust Reinforcement Learning algorithms can feature a large variability in outcome statistics depending on a number of minor factors (such as random

Table 7.2: List of parameters uncertainty sets based on RARL in RRLS

Environment	Uncertainty set $\mathcal{P}$	Uncertainty parameters
Ant Rarl	$[-3.0, 3.0]^{\times 6}$	torso force x; torso force y; front left leg force x; front left leg force y; front right leg force x; front right leg force y
HalfCheetah Rarl	$[-3.0, 3.0]^{\times 6}$	torso force x; torso force y; back foot force x; back foot force y; forward foot force x; forward foot force y
Hopper Rarl	$[-3.0, 3.0]^{\times 2}$	foot force x; foot force y
HumanoidStandup Rarl	$[-3.0, 3.0]^{\times 6}$	torso force x; torso force y; right thigh force x; right thigh force y; left foot force x; left foot force y
InvertedPendulum Rarl	$[-3.0, 3.0]^{\times 2}$	pole force x; pole force y
Walker Rarl	$[-3.0, 3.0]^{\times 4}$	leg force x; leg force y; left foot force x; left foot force y

seeds, initial state, or collection of evaluation transition models). To address this, we propose a systematic approach using a function called `generate_evaluation_set`. This function takes an uncertainty set as input and returns a list of evaluation environments. In the static case, where the transition kernel remains constant across time steps, the evaluation set consists of environments spanned by a uniform mesh over the parameters set. The agent runs multiple trajectories in each environment to ensure comprehensive testing. Each dimension of the uncertainty set is divided by a parameter named `nb_mesh_dim`. This parameter controls the granularity of the evaluation environments. To standardize the process, we provide a default evaluation set for each uncertainty set (Table 9.11). This set allows for worst-case performance and average-case performance evaluation in static conditions.

7.5 Benchmarking Robust RL algorithms

Experimental setup. This section evaluates several baselines in static and dynamic settings using RRLS. We conducted experimental validation by training policies in the Ant, HalfCheetah, Hopper, HumanoidStandup, and Walker environments. We selected five baseline algorithms: TD3, Domain Randomization (DR), NR-MDP, RARL, and M2TD3. We select the most challenging scenarios, the 3D uncertainty set defined in Table 9.11, normalized between $[0, 1]^3$. For static evaluation, we used the standard evaluation procedure proposed in the previous section. Performance metrics were gathered after five million steps to ensure a fair comparison after convergence. All baselines were constructed using TD3 with a consistent architecture across all variants. The results were obtained by averaging over ten distinct random seeds. Appendices 35, 37.1, 37.2, and 37.3 provide further details on hyperparameters, network architectures, implementation choices, and training curves.

Static worst-case performance. Tables 7.6 and 7.7 report normalized scores for each method, averaged across 10 random seeds and 5 episodes per seed, for each transition kernel in the evaluation uncertainty set. To compare metrics across environments, the score v of each method was normalized relative to the reference score of TD3. TD3 was trained on the environment using the reference transition kernel, and its score is denoted as v_{TD3}. The M2TD3 score, v_{M2TD3}, was used as the comparison target. The formula used to get a normalized score is $(v - v_{TD3})/(|v_{M2TD3} - v_{TD3}|)$. This defines v_{TD3} as the minimum baseline and v_{M2TD3} as the target. This standardization provides a metric that quantifies the improvement of each method over TD3 relative to the improvement of M2TD3 over TD3. Non-normalized results are available in Appendix 36. As expected, M2TD3, RARL and DR perform better in terms of worst-case performance, than vanilla TD3. Surprisingly, RARL is outperformed by DR except for HalfCheetah, Hopper, and Walker in worst-case performance. Finally, M2TD3, which is a state-of-the-art algorithm, outperforms all baselines except on HalfCheetah where DR achieves a slightly, non-statistically significant, better score. One potential explanation for the superior

performance of DR over robust reinforcement learning methods in the HalfCheetah environment is that the training of a conservative value function is not necessary. The HalfCheetah environment is inherently well-balanced, even with variations in mass or friction. Consequently, robust training, which typically aims to handle worst-case scenarios, becomes less critical. This insight aligns with the findings of Moskovitz et al. (2021), who observed similar results in this specific environment. The variance in the evaluations also needs to be addressed. In many environments, high variance prevents drawing statistical conclusions. For instance, HumanoidStandup shows a variance of 3.32 for M2TD3, complicating reliable performance assessments. Similar issues arise with DR in the same environment, showing a variance of 4.1. Such variances highlight the difficulty of making definitive comparisons across different robust reinforcement learning methods in these settings.

Table 7.3: Avg. of normalized static worst-case performance over 10 seeds for each method

	Ant	HalfCheetah	Hopper	HumanoidStandup	Walker	Average
TD3	0.0 ± 0.34	0.0 ± 0.06	0.0 ± 0.21	0.0 ± 2.27	0.0 ± 0.1	0.0 ± 0.6
DR	0.06 ± 0.16	$\mathbf{1.07 \pm 0.36}$	0.86 ± 0.82	0.04 ± 4.1	0.57 ± 0.37	0.52 ± 1.16
M2TD3	$\mathbf{1.0 \pm 0.27}$	1.0 ± 0.16	$\mathbf{1.0 \pm 0.65}$	$\mathbf{1.0 \pm 3.32}$	$\mathbf{1.0 \pm 0.63}$	$\mathbf{1.0 \pm 1.01}$
RARL	0.44 ± 0.3	0.13 ± 0.08	0.5 ± 0.22	0.44 ± 2.94	0.12 ± 0.09	0.33 ± 0.73
NR-MDP	-0.25 ± 0.1	-0.10 ± 0.24	-0.31 ± 0.4	-2.22 ± 1.51	-0.04 ± 0.01	-0.58 ± 0.45

Static average performance. Similarly to the worst-case performance described above, average scores across a uniform distribution on the uncertainty set are reported in Table 7.7. While robust policies explicitly optimize for the worst-case circumstances, one still desires that they perform well across all environments. A sound manner to evaluate this is to average their scores across a distribution of environments. First, one can observe that DR outperforms the other algorithms. This was expected since DR is specifically designed to optimize the policy on average across a (uniform) distribution of environments. One can also observe that RARL performs worse on average than a standard TD3 in most environments (except HumanoidStandup), despite having better worst-case scores. This exemplifies how robust RL algorithms can output policies that lack applicability in practice. Finally, M2TD3 is still better than TD3 on average, and hence this study confirms that it optimizes for worst-case performance while preserving the average score.

Table 7.4: Avg. of normalized static average case performance over 10 seeds for each method

	Ant	HalfCheetah	Hopper	HumanoidStandup	Walker	Average
TD3	0.0 ± 0.49	0.0 ± 0.22	0.0 ± 0.83	0.0 ± 1.36	0.0 ± 0.51	0.0 ± 0.68
DR	$\mathbf{1.65 \pm 0.05}$	$\mathbf{2.31 \pm 0.27}$	$\mathbf{2.08 \pm 0.49}$	$\mathbf{1.15 \pm 2.47}$	$\mathbf{1.22 \pm 0.34}$	$\mathbf{1.68 \pm 0.72}$
M2TD3	1.0 ± 0.11	1.0 ± 0.19	1.0 ± 0.55	1.0 ± 1.43	1.0 ± 0.65	1.0 ± 0.59
RARL	0.69 ± 0.13	-1.3 ± 0.54	-0.99 ± 0.11	0.47 ± 1.92	-0.35 ± 0.83	-0.3 ± 0.71
NR-MDP	0.44 ± 0.03	-0.58 ± 0.17	-0.85 ± 0.001	-0.83 ± 0.24	-1.08 ± 0.01	-0.58 ± 0.15

Dynamic adversaries. While the static and dynamic cases of transition kernel uncertainty

lead to the same robust value functions in the idealized framework of rectangular uncertainty sets, most real-life situations (such as those in RRLS) fall short of this rectangularity assumption. Consequently, Robust Value Iteration algorithms, which train an adversarial policy $\bar{\pi}$ (whether they store it or not) might possibly lead to a policy that differs from those which optimize for the original $\max_\pi \min_p$ problem introduced in Section 7.2. RRLS permits evaluating this feature by running rollouts of agent policies versus their adversaries, after optimization. RARL and NR-MDP simultaneously train a policy π and an adversary $\bar{\pi}$. The policy is evaluated against its adversary over ten episodes. Observations in Table 7.5 demonstrate how RRLS can be used to compare RARL and NR-MDP against their respective adversaries, in raw score. However, this comparison should not be interpreted as a dominance of one algorithm over the other, since the uncertainty sets they are trained upon are not the same.

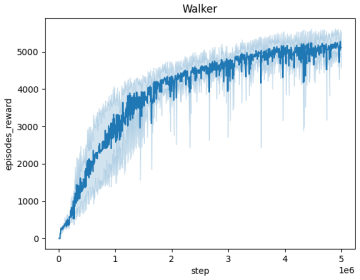
Table 7.5: Comparison of RARL and NR-MDP across different environments

Method	HumanoidStandup (10^4)	Ant (10^3)	HalfCheetah (10^2)	Hopper (10^3)	Walker (10^3)
RARL	9.84 ± 3.36	2.90 ± 0.70	-0.74 ± 6.69	1.04 ± 0.16	3.45 ± 1.13
NR-MDP	9.37 ± 0.14	5.58 ± 0.64	109.90 ± 4.74	3.14 ± 0.53	5.17 ± 0.89

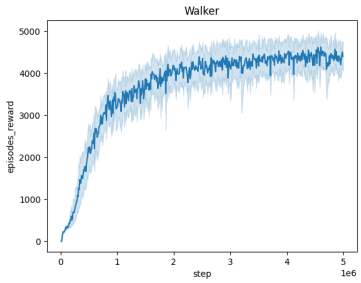
Training curves. Figure 7.4 reports training curves for TD3, DR, RARL, and M2TD3 on the Walker environment, using RRLS (results for all other environments in Appendix 35). Each agent was trained for 5 million steps, with cumulative rewards monitored over trajectories of 1,000 steps. Scores were averaged over 10 different seeds. The training curves illustrate the steep learning curve of TD3 and DR in the initial stages of learning, versus their robust counterparts. The M2TD3 agent ultimately achieves the highest performance at 5 million steps. Similarly, RARL exhibits a significant delay in learning, with stabilization occurring only toward the end of the training. Figures 7.4d and 7.4c show a significant variance in training across different random seeds. This emphasizes the difficulty of comparing different robust reinforcement learning methods along training.

A comparison of algorithms of Chapters 5 and 6 In tables 7.7 and 7.6, we have reported the normalised scores of algorithms `ExpectRL` and TC-MDPs presented in Chapters 5 and 6 such that the score is defined as $(v - v_{TD3})/(|v_{M2TD3} - v_{TD3}|)$. The results are for tasks Ant3, Hopper3, Walker3, etc... where three physical parameters are changing at the time on evaluation. In both tables 7.7 and 7.6, we have separated the performance of oracle algorithms, with the highest values highlighted in green, and the performance of non-oracle algorithms, with the best values underlined in black. The results are as follows:

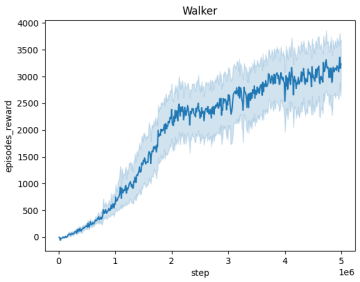
- **In terms of worst-case performance in table 7.6:** the results of the TC-MDP oracle algorithm are the most optimal, as it leverages additional information that is typically unavailable in practical settings. However, M2TD3 performs very well in practice since it is designed to effectively minimize the worst-case scenarios. The performance of both the TC-MDP stack and the classical TC-MDP algorithm is also strong, though slightly lower than that of M2TD3.
- **In terms of average performance in table 7.7:** the average performance of the DR+ `ExpectRL` algorithm significantly surpasses that of M2TD3. In the context of DR+ `ExpectRL`, adding a distributional robustness component is highly beneficial. We also observe that the TC-MDP algorithms achieve considerably superior performance. The



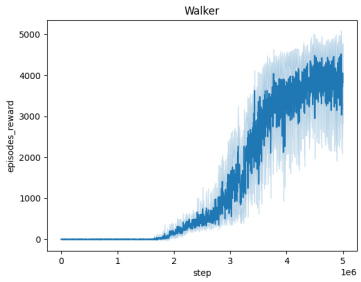
(a) Training curve on Walker with TD3



(b) Training curve on Walker with DR



(c) Training curve on Walker with RARL



(d) Training curve on Walker with M2TD3

Figure 7.4: Averaged training curves for Walker over 10 seeds

inclusion of adversarial constraints leads to a less pessimistic adversary, which in turn improves the average performance. An alternative interpretation is that this approach considers non-rectangular uncertainty sets with dynamics constrained to be Lipschitz-continuous.

In general, the conclusions regarding these methods are as follows: M2TD3 has lower variance compared to the other algorithms and performs very well in terms of worst-case performance. However, TC-MDP offers a better balance, with strong mean performance while still maintaining good worst-case results. Finally, the ExpectRL algorithm is simpler than the others, as it utilizes only a single network, and while it performs slightly lower in terms of worst-case performance, it achieves strong results for mean performance.

7.6 Conclusion

This Chapter introduces the Robust Reinforcement Learning Suite (RRLS), a benchmark for evaluating robust RL algorithms, based on the Gymnasium API. RRLS provides a consistent framework for testing state-of-the-art methods, ensuring reproducibility and comparability. RRLS features six continuous control tasks based on Mujoco environments, each with predefined uncertainty sets for training and evaluation, and is designed to be expandable to more environments and uncertainty sets. This variety allows comprehensive testing across various adversarial conditions. We also offer four compatible baselines and demonstrate their performance in static settings. Our work enables systematic comparisons of algorithms based on practical performance. RRLS addresses the need for reproducibility and comparability in robust RL. By making the source code publicly available, we anticipate that RRLS will become a valuable resource for the RL community, promoting progress in robust reinforcement learning algorithms.

	Ant	HalfCheetah	Hopper	Humanoid	Walker	Agg
Oracle M2TD3	**1.02 ± 0.19**	0.34 ± 0.23	0.97 ± 0.55	**3.9 ± 3.65**	0.3 ± 0.45	**1.31 ± 1.01**
Oracle RARL	0.62 ± 0.32	0.1 ± 0.02	0.48 ± 0.19	-2.59 ± 2.18	0.16 ± 0.21	-0.25 ± 0.58
`Oracle-TC` -M2TD3	0.1 ± 0.25	**1.87 ± 0.1**	0.49 ± 1.07	-0.8 ± 3.05	0.28 ± 0.38	0.39 ± 0.97
`Oracle-TC` -RARL	0.59 ± 0.36	1.55 ± 0.35	0.4 ± 0.16	1.19 ± 1.24	0.56 ± 0.39	0.86 ± 0.5
`Stacked-TC` -M2TD3	-0.05 ± 0.09	**1.56 ± 0.16**	1.08 ± 0.89	-0.83 ± 2.62	1.12 ± 0.5	0.58 ± 0.85
`Stacked-TC` -RARL	0.07 ± 0.13	0.76 ± 0.34	**1.35 ± 0.93**	**1.75 ± 2.48**	0.67 ± 0.32	0.92 ± 0.84
`TC` -M2TD3	-0.06 ± 0.08	1.49 ± 0.23	1.29 ± 0.29	1.21 ± 2.44	**1.19 ± 0.34**	**1.02 ± 0.68**
`TC` -RARL	0.14 ± 0.24	0.89 ± 0.3	1.5 ± 0.76	1.4 ± 4.57	0.67 ± 0.59	0.92 ± 1.29
TD3	0.0 ± 0.34	0.0 ± 0.06	0.0 ± 0.21	0.0 ± 2.27	0.0 ± 0.1	0.0 ± 0.6
DR	0.06 ± 0.16	1.07 ± 0.36	0.86 ± 0.82	0.04 ± 4.1	0.57 ± 0.37	0.52 ± 1.16
M2TD3	**1.0 ± 0.27**	1.0 ± 0.16	1.0 ± 0.65	1.0 ± 3.32	1.0 ± 0.63	1.0 ± 1.01
RARL	0.44 ± 0.3	0.13 ± 0.08	0.5 ± 0.22	0.44 ± 2.94	0.12 ± 0.09	0.33 ± 0.73
ExpecRL + DR	0.74 ± 0.31	0.88 ± 0.29	1.09 ± 0.31	1.12 ± 2.49	0.85 ± 0.60	0.93 ± 0.79

Table 7.6: Avg. of normalized static worst-case performance over 10 seeds for each method

	Ant	HalfCheetah	Hopper	Humanoid	Walker	Agg
Oracle M2TD3	1.13 ± 0.08	1.56 ± 0.24	1.12 ± 0.46	1.96 ± 1.53	1.23 ± 0.3	1.4 ± 0.52
Oracle RARL	0.7 ± 0.22	-1.4 ± 0.13	-0.77 ± 0.24	-2.6 ± 2.88	-1.13 ± 0.84	-1.04 ± 0.86
Oracle-TC -M2TD3	1.73 ± 0.09	$\mathbf{4.35 \pm 0.26}$	$\mathbf{5.54 \pm 0.13}$	2.12 ± 1.4	1.84 ± 0.37	$\mathbf{3.12 \pm 0.45}$
Oracle-TC -RARL	$\mathbf{1.78 \pm 0.02}$	4.32 ± 0.21	5.08 ± 0.48	0.42 ± 2.9	1.68 ± 0.24	2.66 ± 0.77
Stacked-TC -M2TD3	1.45 ± 0.38	$\mathbf{3.78 \pm 0.29}$	$\mathbf{5.2 \pm 0.29}$	-1.38 ± 1.67	$\mathbf{2.11 \pm 0.52}$	2.23 ± 0.63
Stacked-TC -RARL	1.52 ± 0.11	2.29 ± 0.23	2.91 ± 0.67	1.14 ± 2.19	1.21 ± 0.46	1.81 ± 0.73
TC -M2TD3	1.6 ± 0.06	3.71 ± 0.24	4.4 ± 0.6	$\mathbf{3.28 \pm 2.52}$	1.56 ± 0.23	$\mathbf{2.91 \pm 0.73}$
TC -RARL	$\mathbf{1.67 \pm 0.07}$	2.27 ± 0.22	1.79 ± 0.53	0.89 ± 2.19	1.01 ± 0.21	1.53 ± 0.64
TD3	0.0 ± 0.49	0.0 ± 0.22	0.0 ± 0.83	0.0 ± 1.36	0.0 ± 0.51	0.0 ± 0.68
DR	1.65 ± 0.05	2.31 ± 0.27	2.08 ± 0.49	1.15 ± 2.47	1.22 ± 0.34	1.68 ± 0.72
M2TD3	1.0 ± 0.11	1.0 ± 0.19	1.0 ± 0.55	1.0 ± 1.43	1.0 ± 0.65	1.0 ± 0.59
RARL	0.69 ± 0.13	-1.3 ± 0.54	-0.99 ± 0.11	0.47 ± 1.92	-0.35 ± 0.83	-0.3 ± 0.71
ExpecRL + DR	1.08 ± 0.41	1.17 ± 0.35	2.61 ± 0.66	1.05 ± 1.42	1.02 ± 0.5	1.38 ± 0.67

Table 7.7: Avg. of normalized static average case performance over 10 seeds for each method

Part III

Bandit Theory

Chapter 8

VITS : Variational Inference Thompson Sampling for contextual bandits

Contents

8.1 Introduction

In traditional Multi-Armed Bandit (MAB) problems, an agent, has to sequentially choose between several actions (referred to as "arms"), from which he receives a reward from the environment. The arm selection process is induced by a sequence of policies, which is inferred and refined at each round from past observations. These policies are designed to optimize the cumulative rewards over the entire process. The main challenge in this task is to effectively manage a suitable exploitation and exploration trade-off (Robbins 1952, Katehakis and Veinott 1987, Berry and Fristedt 1985, Auer et al. 2002, Lattimore and Szepesvári 2020, Kveton et al. 2020). Here, exploitation refers to selecting an arm that is currently believed to be the best based on past observations, while exploration refers to selecting arms that have not been selected frequently in the past in order to gather more information.

Contextual bandit problems is a particular instance of MAB problem, which supposes, at each round, that the set of arms and the corresponding reward depend on a d-dimensional feature vector called a contextual vector or context. This scenario has been extensively studied over the past decades and learning algorithms have been developed to address this problem (Langford and Zhang 2007, Abbasi-Yadkori et al. 2011, Agrawal and Goyal 2013, Kveton et al. 2020), and they have been successfully applied in several real-world problem such as recommender systems, mobile health and finance (Li et al. 2010, Agarwal et al. 2016, Tewari and Murphy 2017, Bouneffouf et al. 2020). The existing algorithms for addressing contextual bandit problems can be broadly categorized into two groups. The first category is based on maximum likelihood and the principle of optimism in the face of uncertainty (OFU) and has been studied in (Auer et al.

2002, Chu et al. 2011, Abbasi-Yadkori et al. 2011, Li et al. 2017, Ménard and Garivier 2017, Zhou et al. 2020, Foster and Rakhlin 2020, Zenati et al. 2022).

The second category consists in randomized probability matching algorithms, which is based on Bayesian belief and posterior sampling. Thompson Sampling (TS) is one of the most famous algorithms that fall into this latter category. Since its introduction by Thompson (1933), it has been widely studied, both theoretically and empirically (Agrawal and Goyal 2012, Kaufmann et al. 2012, Agrawal and Goyal 2013, Russo and Van Roy 2014; 2016, Lu and Van Roy 2017, Riquelme et al. 2018, Jin et al. 2021). Despite the fact that OFU algorithms offer better theoretical guarantees compared to classic TS-based algorithms, traditional TS methodologies still appeal to us due to their straightforward implementation and empirical advantages. In Agrawal and Goyal (2012), the authors claimed that: "In applications like display advertising and news article recommendation, TS is competitive with or better than popular methods such as UCB". Similarly, Chapelle and Li (2011) has examined the empirical performances of TS on both simulated and real data. Their experiments demonstrate that TS outperforms OFU methods, leading them to conclude: "In any case, TS is very easy to implement and should thus be considered as a standard baseline". Taking all these factors into account, we have decided to focus on TS-based algorithms for addressing contextual bandit problems.

Despite its relative simplicity, effectiveness and convergence guarantees, TS comes with a computational burden which is to sample, at each iteration $t \in \mathbb{N}^\star$, from an appropriate Bayesian posterior distribution $\hat{\mathrm{p}}_t$ defined from the previous observations. Indeed, these posteriors are usually intractable and approximate inference methods have to be used to obtain samples with distributions "close" to the posterior. The family of TS methods using approximate inference methods will be referred to as approximate inference TS in the sequel. Among the simplest approximate inference methods, Laplace approximation has been proposed for TS in Chapelle and Li (2011). This method consists of approximating the posterior distribution $\hat{\mathrm{p}}_t$ by a Gaussian distribution with a carefully chosen mean and covariance matrix. More precisely, the mean is a mode of the target distribution which is typically found using an optimization algorithm, while the covariance matrix is taken to be the negative Hessian matrix of the log posterior at the considered mode. Despite this method is easy to implement, it may lead to poor posterior representations. Indeed, while Laplace method achieves minimal optimality in terms of regret (Faury et al. 2022), it doesn't dictate the posterior convergence rate. More precisely, in Katsevich and Rigollet (2023) it has been demonstrated that VI outperforms Laplace in terms of mean convergence by a factor of $1/n$. It is worth noting that the covariance rates remain the same for both methods. This discrepancy can lead to inadequate approximations, especially in high-dimensional settings, as highlighted in section I.4 of Katsevich and Rigollet (2023).

Another class of popular approximate inference methods are Markov Chain Monte Carlo (MCMC) methods, such as Metropolis or Langevin Monte Carlo (LMC) algorithms. In the bandit literature, LMC has been proposed to get approximate samples from TS posteriors for solving traditional bandit problem in Mazumdar et al. (2020) and for contextual bandit problems in Xu et al. (2022), Huix et al. (2023). Also, Lu and Van Roy (2017) have proposed to adapt Ensemble Methods to the bandit setting. Roughly, the idea here is to maintain and incrementally update an ensemble of statistically plausible models and to draw a uniform sample from this family at each iteration.

Finally, Variational Inference (VI) (Blei et al. 2017) is another class of approximate method that could be used to get samples from the posterior distribution. The core concept behind VI is to find a distribution $\tilde{\mathrm{q}}$, referred to as the variational posterior, to closely match the true posterior $\hat{\mathrm{p}}$ in terms of Kullback-Leibler divergence (KL) within a predefined family of distributions known as the variational family $\mathcal{G}$. In general, the variational family is chosen to make the optimization of the KL tractable and to be easy to sample from. In their work Urteaga

and Wiggins (2018) propose the mean-field mixture of Gaussian variational family for TS. This family of distributions is quite extensive and provides an accurate approximation for a wide range of posterior distributions. However, in our perspective, it might not be the most suitable choice for TS. Firstly, the optimization algorithm at each time step can be computationally expensive. Secondly, the mean-field assumption assumes that the parameters are independent, a premise that holds true in the regime of large, overparameterized models. In our perspective, this regime may not align with the Bandit problem, which often operates in a setting where the number of data points tends towards infinity in comparison to the model size. Finally, Yu et al. (2020) also employs VI in more general graphical models but focuses on structured arms and rewards, where the rewards are correlated through latent variables.

In this Chapter, we develop an efficient VI method that makes use of the whole family of non-degenerate Gaussian distributions. This choice of VI family is supported by the Bernstein-Von Mises theorem (Van der Vaart 2000) . This theorem, subject to specific regularity conditions, asserts that a properly scaled version of the posterior converges to a Gaussian as the sample size grows. When applied to contextual bandits, the data points progressively accumulate over time, leading to the gradual concentration of the posterior around a dominant mode. As a consequence, the Gaussian approximation becomes increasingly suitable for representing the posterior in this particular setting. Furthermore, the covariance of the rescaled posterior distribution tends to converge towards the inverse Fisher information matrix, which may not necessarily be diagonal, thus justifying the need for a non-mean-field hypothesis. Our main contributions can be summarized as follows:

Our **first contribution** is methodological. We develop a novel variant of the TS algorithm, referred to as Variational Inference TS (`VITS`). Our method addresses the main challenges encountered by the existing approximate TS algorithms and can be applied to a very large class of TS posteriors. Moreover, it enjoys a low computational cost both theoretically and empirically, since it boils down to adding a few optimization steps per round. We also propose two approximate versions of `VITS`, called `VITS - II` and `VITS - II Hessian-free`, that scale with the problem dimension.

Our **second contribution** is theoretical. We establish that our proposed methodology achieves a sub-linear regret of order $\tilde{\mathcal{O}}(d^{3/2}\sqrt{T})$ (up to logarithmic term) in the linear contextual bandit framework, where T is the number of rounds and d is the dimension of the policy parameter. To the best of our knowledge, this is the first regret bound derived for VI in the context of sequential learning.

Finally, our **last contribution** is to illustrate the empirical performances of our method on a synthetic and on the real world dataset MovieLens (Lam and Herlocker (Lam and Herlocker)). It has been shown that in many cases, `VITS` outperforms existing approximate TS algorithms such as LMC algorithm.

Related work. The theoretical foundations of TS for linear contextual bandits were initially explored by Agrawal and Goyal (2013). In this paper, the authors establish a sub-linear cumulative regret bound $\tilde{\mathcal{O}}(d^{3/2}\sqrt{T})$ for Linear TS (Lin-TS). Compared to this study, our method achieves a similar regret bound in the linear framework. However, it should be noted that Lin-TS is a specialized algorithm that can be only used when the posterior is known and can be efficiently sampled from.

As mentioned previously, VI has been suggested for TS in Urteaga and Wiggins (2018). This paper introduces a TS algorithm called VTS that utilizes a mixture of mean-field Gaussian distributions to approximate the sequence of posteriors. In comparison to this work, the setting and the variational family we consider are richer than Urteaga and Wiggins (2018). A more

detailed comparison is postponed in Appendix 42. Moreover, the methodology developed in Urteaga and Wiggins (2018) does not come with any convergence guarantees. An empirical and theoretical study of using LMC as approximate inference method for TS for contextual bandit problems was carried out in Xu et al. (2022). This paper establishes that the resulting algorithm, called LMC-TS, achieves a state-of-the-art sub-linear cumulative regret for linear contextual bandits. Compared to this method, our approach yields a similar sub-linear regret in the same setting. Finally, Zhang et al. (2020) suggests a TS method based on Neural Tangent Kernel. While this performs well on real datasets, their method is much more expensive than previously mentioned approaches, as it requires training a neural network.

Notation. For $n \geq 1$, $[n]$ represents the set of integers between 1 and n. $\mathrm{N}(\mu, \Sigma)$ denotes the d-multidimensional Gaussian probability distribution with mean $\mu \in \mathbb{R}^d$ and covariance matrix $\Sigma \in \mathbb{R}^{d\times d}$. The transpose of a matrix M is denoted by $M^\top$. For any symmetric-real matrix A, $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ represent the maximum and minimum eigenvalues of A respectively. The norm $\|\cdot\|_2$ will refer to the 2-norm for vectors, and the operator norm for matrices. For any semi-definite positive matrix A, the norm $\|x\|_A$ denotes the Mahalanobis norm, i.e., $\|x\|_A = \sqrt{xAx^\top}$. For any event E on a probability space, $\bar{\mathsf{E}}$ refers to the complementary of E. Finally, $\mathbb{1}$ is the indicator function and tr is the trace of a matrix.

8.2 Thompson sampling for contextual bandits

Contextual bandit: We now present in more details the contextual bandit framework. Let $\mathcal{S}$ be a contextual space and consider $\mathsf{A} : \mathcal{S} \to 2^{\mathcal{A}}$ a set-valued action map, where $2^{\mathcal{A}}$ stands for the power set of the action space $\mathcal{A}$. For simplicity, we assume here that $\sup_{x\in\mathcal{S}} \mathrm{Card}(\mathsf{A}(s)) < +\infty$. A (deterministic or random) function $\pi : \mathcal{S} \to \mathcal{A}$ is said to be a policy if for any $s \in \mathcal{S}$, $\pi(s) \in \mathsf{A}(s)$. Then, for a fixed horizon $T \in \mathbb{N}^\star$, a contextual bandit process can be defined as follows: at each iteration $t \in [T]$ and given the past observations $\mathrm{D}_{t-1} = \{(s_i, a_s, s_i)\}_{i<t}$:

- The agent receives a contextual feature $s_t \in \mathcal{S}$;
- The agent chooses an action $a_t = \pi_t(s_t)$ where π_t is a policy sampled from $\mathbb{Q}_t(\cdot|\mathrm{D}_{t-1})$;
- Finally, the agent receives a reward r_t sampled from $\mathcal{R}(\cdot|s_t, a_t)$ given D_{t-1}. Here, $\mathcal{R}$ is a Markov kernel on $(\mathcal{A} \times \mathcal{S}) \times \mathsf{R}$, where $\mathsf{R} \subset \mathbb{R}$

For a fixed family of conditional distributions $\mathbb{Q}_{1:T} = \{\mathbb{Q}_t\}_{t\leq T}$, this process defines a random sequence of policies, $\pi_{1:T} = \{\pi_t\}_{t\leq T}$ with distribution still denoted by $\mathbb{Q}_{1:T}$ by abuse of notation. Let's defined the optimal expected reward for a contextual vector $s \in \mathsf{X}$ and the expected reward given x and any action $a \in \mathcal{A}(s)$ as follow

$$f_\star(s) = \max_{a\in\mathsf{A}(s)} f(s,a) \, , f(s,a) = \int r\mathcal{R}(\mathrm{d}r|s,a) \, . \tag{8.1}$$

The main challenge of a contextual bandit problem is to find the distribution $\mathbb{Q}_{1:T}$ that minimizes the cumulative regret defined as

$$\begin{aligned} \mathrm{CRegret}(\mathbb{Q}_{1:T}) &= \textstyle\sum_{i\leq T} \mathrm{Regret}_i^{\pi_i} \\ \text{with} \quad \mathrm{Regret}_i^{\pi_i} &= f_\star(s_i) - f(s_i, \pi_i(s_i)). \end{aligned} \tag{8.2}$$

The main difficulty in the contextual bandit problem, comes from the fact that the reward distribution $\mathcal{R}$ is intractable and must be inferred to find the best policy to minimize the

instantaneous regret $\pi \mapsto f_\star(s) - f(s, \pi(s))$ for a context $s \in \mathcal{S}$. However, the estimation of $\mathcal{R}$ may be in contradiction with the primary objective to minimize the cumulative regret (8.2), since potential non-effective arms has to be chosen to obtain a complete description of $\mathcal{R}$. Therefore, bandit learning algorithms have to achieve an appropriate trade-off between exploitation of arms which have been confidently learned and exploration of misestimated arms.

Thompson sampling: To achieve such a trade-off, we consider the popular Thompson Sampling (TS) algorithm. Consider a parametric model $\{\mathcal{R}_\theta : \theta \in \mathbb{R}^d\}$ for the reward distribution, where for any θ, $\mathcal{R}_\theta$ is a Markov kernel on $(\mathcal{A} \times \mathcal{S}) \times \mathsf{R}$ parameterized by $\theta \in \mathbb{R}^d$. We assume in this paper that $\mathcal{R}_\theta$ admits a density with respect to some dominating measure λ_{ref}. An important example are generalized linear bandits Filippi et al. (2010), Kveton et al. (2020). In particular, it assumes that $\{\mathcal{R}_\theta(\cdot|s,a) : \theta \in \Theta\}$ is an exponential family with respect to λ_{ref}, i.e., for $s \in \mathsf{X}$ and $a \in \mathcal{A}$,

$$\frac{\mathrm{d}\mathcal{R}_\theta}{\mathrm{d}\lambda_{\text{ref}}}(r|s,a) = \mathrm{h}(r)\exp(g(\theta,s,a)\mathrm{T}(r) - \mathrm{C}(\theta,s,a)), \tag{8.3}$$

for $h : \mathsf{R} \to \mathbb{R}_+$, natural parameter and log-partition function $g, \mathrm{C} : \mathbb{R}^d \times \mathsf{X} \times \mathcal{A} \to \mathbb{R}$ and sufficient statistics $\mathrm{T} : \mathsf{R} \to \mathbb{R}$. The family is said to be in canonical form if $g(\theta,s,a) = \langle \phi(s,a), \theta \rangle$ for some feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and $\mathrm{C}(\theta,s,a) = \sigma(\langle \phi(s,a), \theta \rangle)$ for some link function σ. Linear contextual bandits Chu et al. (2011), Abbasi-Yadkori et al. (2011) fall into this model taking $\lambda_{\text{ref}} = \text{Leb}$, T equals to the identity function,

$$\mathrm{h}(r) = \exp\Big(-\eta r^2/2\Big) \text{ and } g(\theta,s,a) = \eta \, \langle \phi(s,a), \theta \rangle, \tag{8.4}$$

for some $\eta > 0$. As a result, $\mathcal{R}_\theta(\cdot|s,a)$ is simply the Gaussian distribution with mean $\langle \phi(s,a), \theta \rangle$ and variance $1/\eta$. Finally Riquelme et al. (2018), Zhou et al. (2020), Xu et al. (2020) introduced an extension of linear contextual bandits, referred to as linear neural contextual bandits where g is a neural network with weights θ and taking as input a pair (x,a). With the introduced notations, the likelihood function associated to the observations D_t at step $t > 1$ is given by

$$\mathrm{L}_t(\theta) \propto \exp\left\{ \sum_{i=1}^{t-1} \ell(\theta|s_i,a_i,r_i) \right\} , \tag{8.5}$$

where the log-likelihood is given by $\ell(\theta|s_i,a_i,r_i) = \log(\mathrm{d}\mathcal{R}_\theta/\mathrm{d}\lambda_{\text{ref}})(r_i|x_i,a_i)$. Choosing a prior on θ with density p_0 with respect to Leb, and applying Bayes formula, the posterior distribution at round $t \in [T]$ is given by

$$\hat{\mathrm{p}}_t = \mathrm{L}_t(\theta)\mathrm{p}_0(\theta)/\mathfrak{Z}_t \tag{8.6}$$

where $\mathfrak{Z}_t = \int \mathrm{L}_t(\theta)\mathrm{p}_0(\theta)\mathrm{d}\theta$ denotes the normalizing constant and we used the convention that $\hat{\mathrm{p}}_1 = \mathrm{p}_0$. Moreover we define the potential function $U(\theta) \propto -\log \hat{\mathrm{p}}_t(\theta)$. Then, at each iteration $t \in [T]$, TS consists in sampling a sample θ_t from the posterior $\hat{\mathrm{p}}_t$ and from it, use as a policy, $\pi_t^{(\mathrm{TS})}(s)$ defined for any x by

$$\pi_t^{(\mathrm{TS})}(s) = a^{\theta_t}(s) \, , a^\theta(s) = \arg\max_a \int r\mathcal{R}_\theta(\mathrm{d}r|s,a) \tag{8.7}$$

Since $\mathfrak{Z}_t$ is generally intractable, sampling from the posterior distribution is not in general an option.

Variational inference TS: To address this challenge, practitioners often employ approximate inference methods to generate samples from a distribution that is expected to be "close" to the actual posterior distribution. In this context, we specifically concentrate on the application of VI.

In this scenario, we consider a variational family $\mathcal{G}$ which is a set of probability densities with respect to the Lebesgue measure, from which it is typically easy to sample from. Then ideally, at each round $t \in [T]$, the posterior distribution $\hat{\mathrm{p}}_t$ is approximated by the variational posterior distribution $\tilde{\mathrm{q}}_t$ which is defined as:

$$\tilde{\mathrm{q}}_t = \arg\min_{\mathrm{p}\in\mathcal{G}} \mathrm{KL}(\mathrm{p}|\hat{\mathrm{p}}_t) \ , \tag{8.8}$$

where KL is the Kullback-Leibler divergence. However, we have to determine at each round a solution to the problem specified in (8.8). In this paper, we consider as variational family the set of non-degenerate Gaussian distribution $\mathcal{G} = \{\mathrm{N}(\mu, \Sigma) \ : \ \mu \in \mathbb{R}^d, \ \Sigma \in \mathcal{S}_+^*\}$ where $\mathrm{N}(\mu, \Sigma)$ is the Gaussian distribution with mean μ and covariance matrix Σ and $\mathcal{S}_+^*$ is the set of symmetric positive definite matrices. As explained in the introduction, this Gaussian variational family is particularly relevant in bandit framework according to Bernstein-Von Mises theorem.

Presentation of VITS − I: As we will see, this choice of variational family will allow to derive an efficient method for solving (8.8) using the Riemannian structure of $\mathcal{G}$. As noted in Lambert et al. (2022), $\mathcal{G}$ equipped with the Wasserstein distance of order 2 is a complete metric space as a closed subset of $\mathcal{P}_2(\mathbb{R}^d)$, the set of probability distributions with finite second moment. Recall that for two Gaussian distributions $p_0 = \mathrm{N}(\mu_0, \Sigma_0)$ and $p_1 = \mathrm{N}(\mu_1, \Sigma_1)$, their Wasserstein distance has a closed form:

$$W_2^2\left(p_0, p_1\right) = \|\mu_0 - \mu_1\|^2 + \mathrm{tr}(\Sigma_0 + \Sigma_1 - 2(\Sigma_0^{1/2}\Sigma_1\Sigma_0^{1/2})^{1/2}) \ .$$

This Wasserstein distance on $\mathcal{G}$ allows to derive a Riemannian metric denoted g. The corresponding geodesic is given through the exponential map. More precisely, for a Gaussian distribution $p = \mathrm{N}(\mu_p, \Sigma_p)$, this map is defined as

$$\exp_p(\mu_v, \Sigma_v) = (\mu_p + \mu_v + (\Sigma_v + \mathrm{I})(\cdot - \mu_p))_{\#}p = \mathrm{N}(\mu_p + \mu_v, \ (\Sigma_v + \mathrm{I})\ \Sigma_p\ (\Sigma_v + \mathrm{I})) \ . \tag{8.9}$$

With all these preliminaries, we can now present and motivate the algorithm developed in Lambert et al. (2022) to efficiently solve (8.8). This method can be formalized as a Riemannian gradient descent scheme on $\mathcal{G}$. Firstly, we define the loss function $\mathcal{F}_t : p \to \mathrm{KL}(p|\hat{\mathrm{p}}_t)$. Then, following Lambert et al. (2022), we derive the gradient operator of $\mathcal{F}_t$ on $\mathcal{G}$ equipped with g as

$$\nabla_g \mathcal{F}_t(p) = (\int \nabla U_t(\theta)\mathrm{d}p(\theta), \ \int \nabla^2 U_t(\theta)\mathrm{d}p(\theta) - \Sigma_p^{-1}) \tag{8.10}$$

where Σ_p is the covariance matrix of p. From this expression, the corresponding Riemannian gradient descent Bonnabel (2013) using a step size $h_t > 0$ defines the sequence of iterates $\{\mathrm{q}_{t,k}\}_{k=1}^{K_t}$ recursively as:

$$\mathrm{q}_{t,k+1} = \exp_{\mathrm{q}_{t,k}}(-h_t \nabla_g \mathcal{F}_t(\mathrm{q}_{t,k})) \ .$$

At each time step t, this sequence is initialized with variational posterior at the previous step, ie, $\mathrm{q}_{t,0} = \mathrm{q}_{t-1,K_{t-1}}$. Please note that this warm initialization of the posterior results in an efficient algorithm and has been directly used in our main theoretical result (see (A.376)). Combining (8.9) and (8.10), this recursion amounts defining a sequence of means $\{\mu_{t,k}\}_{k=1}^{K_t}$ and covariance matrices $\{\Sigma_{t,k}\}_{k=1}^{K_t}$ by the recursions

$$\mu_{t,k+1} = \mu_{t,k} - h_t \int \nabla U_t(\theta)\mathrm{dq}_{t,k}(\theta),$$

$$\Sigma_{t,k+1} = A_{t,k}\Sigma_{t,k}A_{t,k}, \mathrm{q}_{t,k+1} = \mathrm{N}(\mu_{t,k+1}, \Sigma_{t,k+1}) \text{where} \quad A_{t,k} = \mathrm{I} - h_t(\int \nabla^2 U_t(\theta)\mathrm{dq}_{t,k}(\theta) - \Sigma_{t,k}^{-1})$$

The main computational challenge in this recursion stems is that the integrals involved are typically intractable. To overcome this issue, we employ a Monte Carlo procedure to approximate these integrals. Subsequently, we consider a sequence of mean values denoted as $\{\tilde{\mu}_{t,k}\}_{k=1}^{K_t}$ and covariance matrices $\{\tilde{\Sigma}_{t,k}\}_{k=1}^{K_1}$ such that:

$$\begin{aligned}&\tilde{\mu}_{t,k+1} = \tilde{\mu}_{t,k} - h_t \nabla U_t(\tilde{\theta}_{t,k}), \quad \tilde{\Sigma}_{t,k+1} = \tilde{A}_{t,k}\tilde{\Sigma}_{t,k}\,\tilde{A}_{t,k} \\ &\text{with} \quad \tilde{A}_{t,k} = \mathrm{I} - h_t(\nabla^2 U_t(\tilde{\theta}_{t,k}) - \tilde{\Sigma}_{t,k}^{-1}) \,,\end{aligned}$$

where $\tilde{\theta}_{t,k} \sim \mathrm{N}(\tilde{\mu}_{t,k}, \tilde{\Sigma}_{t,k})$. Consequently, following Lambert et al. (2022) we obtain an algorithm capable of addressing the problem defined in (8.8). However, this algorithm exhibits computational inefficiency, particularly in high-dimensional scenarios. This inefficiency arises from the necessity to sample from a Gaussian distribution with a non-diagonal covariance matrix during each updating step $k \in [K_t]$. As a result, it becomes impractical for use in a contextual bandit problem, where, at each time step t, we must solve the problem described in (8.8). This paper introduces an improved version of the earlier algorithm, designed to efficiently address the problem presented in (8.8). To achieve this, we begin by examining a sequence of matrices denoted as $B_{t,k}$, defined by the following

$$B_{t,k+1} = \{\mathrm{I} - h_t \nabla^2 U_t(\tilde{\theta}_{t,k})\} B_{t,k} + h_t (B_{t,k}^{-1})^\top \,. \tag{8.11}$$

It is important to note that $B_{t,k}$ is a square-root matrix of the covariance of the variational distribution $\tilde{\Sigma}_{t,k}$, ie, $B_{t,k}B_{t,k}^\top = \tilde{\Sigma}_{t,k}$. Then we can sample efficiently from the variational distribution using $B_{t,k}$ with $\tilde{\theta}_{t,k} = \tilde{\mu}_{t,k} + B_{t,k}\epsilon_{t,k}$, $\epsilon \sim \mathrm{N}(0, \mathrm{I})$. As a result, note that our method does not require any Cholesky decomposition, which has a complexity of $\mathcal{O}(d^3)$, contrary to the algorithm derived in Lambert et al. (2022) and also in LinTS. The updating strategies for the sequence of $\tilde{\mu}_{t,k}$ and $B_{t,k}$ are given by

$$\begin{aligned}&\tilde{\mu}_{t,k+1} = \tilde{\mu}_{t,k} - h_t \nabla U_t(\tilde{\theta}_{t,k}); \quad B_{t,k+1} = \{\mathrm{I} - h_t \nabla^2 U_t(\tilde{\theta}_{t,k})\} B_{t,k} + h_t (B_{t,k}^{-1})^\top \\ &\tilde{\theta}_{t,k} \sim \mathrm{N}(\tilde{\mu}_{t,k}, B_{t,k}^\top B_{t,k}) \,.\end{aligned}$$

From this methodology, we can now complete the description of our first algorithm, referred to as `VITS-I`. At each step t, we consider the variational distribution $\tilde{\mathrm{q}}_t = \tilde{\mathrm{q}}_{t,K_t} = \mathrm{N}(\tilde{\mu}_{t,K_t}, B_{t,k}^\top B_{t,k})$ which approximates the solution of (8.8). Then, at round $t+1$, `VITS-I` consists in sampling $\tilde{\theta}_{t+1}$ according to $\tilde{\mathrm{q}}_t$ and choosing

$$\pi_{t+1}^{\texttt{VITS-I}}(s) = \underset{a \in \mathsf{A}(s)}{\arg\max}\, a^{\tilde{\theta}_{t+1}}(s) \,. \tag{8.12}$$

As in TS, the likelihood function and the posterior distribution $\hat{\mathrm{p}}_{t+1}$ are updated following equations (8.5) and (8.6) using the new observed reward r_{t+1} distributed according to $\mathcal{R}(\cdot|x_{t+1}, a_{t+1})$ with $a_{t+1} = \pi_{t+1}^{\texttt{VITS-I}}(x)$. The round $t+1$ is then concluded by solving $\tilde{\mathrm{q}}_{t+1} = \tilde{\mathrm{q}}_{t+1,K_{t+1}}$. The pseudo-code associated with this algorithm is given in Algorithm 7 and Algorithm 8.

Algorithm 7: `VITS` algorithm

$B_{1,1} = \mathrm{I}/\sqrt{\lambda\eta}$, $\tilde{W}_{1,1} = \mathrm{I}/(\eta\lambda)$,$\tilde{\mu}_{1,1} \sim \mathrm{N}(0, \tilde{W}_{1,1})$
for $t = 1, \ldots, T$ **do**
  receive $x_t \in \mathcal{S}$
  sample $\tilde{\theta}_t$ from $\tilde{\mathrm{q}}_{t,K_t} = \mathrm{N}(\tilde{\mu}_{t,K_t}, B_{t,K_t}^\top B_{t,k})$
  choose $a_t = \pi^{(\mathrm{VITS})}(x_t)$ presented in (8.12)
  receive $r_t \sim \mathcal{R}(\cdot|x_t, a_t)$
  update $\tilde{\mathrm{q}}_{t+1,K_{t+1}}$ using Alg. 8 or 9.
end for

Algorithm 8: VITS-I

Parameters: step-size h_t, number of iterations K_t
$\tilde{\mu}_{t,1} \leftarrow \tilde{\mu}_{t-1,K_{t-1}}$,$B_{t,1} \leftarrow B_{t-1,K_{t-1}}$
for $k = 1, \ldots, K_t$ **do**
  draw $\tilde{\theta}_{t,k} \sim \tilde{\mathrm{q}}_{t,k} = \mathrm{N}(\tilde{\mu}_{t,k}, B_{t,k}^\top B_{t,k})$
  $\tilde{\mu}_{t,k+1} \leftarrow \tilde{\mu}_{t,k} - h_t \nabla U_t(\tilde{\theta}_{t,k})$
  $B_{t,k+1} \leftarrow \{\mathrm{I} - h_t \nabla^2(U_t(\tilde{\theta}_{t,k}))\} B_{t,k} + h_t (B_{t,k}^{-1})^\top$
end for

Presentation of VITS-II: In high dimension, the computational cost of the recursion of mean values and covariance matrices may be prohibitive since at each iteration $k \in [K_t]$, it requires inverting the matrix $B_{t,k}$. To tackle this computational issue, we propose a new version of VITS. More precisely, the inverse of the square root covariance matrix $B_{t,k}^{-1}$ can be approximated using a first order Taylor expansion in h_t; see Appendix 41 for more details. We denote by $C_{t,k}$ the approximation of $B_{t,k}^{-1}$, and we obtain recursions for the sequence of $\{C_{t,k}\}_{k \le K_t}$ and $\{B_{t,k}\}_{k \le K_t}$ such that:

$$C_{t,k+1} = C_{t,k}\{\mathrm{I} - h_t(C_{t,k}^\top C_{t,k} - \nabla^2 U_t(\tilde{\theta}_{t,k}))\} ,$$
$$B_{t,k+1} = (\mathrm{I} - h_t \nabla^2 U_t(\tilde{\theta}_{t,k})) B_{t,k} + h_t C_{t,k}^\top .$$

This trick reduces the complexity from $\mathcal{O}(d^3)$ to $\mathcal{O}(d^2)$ for the computation of the inverse. This version of VITS is referred to as VITS − II and is given in Algorithm 7 and 9.

Presentation of VITS − II Hessian-free: The most computationally intensive step in VITS − II remains the computation of the Hessian of U_t. In scenarios with a large number of data points and high dimensions, this step can become highly demanding. To avoid computing the Hessian of U_t, we suggest to use the following property of Gaussian distribution which is the result of a simple integration by part:

$$\int \nabla^2 U_t \mathrm{dN}(\mu, \Sigma) = \int \Sigma^{-1}(\mathrm{I} - \mu) \nabla U_t^\top \, \mathrm{dN}(\mu, \Sigma) . \tag{8.13}$$

After approximating this right side integral using Monte Carlo, we derive a new sequence of square-root covariance matrix $\{B_{t,k}\}_{k \le K_t}$ and inverse square-root covariance matrix $\{C_{t,k}\}_{k \le K_t}$, defined recursively by:

$$C_{t,k+1} = C_{t,k}\{\mathrm{I} - h_t(C_{t,k}^\top C_{t,k} - A_{t,k})\} ,$$
$$B_{t,k+1} = (\mathrm{I} - h_t A_{t,k}) B_{t,k} + h_t C_{t,k}^\top ,$$

where $A_{t,k} = C_{t,k}^\top C_{t,k}(\tilde{\theta}_{t,k} - \tilde{\mu}_{t,k}) \nabla U_t^\top(\tilde{\theta}_{t,k})$ and $\tilde{\theta}_{t,k} \sim \mathrm{N}(\tilde{\mu}_{t,k}, B_{t,k}^\top B_{t,k})$. This last version of VITS is referred to as VITS − II Hessian-free and its pseudo-code is given in Algorithm 7 and Algorithm 9, where (H) and (H free) are for respectively Hessian and Hessian Free version. The computational complexity of all methods has been experimentally studied in a simple case, as discussed in Section 47.

8.3 Main results

8.3.1 Linear Bandit

In this section, we are interested in convergence guarantees for VITS − I applied to the linear contextual bandit framework. This framework consists in assuming that $\mathcal{R}_\theta$ has form (8.3) with

Algorithm 9: VITS − II / VITS − II Hessian-free

Parameters: step-size h_t, number of iterations K_t

$\tilde{\mu}_{t,1} \leftarrow \tilde{\mu}_{t-1,K_{t-1}}$, $B_{t,1} \leftarrow B_{t-1,K_{t-1}}$

for $k = 1, \ldots, K_t$ **do**

draw $\tilde{\theta}_{t,k} \sim \tilde{\mathrm{q}}_{t,k} = \mathrm{N}(\tilde{\mu}_{t,k}, B_{t,k}^\top B_{t,k})$

$\tilde{\mu}_{t,k+1} \leftarrow \tilde{\mu}_{t,k} - h_t \nabla U_t(\tilde{\theta}_{t,k})$

$$A_{t,k} = \begin{cases} \nabla^2(U_t(\tilde{\theta}_{t,k})) & \text{(Hessian)} \\ C_{t,k}^2(\tilde{\theta}_{t,k} - \tilde{\mu}_{t,k})(\nabla U_t(\tilde{\theta}_{t,k}))^\top & \text{(Hessian free)} \end{cases}$$

$B_{t,k+1} \leftarrow \{\mathrm{I} - h_t A_{t,k}\} B_{t,k} + h_t C_{t,k}^\top$

$C_{t,k+1} \leftarrow C_{t,k}(\mathrm{I} - h_t(C_{t,k}^\top C_{t,k} - A_{t,k}))$

end for

$\lambda_{\mathrm{ref}} = \mathrm{Leb}$, T is the identity function and h and g are specified by (8.4):

$$\frac{\mathrm{d}R_\theta}{\mathrm{dLeb}}(r|s,a) \propto \exp\Big[\eta(r - \langle\phi(s,a),\theta\rangle)^2/2\Big] \,. \tag{8.14}$$

Assumption on the reward kernel $\mathcal{R}$ is the following:

Assumption 8.3.1. *(Sub-Gaussian Reward Distribution) There exists $R > 1$ such that for any $s \in \mathcal{S}$, $a \in \mathsf{A}(s)$, $\rho > 0$, $\log \int \exp\{\rho(r - f(s,a))\}\mathcal{R}(\mathrm{d}r|s,a) \le R\rho^2$, where f is defined in 8.1*

We could only assume that $R > 0$ in Assumption 8.3.1 since if a distribution is R-sub-Gaussian, it is also R'-sub-Gaussian for any $R' \ge R$, however, we choose to set $R \ge 1$ to ease the presentation of our main results. We also assume that the model is well-specified.

Assumption 8.3.2. *There exists $\theta^\star$ such that $\mathcal{R} = \mathcal{R}_{\theta^\star}$ and satisfying $\|\theta^\star\|_2 \le 1$. Feature map ϕ satisfies the boundedness condition.*

Assumption 8.3.3. *For any contextual vector $x \in \mathbb{R}^d$ and action $a \in \mathsf{A}(s)$, it holds that $\|\phi(s,a)\|_2 \le 1$.*

Uniform boundedness condition on the feature map is relatively common for obtaining regret bounds for linear bandit problems (Agrawal and Goyal 2013, Xu et al. 2022, Kveton et al. 2020, Abbasi-Yadkori et al. 2011). Note that Assumption (8.3.3) is equivalent to $\sup_{s\in\mathsf{X},\, a\in\mathcal{A}(a)}\|\phi(s,a)\|_2 \le M_\phi$ for some arbitrary but fixed constant $M_\phi > 0$, changing the feature map ϕ by ϕ/M_ϕ. Finally, we specify the prior distribution.

Assumption 8.3.4. *The prior distribution is assumed to be zero-mean Gaussian distribution with variance $1/(\lambda\eta)$, where η also appears in the definition $\mathcal{R}_\theta$ in (8.14),*

While our theoretical results can readily be extended to accommodate a non-zero mean Gaussian prior, for the sake of simplicity, we have chosen to center the prior. Under Assumption 8.3.4, combining (8.6) and (8.14), the negative log posterior $-\log \hat{\mathrm{p}}_t$ denoted by U_t is given by

$$U_t(\theta) = \frac{\eta}{2}\left(\sum_{i=1}^{t-1}(\phi(a_i,s_i)^\top\theta - r_i)^2 + \lambda\|\theta\|_2^2\right) = \frac{\eta}{2}(\theta^\top V_t\theta - 2\theta^\top b_t + \sum_{i=1}^{t-1} r_i^2) \,, \tag{8.15}$$

$$V_t = \lambda\mathrm{I}_d + \sum_{i=1}^{t-1}\phi_i\phi_i^\top \in \mathbb{R}^{d\times d},\; b_t = \sum_{i=1}^{t-1} r_i\phi_i \in \mathbb{R}^{d\times 1}.$$

Therefore, it follows that the gradient of U_t is given by $\nabla U_t(\theta) = \eta(V_t\theta - b_t)$ and its hessian matrix is equal to $\nabla^2 U_t(\theta) = \eta V_t$. Consequently, we recover the well-known fact that the posterior is a Gaussian distribution with mean $\hat{\mu}_t = V_t^{-1} b_t$ and covariance matrix $\hat{\Sigma}_t = (\eta V_t)^{-1}$. Denote by $\tilde{\mathbb{Q}}_{1:T}$ the distribution on the sequence of policies induced by the sequence of variational posterior $\{\tilde{\mathrm{q}}_t = \mathrm{N}(\tilde{\mu}_{t,K_t}, B_{t,K_t}^\top B_{t,K_t})\}_{t\in[T]}$ obtained with VITS − I. We now state our main result on the cumulative regret associated to VITS − I for linear contextual bandit, where a the proof is provided in Appendix 39.

Theorem 8.3.5. *Assume Assumptions 8.3.1 to 8.3.4 hold. For the choice of hyperparameters $\{K_t, h_t\}_{t\in[T]}$ and η specified in Section 39.2, for any $\delta \in (0,1)$, with probability at least $1-\delta$, the cumulative regret is bounded by*

$$\mathrm{CRegret}(\tilde{\mathbb{Q}}_{1:T}) \leq \frac{CR^2 d\sqrt{dT}\log(3T^3)}{\lambda^2} \log\Big(\frac{(1+T/\lambda d)}{\delta}\Big)$$

where $C \geq 0$ is a constant independent of the problem. Our main result shows that the distribution of the sequence of policies generated by VITS − I results in a cumulative regret of order $\tilde{\mathcal{O}}(d\sqrt{dT})$. It is in the same order as the state-of-the-art cumulative regret obtained in Agrawal and Goyal (2013) for LinTS. The number of optimization steps K_t we found are of order $\kappa_t^2 \log(dT\log(T))$ where $\kappa_t = \lambda_{\max}(V_t)/\lambda_{\min}(V_t)$. Following (Hamidi and Bayati 2020, Wu et al. 2020), if the diverse context assumption holds, the condition number is $\kappa_t = \mathcal{O}(1)$. Therefore, under this previous assumption, VITS − I require a number of optimization steps that scale as $\log(dT\log(T))$. Finally, Xu et al. (2022) derived similar bounds for TS using LMC for linear contextual bandit problems. Although our proof is based on the linear case, it could be extended to more general cases insofar as our updates remain Gaussian by definition of the variational family. This allows the use of Gaussian (anti) concentration bound in the theoretical analysis. This is in contrast to other approximation methods, which do not possess this advantage.

Comparison table. In this paragraph we have added a comparison table between Linear TS (LinTS), Linear UCB (LinUCB), Feel-Good TS Huix et al. (2023), Zhang (2022), VITS − I, VITS − II (VITS-I/II), VITS − II Hessian-free (VITS-II HF), Langevin Monte Carlo TS (LMCTS) and Variational TS (VTS). The column "Regret" corresponds to the theoretical regret bound obtained by the algorithm. "Complexity" is the computational complexity, more precisely the symbol (++) corresponds to a regret $O(\sqrt{dT})$, (+) to $O(d^{3/2}\sqrt{T})$ and (−) to no existing regret bound. " Linear" is set to Yes when the algorithm is designed only for the Linear Bandit setting and No for general setting including Linear. The "Conditioning" column describes the algorithm's robustness against the conditioning of the problem.

8.4 Numerical experiments

8.4.1 Linear and quadratic bandit

Our initial investigation focused on a toy setting where contextual vectors are sampled from a Gaussian distribution. However, in this specific setting, the contextual vectors exhibit high diversity, resulting in a posterior covariance matrix with a condition number of $O(1)$. This condition makes the optimization problem overly simplistic, as a result, all approximation methods seem to perform identically in this simple well-conditioned problem. So we introduce a novel setting in which the diversity of arms is controlled by a parameter, denoted as ζ. Firstly, we consider a fixed pool of arms denoted as $P = [\tilde{s}_1, \ldots, \tilde{s}_n]$ with $n = 50$, where each arm $\tilde{s}_i$ follows a normal distribution $\mathrm{N}(0_d, \mathrm{I}_d)$. This fixed pool is relevant in real-world scenarios, such

	Regret	Complexity	Linear	Conditioning
LinTS	+	++	Yes	++
LinUCB	++	++	Yes	++
FG-TS	++		No	
VITS-I/II	+	+	No	+
VITS-II HF	-	+ +	No	+
LMC-TS	+	++	No	-
VTS	-	-	No	

as in a Recommender system, where this pool corresponds to the concept of a meta-user. Then, at each step $t \in [T]$, for every arm, we randomly sample a vector $\tilde{s}_i$ from the pool P, and the contextual vector associated with this arm is defined as $x = \tilde{s}_i + \zeta\epsilon$, where $\epsilon \sim \mathcal{N}(0_d, \mathrm{I}_d)$. When ζ has a high value, the corresponding user is far from the meta-user. Consequently, the diversity among arms is high, resulting in a well-conditioned problem. However, in cases where ζ is low, the problem is ill-conditioned and the optimization becomes challenging.

We consider the linear bandit and the quadratic bandit problems. In both settings, the bandit environment is simulated using a random vector $\theta^\star$ sampled from a normal distribution $\mathrm{N}(0_d, \sigma^\star \mathrm{I}_d)$. We opted for $\sigma^\star = 1/d$ to ensure that the variance of the scalar product $x^\top \theta^\star$ remains independent of the dimension d. The parameter dimension d is set to 20 and we consider a number of arms $K = 50$. In the linear bandit setting, the reward associated with the contextual vector x, is $r = s^\top \theta^\star + \alpha\epsilon$ where $\epsilon \sim \mathrm{N}(0_d, \mathrm{I}_d)$. However, to maintain problem complexity independent of ζ, we have set the signal-to-noise ratio to a fixed value of 1, meaning $\mathbb{E}[(s^\top \theta^\star)^2]/\mathbb{E}[(\alpha\epsilon)^2] = 1$. This implies that $\sqrt{1+\zeta^2} = \alpha$. See Appendix 46 for more details about the setting. In these experiments, we have chosen to compare `VITS − II`, `VITS − II Hessian-free`, Linear TS (LinTS), and LMC-TS, with 10 and 50 iterations of Langevin diffusion at each step. For `VITS` based algorithm, we have only used 10 updating steps. We have omitted the performance of `VITS − I` since it experimentally performs identically to `VITS − II`. For the algorithm `VITS − II Hessian-free`, we approximate the integral presented in (8.13) using 20 Monte Carlo samples. This choice is made due to the observed instability caused by the Monte Carlo error when considering high values of η. However, in our setting, even with 20 Monte Carlo samples, `VITS − II Hessian-free` remains a faster method compared to `VITS − II`. We also attempted to assess the performance of VTS, but, in the ill-conditioned setting, it exhibited a linear and notably high cumulative regret. Consequently, we have opted to exclude it from the figure for the sake of clarity and visibility. The mean and standard error are reported for all experiments over 50 runs. The hyperparameter is provided in Appendix 43.

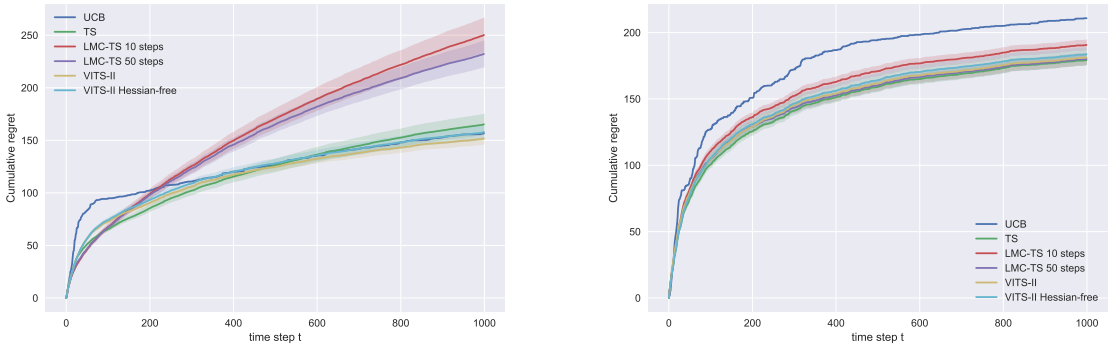


Figure 8.1: Linear bandits, $\zeta = 0.1$ (left), $\zeta = 1$ (right).

Figure 8.1 illustrates the cumulative regret with respect to the time step t for a well-conditioned problem ($\zeta = 1$) and a ill-conditioned problem ($\zeta = 0.1$). Firstly, for $\zeta = 1$, it appears that all methods exhibit similar performance, with the exception of LMC-TS with 10 steps, which slightly underperforms. However, for $\zeta = 0.1$, the optimization problem becomes harder and LMC-TS underperforms even with 50 Langevin steps. This behaviour was expected in our setting, because LMC requires a lot of iterations to converge to the posterior compared to VI. A more complete explanation of this phenomenon can be found in Appendix 44. Finally, we can conclude that `VITS − II` performs similarly to LinTS and that its `Hessian-free` version slightly underperforms but is computationally more efficient.

For Quadratic bandit in Fig 8.2, the reward is $r = (s^\top \theta^\star)^2 + \alpha\epsilon$. This setting is similar to the Linear setting, but we ensure the condition $\mathbb{E}[(s^\top \theta^\star)^4]/\mathbb{E}[(\alpha\epsilon)^2] = 1$ to still get the signal-to-noise ratio equals to 1. This implies a slight different condition $\alpha = (\zeta^2 + 1)\sqrt{3 + 6/d}$, see Appendix 46. Moreover, a simple MLP with two hidden layers of 20 neurons is used for LMC, `VITS − II`, and its `Hessian-free` version as neural network architecture. Performance in Fig 8.2 are similar to linear bandits where `VITS − II` slightly performs better than its `Hessian-free` version but outperforms both LMC and LinTS algorithms as LinTS is not adapted for this setting. The gap between LMC and our algorithm is smaller in the well-conditioned setting than in the ill-conditioned, which was also expected. Finally, additional experience on non-contextual bandits can also be found in Appendix 45.

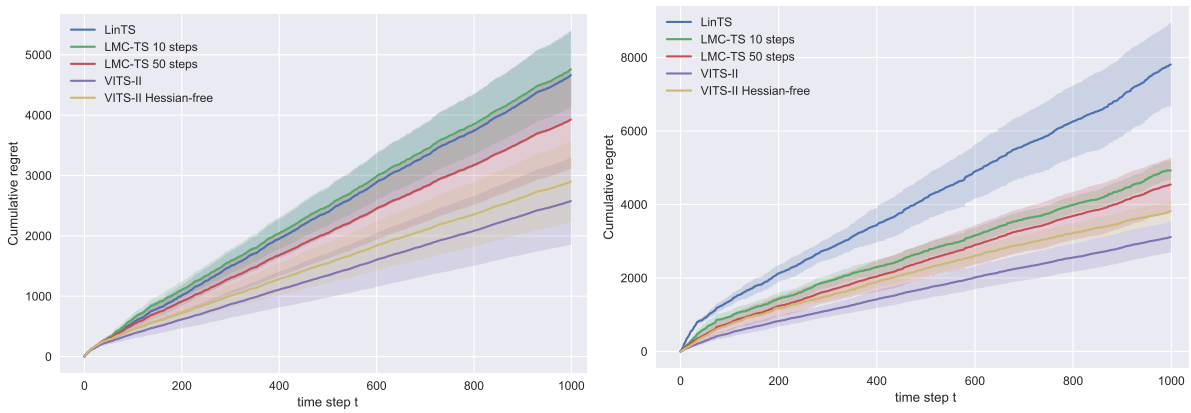


Figure 8.2: Quadratic bandit, $\zeta = 0.1$(left), $\zeta = 1$(right).

8.5 MovieLens Dataset

In this section, we evaluate `VITS` on the MovieLens dataset, consisting of one million ratings by 6040 users for 3952 movies. We adopt the setup proposed in Aouali et al. (2022), involving a low-rank factorization of the rating matrix to yield 5-dimensional representations for users ($s_j \in \mathbb{R}^5$) and movies ($\theta_i \in \mathbb{R}^5$). Movies are treated as potential actions, and context x_t is uniformly sampled from the pool of user vectors. We consider logistic rewards, sampled from $\text{Ber}(\mu(s_j^\top \theta_i))$, where μ is the sigmoid function. We conduct 50 simulations, each involving 100 randomly selected movies.Our prior distribution employs a Gaussian distribution with mean μ_0 and covariance $\Sigma_0 = \text{diag}(\sigma_0)$. Here, μ_0 and σ_0 represent the mean and variance of movie vectors across all dimensions. This setting deviates somewhat from our theoretical framework, where we consider a unified posterior distribution for all arms using a feature map function ϕ representing context-action pairs. In the MovieLens context, each arm possesses an individual posterior distribution. These two settings closely align when the feature map is the vector concatenation function. In practice, we can apply `VITS` or LMC at each arm to obtain posterior samples. In this experiment, we compare LinTS against LMC-TS, `VITS − II`, and the `VITS − II Hessian-free` variant. LMC-TS uses 10 Langevin updating steps. It's crucial to note that for each time step t and each arm a, LMC-TS requires running Langevin diffusion to obtain a new parameter with low correlation to the previous one. This leads to a high computational complexity for LMC-TS. In contrast, `VITS` for each arm only involves sampling from a low-dimensional Gaussian distribution and updating the variational posterior corresponding to the chosen arm. This approach offers significant computational efficiency.

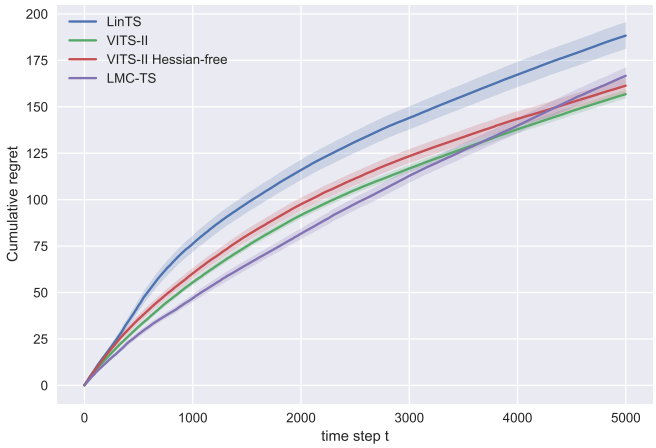


Figure 8.3: Cumulative regret for MovieLens dataset.

Figure 8.3 reveals that LinTS is ill-suited for this particular task, as it assumes rewards to be linear while the approximated algorithms outperform LinTS, as they specifically target the logistic posterior. Remarkably, `VITS` appears to slightly outperform LMC-TS, despite its computational efficiency advantages.

8.6 Conclusion and perspectives

This paper presents two novel TS algorithms called `VITS − I`, `VITS − II` that use VI as an approximation method. Moreover, `VITS − I` algorithms provide robust theoretical guarantees, in particular a cumulative regret bound of $\tilde{\mathcal{O}}(d\sqrt{dT})$ in the linear setting.

One limitation of our analysis is that the regret bound derived is limited to the linear

setting while the interest of our algorithm relies on nonlinear tasks. Additionally, we introduce a third algorithm named VITS − II Hessian-free, which offers enhanced computational efficiency. This algorithm removes the computations of Hessian, resulting in faster execution. Finally, all algorithms have been extensively evaluated in both simulated and real problems.

Part IV

Conclusion, Bibliography and Appendix

Conclusion & Perspectives

In this conclusion we first summarize or contributions and then raise some open questions related to our work.

8.6.1 Conclusion on our Contribution

In this thesis, we have built brick by brick all the ingredients to solve the Robust RL problem in real world settings. Our first question was how to design more sample efficient algorithm and use robust RL algorithm? Let us see what elements of answer we brought to answer this question.

In Chapter 2, we study the sample complexity of obtaining an ϵ-optimal policy in *Robust* discounted Markov Decision Processes (RMDPs), given only access to a generative model of the nominal kernel. We consider uncertainty sets defined with an L_p-ball (recovering the TV case), and study the sample complexity of *any* planning algorithm (with high accuracy guarantee on the solution) applied to an empirical RMDP estimated using the generative model. In the general case, we prove a sample complexity of $\tilde{\mathcal{O}}(\frac{H^4|S||A|}{\epsilon^2})$ for both the *sa*- and *s*-rectangular cases (improvements of $|S|$ and $|S||A|$ respectively). When the size of the uncertainty is small enough, we improve the sample complexity to $\tilde{\mathcal{O}}(\frac{H^3|S||A|}{\epsilon^2})$, recovering the lower-bound for the non-robust case for the first time and a robust lower-bound.

In Chapter 3 , we refine the result of Chapter 2, assuming access to a generative model that samples from the nominal MDP, we examine the sample complexity of RMDPs using a class of generalized L_p norms as the 'distance' function for the uncertainty set, under two commonly adopted *sa*-rectangular and *s*-rectangular conditions. Our results imply that RMDPs can be more sample-efficient to solve than standard MDPs using generalized L_p norms in both *sa*- and *s*-rectangular cases, potentially inspiring more empirical research. We provide a near-optimal upper bound and a matching minimax lower bound for the *sa*-rectangular scenarios. For *s*-rectangular cases, we improve the state-of-the-art upper bound and also derive a lower bound using L_∞ norm that verifies the tightness. Compared to Chapter 2, we improve the sample complexity, showing that it is possible to obtain sample complexity that are lower than in classical MDPs. This part gives a promising avenue to derive algorithm that can achieve lower sample complexity while be more robust on perturbations.

Then we study Deep Robust RL in In Chapter 4 where we try to approximate the Robust Reinforcement Learning constrained with a χ^2-divergence using an approximate Risk-Averse formulation. We show that the classical Reinforcement Learning formulation can be robustified using Standard deviation penalization of the objective. Two algorithms based on Distributional Reinforcement Learning, one for discrete and one for continuous action space are proposed and tested on classical Gym environment to demonstrate the robustness of the algorithms.

In Chapter 5, a new form of implicit robustness in RL using expectile boostraping. Using these technique avoid to estimate a penalisation like in 4. Many classic Reinforcement Learning (RL) algorithms rely on a Bellman operator, which involves an expectation over the next states, leading to the concept of bootstrapping. To introduce a form of pessimism, we propose to replace

this expectation with an expectile. In practice, this can be very simply done by replacing the L_2 loss with a more general expectile loss for the critic. Introducing pessimism in RL is desirable for various reasons, such as tackling the overestimation problem (for which classic solutions are double Q-learning or the twin-critic approach of TD3) or robust RL (where transitions are adversarial). We study empirically these two cases. For the overestimation problem, we show that the proposed approach, `ExpectRL`, provides better results than a classic twin-critic. On robust RL benchmarks, involving changes of the environment, we show that our approach is more robust than classic RL algorithms. We also introduce a variation of `ExpectRL` combined with domain randomization which is competitive with state-of-the-art robust RL agents. Eventually, we also extend `ExpectRL` with a mechanism for choosing automatically the expectile value, that is the degree of pessimism.

Subsequently in the Chapter 6, we try to derive new algorithm without rectangularity assumptions. The rectangularity assumptions in RL Traditional robust reinforcement learning often depends on rectangularity assumptions, where adverse probability measures of outcome states are assumed to be independent across different states and actions. This assumption, rarely fulfilled in practice, leads to overly conservative policies. To address this problem, we introduce a new time-constrained robust MDP (TC-RMDP) formulation that considers multifactorial, correlated, and time-dependent disturbances, thus more accurately reflecting real-world dynamics. This formulation goes beyond the conventional rectangularity paradigm, offering new perspectives and expanding the analytical framework for robust RL. We propose three distinct algorithms, each using varying levels of environmental information, and evaluate them extensively on continuous control benchmarks. Our results demonstrate that these algorithms yield an efficient tradeoff between performance and robustness, outperforming traditional deep robust RL methods in time-constrained environments while preserving robustness in classical benchmarks.

In the Chapter 7, we introduce the Robust Reinforcement Learning Suite (RRLS), a benchmark suite based on Mujoco environments. RRLS provides six continuous control tasks with two types of uncertainty sets for training and evaluation. Our benchmark aims to standardize robust reinforcement learning tasks, facilitating reproducible and comparable experiments, in particular those from recent state-of-the-art contributions, for which we demonstrate the use of RRLS. It is also designed to be easily expandable to new environments. The source code is available at https://github.com/SuReLI/RRLS.

Finally, in the Chapter 8, we tackle the problem of representation of the posterior in the bandit problem using Thompson sampling algorithms with arbitrary posterior distribution learned using Variational inference. We introduce and analyze a variant of the Thompson sampling (TS) algorithm for contextual bandits. At each round, traditional TS requires samples from the current posterior distribution, which is usually intractable. To circumvent this issue, approximate inference techniques can be used and provide samples with distribution close to the posteriors. However, current approximate techniques yield to either poor estimation (Laplace approximation) or can be computationally expensive (MCMC methods, Ensemble sampling...). In this paper, we propose a new algorithm, Varational Inference TS (`VITS`), based on Gaussian Variational Inference. This scheme provides powerful posterior approximations which are easy to sample from, and is computationally efficient, making it an ideal choice for TS. In addition, we show that `VITS` achieves a sub-linear regret bound of the same order in the dimension and number of round as traditional TS for linear contextual bandit. Finally, we demonstrate experimentally the effectiveness of `VITS` on both synthetic and real world datasets.

8.6.2 Future Work and Perspectives

Finally, I would like to end this dissertation with a more personal view on what remains to be done and how our work can be applied to real-world scenarios. The different contributions of this thesis remains mostly theoretical but we could use these tools for practical applications and Simulation to Real. Therefore, I believe there are still many issues and open question that nay need to be addressed before use our method, algorithm and results to other applications.

Extension of the theoretical result to other robust definition settings. A first step would be to adapt our approach to encompass alternative definitions and settings, with a view to enhancing our understanding. It would be beneficial to examine the divergence between probabilities in the definition of RMDPs, such as the KL or χ^2. Furthermore, while the model-based approach with a generative model is a promising avenue for investigation within the Simulation to Real framework, it would be beneficial to consider the question of online settings, as recently explored in Lu et al. (2024), and the potential of model-free settings. Additionally, it would be valuable to investigate how robustness for different divergences could potentially reduce the range of the value function. This question could also be relevant for understanding the role of different pessimistic penalisation in certain offline RL algorithms.

Identify novel Deep Robust algorithms combine concepts from Deep RL and theory. A principal objective of this thesis was to derive a novel Deep Robust RL algorithm in practice, based on the existing theoretical framework. Further investigation may be required to ascertain the potential benefits of combining computer science concepts such as DR with risk-averse formulations such as the expectile in Chapter 5. It would maybe be beneficial to investigate whether ideas from the foundations model and meta-reinforcement learning can be employed to identify a policy that generalises well to downstream tasks with robust Markov decision processes (MDPs). This could potentially lead to the design of a more sample-efficient algorithm. Furthermore, the question of how to derive implicit robustness with straightforward penalisation/estimation remains an avenue for further exploration.

Generalisation versus Performance in RL. Further investigation is required to gain a deeper understanding of the trade-off between generalisation and performance in (Robust) RL. This will enable the development of policies that generalise more effectively while maintaining good performance on the nominal kernel with low sample complexity. It may be the case that a different form of robustness is more suitable in practice than that which is based on theory. Finally, the question of how to circumvent rectangularity assumptions, as discussed in Chapter 6, is also pivotal in practice to achieve algorithms and performances that are not excessively conservative.

Evaluation, metric and benchmark to understand Robustness in RL As is the case in numerous domains within machine learning, the question of how to evaluate and identify pertinent tasks represents a fundamental challenge in the field of robust reinforcement learning (RL). Based on the Mujoco simulator, we propose RRLS, a normalised benchmark presented in Chapter 7. However, this question remains incomplete and would require the inclusion of more realistic and challenging tasks to evaluate the robustness and generalisation of RL algorithms.

Are RMDPs a new way for doing some exploration in RL ? A final proposition for consideration is whether Robust RL facilitates superior exploration, given that it necessitates a

minimal number of samples to reach a solution in theory. It may be the case that, in certain instances, the robust value functions exhibit relatively favourable performance. One potential approach would be to initially target a robust value function, capitalising on the concept of reducing the variability of the value function at the outset. Subsequently, at the conclusion of the training period, a non-robust value function could be targeted in order to enhance performance. This could be achieved by reducing the parameter that controls robustness. (the parameter α in Chapter 4 and 5) during the training represents a potential method for implementing the aforementioned idea.

Bibliography

Abbasi Yadkori, Y., P. L. Bartlett, V. Kanade, Y. Seldin, and C. Szepesvári (2013). Online learning in markov decision processes with adversarially chosen transition probability distributions. Advances in neural information processing systems 26.

Abbasi-Yadkori, Y., D. Pál, and C. Szepesvári (2011). Improved algorithms for linear stochastic bandits. Advances in neural information processing systems 24.

Abbasi-Yadkori, Y., D. Pál, and C. Szepesvári (2011). Improved algorithms for linear stochastic bandits. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger (Eds.), Advances in Neural Information Processing Systems, Volume 24. Curran Associates, Inc.

Abdullah, M. A., H. Ren, H. B. Ammar, V. Milenkovic, R. Luo, et al. (2019). Wasserstein robust reinforcement learning. arXiv preprint arXiv:1907.13196.

Abramowitz, M. and I. A. Stegun (1964). Handbook of mathematical functions with formulas, graphs, and mathematical tables, Volume 55. US Government printing office.

Achiam, J., S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, et al. (2023). Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

Achiam, J. and D. Amodei (2019). Benchmarking safe exploration in deep reinforcement learning.

Agarwal, A., S. Bird, M. Cozowicz, L. Hoang, J. Langford, et al. (2016). Making contextual decisions with low technical debt.

Agarwal, A., S. Kakade, and L. F. Yang (2020). Model-based reinforcement learning with a generative model is minimax optimal. In Conference on Learning Theory, pp. 67–83. PMLR.

Agrawal, S. and N. Goyal (2012). Analysis of thompson sampling for the multi-armed bandit problem. In Conference on learning theory, pp. 39–1. JMLR Workshop and Conference Proceedings.

Agrawal, S. and N. Goyal (2013). Thompson sampling for contextual bandits with linear payoffs. In International conference on machine learning, pp. 127–135. PMLR.

Akkaya, I., M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, et al. (2019). Solving rubik's cube with a robot hand. arXiv preprint arXiv:1910.07113.

Aouali, I., B. Kveton, and S. Katariya (2022). Generalizing hierarchical bayesian bandits. arXiv preprint arXiv:2205.15124.

Artzner, P., F. Delbaen, J.-M. Eber, and D. Heath (1999). Coherent measures of risk. Mathematical finance 9(3), 203–228.

Auer, P. (2002). Using confidence bounds for exploitation-exploration trade-offs. Journal of Machine Learning Research 3(Nov), 397–422.

Auer, P., N. Cesa-Bianchi, and P. Fischer (2002). Finite-time analysis of the multiarmed bandit problem. Machine learning 47(2), 235–256.

Azar, M., R. Munos, and H. J. Kappen (2013a). Minimax pac bounds on the sample complexity of reinforcement learning with a generative model. Machine learning 91(3), 325–349.

Azar, M. G., R. Munos, M. Ghavamzadeh, and H. Kappen (2011). Reinforcement learning with a near optimal rate of convergence.

Azar, M. G., R. Munos, and H. J. Kappen (2013b). Minimax PAC bounds on the sample complexity of reinforcement learning with a generative model. Machine learning 91(3), 325–349.

Badia, A. P., B. Piot, S. Kapturowski, P. Sprechmann, A. Vitvitskyi, et al. (2020). Agent57: Outperforming the atari human benchmark. In International conference on machine learning, pp. 507–517. PMLR.

Badrinath, K. P. and D. Kalathil (2021). Robust reinforcement learning using least squares policy iteration with provable performance guarantees. In International Conference on Machine Learning, pp. 511–520. PMLR.

Bai, C., L. Wang, Z. Yang, Z. Deng, A. Garg, et al. (2022). Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. arXiv preprint arXiv:2202.11566.

Bai, Y., T. Xie, N. Jiang, and Y.-X. Wang (2019). Provably efficient Q-learning with low switching cost. arXiv preprint arXiv:1905.12849.

Beattie, C., J. Z. Leibo, D. Teplyashin, T. Ward, M. Wainwright, et al. (2016). Deepmind lab.

Beck, C. L. and R. Srikant (2012). Error bounds for constant step-size Q-learning. Systems & control letters 61(12), 1203–1208.

Behzadian, B., M. Petrik, and C. P. Ho (2021). Fast algorithms for l_∞-constrained s-rectangular robust mdps. Advances in Neural Information Processing Systems 34.

Bellemare, M., Y. Naddaf, J. Veness, and M. Bowling (2012). The arcade learning environment: An evaluation platform for general agents. Journal of Artificial Intelligence Research 47.

Bellemare, M. G., W. Dabney, and R. Munos (2017). A distributional perspective on reinforcement learning. 34th International Conference on Machine Learning, ICML 2017 1, 693–711.

Bellemare, M. G., Y. Naddaf, J. Veness, and M. Bowling (2013). The arcade learning environment: An evaluation platform for general agents. Journal of Artificial Intelligence Research 47, 253–279.

Bellini, F. and E. Di Bernardino (2017). Risk management with expectiles. The European Journal of Finance 23(6), 487–506.

Bellini, F., B. Klar, A. Müller, and E. R. Gianin (2014). Generalized quantiles as risk measures. Insurance: Mathematics and Economics 54, 41–48.

Bellman, R. (1957). A markovian decision process. Journal of mathematics and mechanics, 679–684.

Bellman, R. (1966). Dynamic programming. science 153(3731), 34–37.

Berridge, K. C. (2007). The debate over dopamine's role in reward: the case for incentive salience. Psychopharmacology 191, 391–431.

Berry, D. A. and B. Fristedt (1985). Bandit problems: sequential allocation of experiments (monographs on statistics and applied probability). London: Chapman and Hall 5(71-87), 7–7.

Bertsekas, D. P. (2017). Dynamic programming and optimal control (4th edition). Athena Scientific.

Bertsimas, D., V. Gupta, and N. Kallus (2018). Data-driven robust optimization. Mathematical Programming 167(2), 235–292.

Blanchet, J., M. Lu, T. Zhang, and H. Zhong (2023). Double pessimism is provably efficient for distributionally robust offline reinforcement learning: Generic algorithm and robust partial coverage. arXiv preprint arXiv:2305.09659.

Blanchet, J. and K. Murthy (2019). Quantifying distributional model risk via optimal transport. Mathematics of Operations Research 44(2), 565–600.

Blei, D. M., A. Kucukelbir, and J. D. McAuliffe (2017). Variational inference: A review for statisticians. Journal of the American statistical Association 112(518), 859–877.

Bonnabel, S. (2013). Stochastic gradient descent on riemannian manifolds. IEEE Transactions on Automatic Control 58(9), 2217–2229.

Bouneffouf, D., I. Rish, and C. Aggarwal (2020). Survey on applications of multi-armed and contextual bandits. In 2020 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE.

Brekelmans, R., T. Genewein, J. Grau-Moya, G. Delétang, M. Kunesch, et al. (2022). Your policy regularizer is secretly an adversary. arXiv preprint arXiv:2203.12592.

Brockman, G., V. Cheung, L. Pettersson, J. Schneider, J. Schulman, et al. (2016). Openai gym.

Cesa-Bianchi, N. and G. Lugosi (2006). Prediction, learning, and games. Cambridge university press.

Chapelle, O. and L. Li (2011). An empirical evaluation of thompson sampling. Advances in neural information processing systems 24.

Chen, Z., S. T. Maguluri, S. Shakkottai, and K. Shanmugam (2020). Finite-sample analysis of stochastic approximation using smooth convex envelopes. arXiv preprint arXiv:2002.00874.

Cheung, W. C., D. Simchi-Levi, and R. Zhu (2019). Reinforcement learning under drift. arXiv preprint arXiv:1906.02922.

Chow, Y., A. Tamar, S. Mannor, and M. Pavone (2015). Risk-sensitive and robust decision-making: a cvar optimization approach. Advances in neural information processing systems 28.

Chu, W., L. Li, L. Reyzin, and R. Schapire (2011). Contextual bandits with linear payoff functions. In G. Gordon, D. Dunson, and M. Dudík (Eds.), Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Volume 15 of Proceedings of Machine Learning Research, Fort Lauderdale, FL, USA, pp. 208–214. PMLR.

Clavier, P., S. Allassonière, and E. L. Pennec (2022). Robust reinforcement learning with distributional risk-averse formulation. arXiv preprint arXiv:2206.06841.

Clavier, P., T. Huix, and A. Durmus (2023). Vits: Variational inference thomson sampling for contextual bandits. arXiv preprint arXiv:2307.10167.

Clavier, P., E. L. Pennec, and M. Geist (2023). Towards minimax optimality of model-based robust reinforcement learning. arXiv preprint arXiv:2302.05372.

Clavier, P., E. Rachelson, E. L. Pennec, and M. Geist (2024). Bootstrapping expectiles in reinforcement learning. arXiv preprint arXiv:2406.04081.

Cobbe, K., C. Hesse, J. Hilton, and J. Schulman (2019). Leveraging procedural generation to benchmark reinforcement learning. arXiv preprint arXiv:1912.01588.

Cobbe, K., C. Hesse, J. Hilton, and J. Schulman (2020). Leveraging procedural generation to benchmark reinforcement learning. In International conference on machine learning, pp. 2048–2056. PMLR.

Dabney, W., G. Ostrovski, D. Silver, and R. Munos (2018a). Implicit quantile networks for distributional reinforcement learning. 35th International Conference on Machine Learning, ICML 2018 3, 1774–1787.

Dabney, W., G. Ostrovski, D. Silver, and R. Munos (2018b). Implicit quantile networks for distributional reinforcement learning. In International conference on machine learning, pp. 1096–1105. PMLR.

Dabney, W., M. Rowland, M. G. Bellemare, and R. Munos (2017). Distributional reinforcement learning with quantile regression. 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, 2892–2901.

Delbaen, F. (2000). Draft: Coherent risk measures. Lecture notes, Pisa.

Delbaen, F. (2002). Coherent risk measures on general probability spaces. Advances in finance and stochastics: essays in honour of Dieter Sondermann, 1–37.

Delbaen, F. (2013). A remark on the structure of expectiles. arXiv preprint arXiv:1307.5881.

Dennis, M., N. Jaques, E. Vinitsky, A. Bayen, S. J. Russell, et al. (2020). Emergent complexity and zero-shot transfer via unsupervised environment design. Neural Information Processing Systems.

Derman, E., M. Geist, and S. Mannor (2021). Twice regularized mdps and the equivalence between robustness and regularization. Advances in Neural Information Processing Systems 34, 22274–22287.

Derman, E. and S. Mannor (2020). Distributional robustness and regularization in reinforcement learning. arXiv preprint arXiv:2003.02894.

Derman, E., Y. Men, M. Geist, and S. Mannor. Robustness and regularization in reinforcement learning. In NeurIPS 2023 Workshop on Generalization in Planning.

Dong, J., J. Li, B. Wang, and J. Zhang (2022). Online policy optimization for robust MDP. arXiv preprint arXiv:2209.13841.

Dong, K., Y. Wang, X. Chen, and L. Wang (2019). Q-learning with UCB exploration is sample efficient for infinite-horizon MDP. arXiv preprint arXiv:1901.09311.

Duchi, J., P. Glynn, and H. Namkoong (2016). Statistics of robust optimization: A generalized empirical likelihood approach. arXiv preprint arXiv:1610.03425.

Duchi, J. and H. Namkoong (2018). Learning models with uniform performance via distributionally robust optimization. arXiv preprint arXiv:1810.08750.

Duchi, J. C. and H. Namkoong (2021). Learning models with uniform performance via distributionally robust optimization. The Annals of Statistics 49(3), 1378–1406.

Even-Dar, E., S. M. Kakade, and Y. Mansour (2004). Experts in a markov decision process. Advances in neural information processing systems 17.

Eysenbach, B. and S. Levine (2021). Maximum entropy rl (provably) solves some robust rl problems. arXiv preprint arXiv:2103.06257.

Fatemi, M., T. W. Killian, J. Subramanian, and M. Ghassemi (2021). Medical dead-ends and learning to identify high-risk states and treatments. Advances in Neural Information Processing Systems 34, 4856–4870.

Faury, L., M. Abeille, K.-S. Jun, and C. Calauzènes (2022). Jointly efficient and optimal algorithms for logistic bandits. In International Conference on Artificial Intelligence and Statistics, pp. 546–580. PMLR.

Filippi, S., O. Cappe, A. Garivier, and C. Szepesvári (2010). Parametric bandits: The generalized linear case. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta (Eds.), Advances in Neural Information Processing Systems, Volume 23. Curran Associates, Inc.

Foster, D. and A. Rakhlin (2020). Beyond ucb: Optimal and efficient contextual bandits with regression oracles. In International Conference on Machine Learning, pp. 3199–3210. PMLR.

Fu, J., A. Kumar, O. Nachum, G. Tucker, and S. Levine (2021). D4rl: Datasets for deep data-driven reinforcement learning.

Fujimoto, S., H. Hoof, and D. Meger (2018). Addressing function approximation error in actor-critic methods. In International conference on machine learning, pp. 1587–1596. PMLR.

Gajane, P., R. Ortner, and P. Auer (2018). A sliding-window algorithm for Markov decision processes with arbitrarily changing rewards and transitions. arXiv preprint arXiv:1805.10066.

Gao, R. (2020). Finite-sample guarantees for wasserstein distributionally robust optimization: Breaking the curse of dimensionality. arXiv preprint arXiv:2009.04382.

Garivier, A. and E. Moulines (2008). On upper-confidence bound policies for non-stationary bandit problems. arXiv preprint arXiv:0805.3415.

Gautron, R., D. Baudry, M. Adam, G. N. Falconnier, G. Hoogenboom, et al. (2024). A new adaptive identification strategy of best crop management with farmers. Field Crops Research 307, 109249.

Geist, M., B. Scherrer, and O. Pietquin (2019). A theory of regularized markov decision processes. In International Conference on Machine Learning, pp. 2160–2169. PMLR.

Gneiting, T. (2011). Making and evaluating point forecasts. Journal of the American Statistical Association 106(494), 746–762.

Gotoh, J.-y., M. J. Kim, and A. E. Lim (2018). Robust empirical optimization is almost the same as mean–variance optimization. Operations research letters 46(4), 448–452.

Goyal, V. and J. Grand-Clement (2018). Robust markov decision process: Beyond rectangularity. arXiv preprint arXiv:1811.00215.

Goyal, V. and J. Grand-Clement (2022). Robust markov decision processes: Beyond rectangularity. Mathematics of Operations Research.

Grand-Clément, J. and C. Kroer (2020a). First-order methods for wasserstein distributionally robust mdp. arXiv preprint arXiv:2009.06790.

Grand-Clément, J. and C. Kroer (2020b). Scalable first-order methods for robust mdps. arXiv preprint arXiv:2005.05434.

Grand-Clément, J. and C. Kroer (2021). Scalable first-order methods for robust mdps. In Proceedings of the AAAI Conference on Artificial Intelligence, Volume 35, pp. 12086–12094.

Greenberg, I., Y. Chow, M. Ghavamzadeh, and S. Mannor (2022). Efficient risk-averse reinforcement learning. Advances in Neural Information Processing Systems 35, 32639–32652.

Gu, S., L. Shi, M. Wen, M. Jin, E. Mazumdar, et al. (2024). Robust gymnasium: A unified modular benchmark for robust reinforcement learning. Github.

Gulcehre, C., Z. Wang, A. Novikov, T. L. Paine, S. G. Colmenarejo, et al. (2020). Rl unplugged: Benchmarks for offline reinforcement learning.

Haarnoja, T., A. Zhou, P. Abbeel, and S. Levine (2018a). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In International conference on machine learning, pp. 1861–1870. PMLR.

Haarnoja, T., A. Zhou, P. Abbeel, and S. Levine (2018b). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. 35th International Conference on Machine Learning, ICML 2018 5, 2976–2989.

Hamidi, N. and M. Bayati (2020). On worst-case regret of linear thompson sampling. arXiv preprint arXiv:2006.06790.

Han, S., S. Su, S. He, S. Han, H. Yang, et al. (2022). What is the solution for state adversarial multi-agent reinforcement learning? arXiv preprint arXiv:2212.02705.

Hasselt, H. (2010). Double q-learning. Advances in neural information processing systems 23.

Ho, C. P., M. Petrik, and W. Wiesemann (2018). Fast bellman updates for robust mdps. In International Conference on Machine Learning, pp. 1979–1988. PMLR.

Ho, C. P., M. Petrik, and W. Wiesemann (2021). Partial policy iteration for l1-robust markov decision processes. J. Mach. Learn. Res. 22, 275–1.

Hoeffding, W. (1994). Probability inequalities for sums of bounded random variables. In The collected works of Wassily Hoeffding, pp. 409–426. Springer.

Huang, S., R. F. J. Dossa, C. Ye, J. Braga, D. Chakraborty, et al. (2022). Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. Journal of Machine Learning Research 23(274), 1–18.

Huix, T., M. Zhang, and A. Durmus (2023). Tight regret and complexity bounds for thompson sampling via langevin monte carlo. In F. Ruiz, J. Dy, and J.-W. van de Meent (Eds.), Proceedings of The 26th International Conference on Artificial Intelligence and Statistics, Volume 206 of Proceedings of Machine Learning Research, pp. 8749–8770. PMLR.

Husain, H., K. Ciosek, and R. Tomioka (2021). Regularized policies are reward robust. In International Conference on Artificial Intelligence and Statistics, pp. 64–72. PMLR.

Iyengar, G. (2022). Robust dynamic programming. Technical report, CORC Tech Report TR-2002-07.

Iyengar, G. N. (2005). Robust dynamic programming. Mathematics of Operations Research 30(2), 257–280.

Jafarnia-Jahromi, M., C.-Y. Wei, R. Jain, and H. Luo (2020). A model-free learning algorithm for infinite-horizon average-reward MDPs with near-optimal regret. arXiv preprint arXiv:2006.04354.

Jain, A., G. Patil, A. Jain, K. Khetarpal, and D. Precup (2021). Variance penalized on-policy and off-policy actor-critic. arXiv preprint arXiv:2102.01985.

James, S., Z. Ma, D. R. Arrojo, and A. J. Davison (2020). Rlbench: The robot learning benchmark and learning environment. IEEE Robotics and Automation Letters 5(2), 3019–3026.

Jin, C., Z. Allen-Zhu, S. Bubeck, and M. I. Jordan (2018). Is Q-learning provably efficient? In Advances in Neural Information Processing Systems, pp. 4863–4873.

Jin, C., A. Krishnamurthy, M. Simchowitz, and T. Yu (2020). Reward-free exploration for reinforcement learning. In International Conference on Machine Learning, pp. 4870–4879. PMLR.

Jin, T., P. Xu, J. Shi, X. Xiao, and Q. Gu (2021). Mots: Minimax optimal thompson sampling. In International Conference on Machine Learning, pp. 5074–5083. PMLR.

Jin, Y., Z. Yang, and Z. Wang (2021). Is pessimism provably efficient for offline RL? In International Conference on Machine Learning, pp. 5084–5096.

Jong, N. K. and P. Stone (2005). Bayesian models of nonstationary Markov decision processes. Planning and Learning in A Priori Unknown or Dynamic Domains, 132.

Jullien, S., R. Deffayet, J.-M. Renders, P. Groth, and M. de Rijke (2023). Distributional reinforcement learning with dual expectile-quantile regression. arXiv preprint arXiv:2305.16877.

Kakade, S. (2003). On the sample complexity of reinforcement learning. Ph. D. thesis, University of London.

Kamalaruban, P., Y. ting Huang, Y.-P. Hsieh, P. Rolland, C. Shi, et al. (2020). Robust reinforcement learning via adversarial training with langevin dynamics. Neural Information Processing Systems.

Karush, W. (2013). Minima of functions of several variables with inequalities as side conditions. In Traces and emergence of nonlinear programming, pp. 217–245. Springer.

Katehakis, M. N. and A. F. Veinott (1987). The multi-armed bandit problem: Decomposition and computation. Math. Oper. Res. 12, 262–268.

Katsevich, A. and P. Rigollet (2023). On the approximation accuracy of gaussian variational inference. arXiv preprint arXiv:2301.02168.

Kaufman, D. L. and A. J. Schaefer (2013). Robust modified policy iteration. INFORMS Journal on Computing 25(3), 396–410.

Kaufmann, E., N. Korda, and R. Munos (2012). Thompson sampling: An asymptotically optimal finite-time analysis. In International conference on algorithmic learning theory, pp. 199–213. Springer.

Kearns, M. J. and S. P. Singh (1999). Finite-sample convergence rates for Q-learning and indirect algorithms. In Advances in neural information processing systems, pp. 996–1002.

Klopp, O., K. Lounici, and A. B. Tsybakov (2017). Robust matrix completion. Probability Theory and Related Fields 169(1-2), 523–564.

Kober, J., J. A. Bagnell, and J. Peters (2013). Reinforcement learning in robotics: A survey. The International Journal of Robotics Research 32(11), 1238–1274.

Kostrikov, I., A. Nair, and S. Levine (2021). Offline reinforcement learning with implicit q-learning. arXiv preprint arXiv:2110.06169.

Kozuno, T., W. Yang, N. Vieillard, T. Kitamura, Y. Tang, et al. (2022). Kl-entropy-regularized rl with a generative model is minimax optimal. arXiv preprint arXiv:2205.14211.

Kumar, A., A. Levine, T. Goldstein, and S. Feizi (2022). Certifying model accuracy under distribution shifts. arXiv preprint arXiv:2201.12440.

Kumar, N., E. Derman, M. Geist, K. Levy, and S. Mannor (2023). Policy gradient for s-rectangular robust markov decision processes. arXiv preprint arXiv:2301.13589.

Kumar, N., K. Levy, K. Wang, and S. Mannor (2022). Efficient policy iteration for robust markov decision processes via regularization. arXiv preprint arXiv:2205.14327.

Kuznetsov, A., P. Shvechikov, A. Grishin, and D. Vetrov (2020). Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In International Conference on Machine Learning, pp. 5556–5566. PMLR.

Kveton, B., M. Zaheer, C. Szepesvari, L. Li, M. Ghavamzadeh, et al. (2020). Randomized exploration in generalized linear bandits. In S. Chiappa and R. Calandra (Eds.), Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, Volume 108 of Proceedings of Machine Learning Research, pp. 2066–2076. PMLR.

Lam, S. and J. Herlocker. Movielens dataset.

Lambert, M., S. Bonnabel, and F. Bach (2022). The recursive variational gaussian approximation (r-vga). Statistics and Computing 32(1), 10.

Lambert, M., S. Chewi, F. Bach, S. Bonnabel, and P. Rigollet (2022). Variational inference via wasserstein gradient flows. arXiv preprint arXiv:2205.15902.

Langford, J. and T. Zhang (2007). The epoch-greedy algorithm for contextual multi-armed bandits. Advances in neural information processing systems 20(1), 96–1.

Lattimore, T. and C. Szepesvári (2020). Bandit algorithms. Cambridge University Press.

Lecarpentier, E. and E. Rachelson (2019). Non-stationary markov decision processes, a worst-case approach using model-based reinforcement learning. Advances in neural information processing systems 32.

Lee, K., L. Smith, A. Dragan, and P. Abbeel (2021). B-pref: Benchmarking preference-based reinforcement learning. arXiv preprint arXiv:2111.03026.

Li, G., C. Cai, Y. Chen, Y. Wei, and Y. Chi (2023). Is Q-learning minimax optimal? a tight sample complexity analysis. Operations Research.

Li, G., Y. Chi, Y. Wei, and Y. Chen (2022). Minimax-optimal multi-agent RL in Markov games with a generative model. Neural Information Processing Systems.

Li, G., L. Shi, Y. Chen, Y. Chi, and Y. Wei (2022). Settling the sample complexity of model-based offline reinforcement learning. arXiv preprint arXiv:2204.05275.

Li, G., L. Shi, Y. Chen, Y. Gu, and Y. Chi (2021). Breaking the sample complexity barrier to regret-optimal model-free reinforcement learning. Advances in Neural Information Processing Systems 34.

Li, G., Y. Wei, Y. Chi, and Y. Chen (2024). Breaking the sample size barrier in model-based reinforcement learning with a generative model. Operations Research 72(1), 203–221.

Li, G., Y. Wei, Y. Chi, Y. Gu, and Y. Chen (2020). Breaking the sample size barrier in model-based reinforcement learning with a generative model. Advances in neural information processing systems 33, 12861–12872.

Li, G., Y. Yan, Y. Chen, and J. Fan (2023). Minimax-optimal reward-agnostic exploration in reinforcement learning. arXiv preprint arXiv:2304.07278.

Li, L., W. Chu, J. Langford, and R. E. Schapire (2010). A contextual-bandit approach to personalized news article recommendation. In Proceedings of the 19th international conference on World wide web, pp. 661–670.

Li, L., Y. Lu, and D. Zhou (2017). Provably optimal algorithms for generalized linear contextual bandits. In D. Precup and Y. W. Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning, Volume 70 of Proceedings of Machine Learning Research, pp. 2071–2080. PMLR.

Li, M., T. Sutter, and D. Kuhn (2023). Policy gradient algorithms for robust mdps with non-rectangular uncertainty sets. arXiv preprint arXiv:2305.19004.

Li, S., Y. Wu, X. Cui, H. Dong, F. Fang, et al. (2019a). Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient. In Proceedings of the AAAI conference on artificial intelligence, Volume 33, pp. 4213–4220.

Li, S., Y. Wu, X. Cui, H. Dong, F. Fang, et al. (2019b). Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient. In Proceedings of the AAAI conference on artificial intelligence, Volume 33, pp. 4213–4220.

Li, Y., T. Zhao, and G. Lan (2022). First-order policy optimization for robust markov decision process. arXiv preprint arXiv:2209.10579.

Liang, Y., Y. Sun, R. Zheng, X. Liu, T. Sandholm, et al. (2023). Game-theoretic robust reinforcement learning handles temporally-coupled perturbations. In The Second Workshop on New Frontiers in Adversarial Machine Learning.

Lillicrap, T. P., J. J. Hunt, A. Pritzel, N. Heess, T. Erez, et al. (2015). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.

Lin, Z., G. Thomas, G. Yang, and T. Ma (2020). Model-based adversarial meta-reinforcement learning. In Advances in Neural Information Processing Systems, Volume 33, pp. 10161–10173.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In Machine learning proceedings 1994, pp. 157–163. Elsevier.

Liu, S., K. Y. Ngiam, and M. Feng (2019). Deep reinforcement learning for clinical decision support: a brief survey. arXiv preprint arXiv:1907.09475.

Liu, Z., Q. Bai, J. Blanchet, P. Dong, W. Xu, et al. (2022). Distributionally robust q-learning. In International Conference on Machine Learning, pp. 13623–13643. PMLR.

Lu, M., H. Zhong, T. Zhang, and J. Blanchet (2024). Distributionally robust reinforcement learning with interactive data collection: Fundamental hardness and near-optimal algorithm. arXiv preprint arXiv:2404.03578.

Lu, X. and B. Van Roy (2017). Ensemble sampling. Advances in neural information processing systems 30.

Ma, Y. J., D. Jayaraman, and O. Bastani (2021). Conservative offline distributional reinforcement learning.

Mahmood, A. R., D. Korenkevych, G. Vasan, W. Ma, and J. Bergstra (2018). Benchmarking reinforcement learning algorithms on real-world robots. In Conference on robot learning, pp. 561–591. PMLR.

Mannor, S., D. Simester, P. Sun, and J. N. Tsitsiklis (2004). Bias and variance in value function estimation. In Proceedings of the twenty-first international conference on Machine learning, pp. 72.

Mazumdar, E., A. Pacchiano, Y.-a. Ma, P. L. Bartlett, and M. I. Jordan (2020). On thompson sampling with langevin algorithms. arXiv preprint arXiv:2002.10002.

McDiarmid, C. et al. (1989). On the method of bounded differences. Surveys in combinatorics 141(1), 148–188.

Mehta, B., M. Diaz, F. Golemo, C. J. Pal, and L. Paull (2020a). Active domain randomization. In Proceedings of the Conference on Robot Learning, Volume 100, pp. 1162–1176.

Mehta, B., M. Diaz, F. Golemo, C. J. Pal, and L. Paull (2020b). Active domain randomization. In Conference on Robot Learning, pp. 1162–1176. PMLR.

Ménard, P. and A. Garivier (2017). A minimax and asymptotically optimal algorithm for stochastic bandits. In International Conference on Algorithmic Learning Theory, pp. 223–237. PMLR.

Mnih, V., K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, et al. (2013). Playing atari with deep reinforcement learning.

Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, et al. (2015). Human-level control through deep reinforcement learning. Nature 518(7540), 529–533.

Montague, P. R., P. Dayan, and T. J. Sejnowski (1996). A framework for mesencephalic dopamine systems based on predictive hebbian learning. Journal of neuroscience 16(5), 1936–1947.

Moos, J., K. Hansel, H. Abdulsamad, S. Stark, D. Clever, et al. (2022). Robust reinforcement learning: A review of foundations and recent advances. Machine Learning and Knowledge Extraction 4(1), 276–315.

Morimoto, J. and K. Doya (2005). Robust reinforcement learning. Neural computation 17(2), 335–359.

Moskovitz, T., J. Parker-Holder, A. Pacchiano, M. Arbel, and M. Jordan (2021). Tactical optimism and pessimism for deep reinforcement learning. Advances in Neural Information Processing Systems 34, 12849–12863.

Nam, D. W., Y. Kim, and C. Y. Park (2021). Gmac: A distributional perspective on actor-critic framework. In International Conference on Machine Learning, pp. 7927–7936. PMLR.

Nichol, A., V. Pfau, C. Hesse, O. Klimov, and J. Schulman (2018). Gotta learn fast: A new benchmark for generalization in rl. arXiv preprint arXiv:1804.03720.

Nilim, A. and L. El Ghaoui (2005). Robust control of markov decision processes with uncertain transition matrices. Operations Research 53(5), 780–798.

OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, et al. (2019). Solving rubik's cube with a robot hand. arXiv preprint arXiv: Arxiv-1910.07113.

Ornik, M. and U. Topcu (2019). Learning and planning for time-varying mdps using maximum likelihood estimation. arXiv preprint arXiv:1911.12976.

Osband, I., Y. Doron, M. Hessel, J. Aslanides, E. Sezener, et al. (2019). Behaviour suite for reinforcement learning. arXiv preprint arXiv:1908.03568.

Pacchiano, A., M. Phan, Y. Abbasi Yadkori, A. Rao, J. Zimmert, et al. (2020). Model selection in contextual stochastic bandit problems. Advances in Neural Information Processing Systems 33, 10328–10337.

Packer, C., K. Gao, J. Kos, P. Krähenbühl, V. Koltun, et al. (2018). Assessing generalization in deep reinforcement learning. arXiv preprint arXiv:1810.12282.

Pan, X., D. Seita, Y. Gao, and J. Canny (2019). Risk averse robust adversarial reinforcement learning. In 2019 International Conference on Robotics and Automation (ICRA), pp. 8522–8528. IEEE.

Panaganti, K. and D. Kalathil (2022a). Sample complexity of robust reinforcement learning with a generative model. In International Conference on Artificial Intelligence and Statistics, pp. 9582–9602. PMLR.

Panaganti, K. and D. Kalathil (2022b). Sample complexity of robust reinforcement learning with a generative model. In International Conference on Artificial Intelligence and Statistics, pp. 9582–9602. PMLR.

Panaganti, K., Z. Xu, D. Kalathil, and M. Ghavamzadeh (2022). Robust reinforcement learning using offline data. arXiv preprint arXiv:2208.05129.

Petrik, M. and R. H. Russel (2019). Beyond confidence regions: Tight bayesian ambiguity sets for robust mdps. Advances in neural information processing systems 32.

Pinto, L., J. Davidson, R. Sukthankar, and A. Gupta (2017). Robust adversarial reinforcement learning. In International Conference on Machine Learning, pp. 2817–2826. PMLR.

Puterman, M. L. (1990). Markov decision processes. Handbooks in operations research and management science 2, 331–434.

Puterman, M. L. (2014). Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons.

Qiaoben, Y., X. Zhou, C. Ying, and J. Zhu (2021). Strategically-timed state-observation attacks on deep reinforcement learning agents. In ICML 2021 Workshop on Adversarial Machine Learning.

Raffin, A., A. Hill, M. Ernestus, A. Gleave, A. Kanervisto, et al. (2019). Stable baselines3.

Raffin, A., A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, et al. (2021). Stable-baselines3: Reliable reinforcement learning implementations. Journal of Machine Learning Research 22(268), 1–8.

Rahimian, H. and S. Mehrotra (2019). Distributionally robust optimization: A review. arXiv preprint arXiv:1908.05659.

Rashidinejad, P., B. Zhu, C. Ma, J. Jiao, and S. Russell (2021). Bridging offline reinforcement learning and imitation learning: A tale of pessimism. Neural Information Processing Systems (NeurIPS).

Réda, C. (2022). Combination of gene regulatory networks and sequential machine learning for drug repurposing. Ph. D. thesis, Université Paris Cité.

Riedmiller, M. (2005). Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method. In Machine Learning: ECML 2005: 16th European Conference on Machine Learning, Porto, Portugal, October 3-7, 2005. Proceedings 16, pp. 317–328. Springer.

Riquelme, C., G. Tucker, and J. Snoek (2018). Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. In International Conference on Learning Representations.

Robbins, H. (1952). Some aspects of the sequential design of experiments. Bulletin of the American Mathematical Society 58(5), 527–535.

Rowland, M., R. Dadashi, S. Kumar, R. Munos, M. G. Bellemare, et al. (2019). Statistics and samples in distributional reinforcement learning. In International Conference on Machine Learning, pp. 5528–5536. PMLR.

Roy, A., H. Xu, and S. Pokutta (2017). Reinforcement learning under model mismatch. Advances in neural information processing systems 30.

Rudin, W. et al. (1964). Principles of mathematical analysis, Volume 3. McGraw-hill New York.

Russel, R. H., B. Behzadian, and M. Petrik (2019). Optimizing norm-bounded weighted ambiguity sets for robust mdps. arXiv preprint arXiv:1912.02696.

Russo, D. and B. Van Roy (2014). Learning to optimize via posterior sampling. Mathematics of Operations Research 39(4), 1221–1243.

Russo, D. and B. Van Roy (2016). An information-theoretic analysis of thompson sampling. The Journal of Machine Learning Research 17(1), 2442–2471.

Schaul, T., J. Quan, I. Antonoglou, and D. Silver (2015). Prioritized experience replay. arXiv preprint arXiv:1511.05952.

Scherrer, B. (2013). Performance bounds for λ policy iteration and application to the game of tetris. Journal of Machine Learning Research 14(4).

Scherrer, B., M. Ghavamzadeh, V. Gabillon, B. Lesner, and M. Geist (2015). Approximate modified policy iteration and its application to the game of tetris. J. Mach. Learn. Res. 16(49), 1629–1676.

Scherrer, B. and B. Lesner (2012). On the use of non-stationary policies for stationary infinite-horizon markov decision processes. Advances in Neural Information Processing Systems 25.

Schulman, J., S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel (2015). Trust region policy optimization. 32nd International Conference on Machine Learning, ICML 2015 3, 1889–1897.

Schulman, J., F. Wolski, P. Dhariwal, A. Radford, and O. Klimov (2017a). Proximal policy optimization algorithms. arXiv. PPO algorithm premier papier¡br/¿Important à citer¡br/¿¡br/¿.

Schulman, J., F. Wolski, P. Dhariwal, A. Radford, and O. Klimov (2017b). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

Shi, L. and Y. Chi (2022). Distributionally robust model-based offline reinforcement learning with near-optimal sample complexity. arXiv preprint arXiv:2208.05767.

Shi, L., G. Li, Y. Wei, Y. Chen, and Y. Chi (2022). Pessimistic Q-learning for offline reinforcement learning: Towards optimal sample complexity. In Proceedings of the 39th International Conference on Machine Learning, Volume 162, pp. 19967–20025. PMLR.

Shi, L., G. Li, Y. Wei, Y. Chen, M. Geist, et al. (2023). The curious price of distributional robustness in reinforcement learning with a generative model. arXiv preprint arXiv:2305.16589.

Shi, L., G. Li, Y. Wei, Y. Chen, M. Geist, et al. (2024). The curious price of distributional robustness in reinforcement learning with a generative model. Advances in Neural Information Processing Systems 36.

Sidford, A., M. Wang, X. Wu, L. Yang, and Y. Ye (2018). Near-optimal time and sample complexities for solving markov decision processes with a generative model. Advances in Neural Information Processing Systems 31.

Silver, D., J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, et al. (2017). Mastering the game of Go without human knowledge. Nature 550(7676), 354–359.

Singh, R., Q. Zhang, and Y. Chen (2020). Improving robustness via risk averse distributional reinforcement learning. In Learning for Dynamics and Control, pp. 958–968. PMLR.

Singh, S. P. and R. C. Yee (1994). An upper bound on the loss from approximate optimal-value functions. Machine Learning 16(3), 227–233.

Smirnova, E., E. Dohmatob, and J. Mary (2019a). Distributionally robust reinforcement learning. arXiv preprint arXiv:1902.08708.

Smirnova, E., E. Dohmatob, and J. Mary (2019b). Distributionally robust reinforcement learning.

Stanton, S., R. Fakoor, J. Mueller, A. G. Wilson, and A. Smola (2021). Robust reinforcement learning for shifting dynamics during deployment. In Workshop on Safe and Robust Control of Uncertain Systems at NeurIPS.

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. Machine learning 3(1), 9–44.

Sutton, R. S. and A. G. Barto (2018). Reinforcement learning: An introduction. MIT press.

Tamar, A., S. Mannor, and H. Xu (2014). Scaling up robust MDPs using function approximation. In International conference on machine learning, pp. 181–189. PMLR.

Tan, K. L., Y. Esfandiari, X. Y. Lee, and S. Sarkar (2020). Robustifying reinforcement learning agents via action space adversarial training. In 2020 American control conference (ACC), pp. 3959–3964. IEEE.

Tanabe, T., R. Sato, K. Fukuchi, J. Sakuma, and Y. Akimoto (2022a). Max-min off-policy actor-critic method focusing on worst-case robustness to model misspecification. In Advances in Neural Information Processing Systems.

Tanabe, T., R. Sato, K. Fukuchi, J. Sakuma, and Y. Akimoto (2022b). Max-min off-policy actor-critic method focusing on worst-case robustness to model misspecification. Advances in Neural Information Processing Systems 35, 6967–6981.

Tassa, Y., Y. Doron, A. Muldal, T. Erez, Y. Li, et al. (2018). Deepmind control suite.

Tessler, C., Y. Efroni, and S. Mannor (2019). Action robust reinforcement learning and applications in continuous control. In International Conference on Machine Learning, pp. 6215–6224. PMLR.

Tewari, A. and S. A. Murphy (2017). From ads to interventions: Contextual bandits in mobile health. In Mobile Health - Sensors, Analytic Methods, and Applications.

Thompson, W. R. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika 25(3-4), 285–294.

Tobin, J., R. Fong, A. Ray, J. Schneider, W. Zaremba, et al. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp. 23–30. IEEE.

Todorov, E., T. Erez, and Y. Tassa (2012). Mujoco: A physics engine for model-based control. In 2012 IEEE/RSJ international conference on intelligent robots and systems, pp. 5026–5033. IEEE.

Towers, M., J. K. Terry, A. Kwiatkowski, J. U. Balis, G. d. Cola, et al. (2023). Gymnasium.

Tsybakov, A. B. (2009). Introduction to nonparametric estimation, Volume 11. Springer.

Urpí, N. A., S. Curi, and A. Krause (2021). Risk-averse offline reinforcement learning. arXiv preprint arXiv:2102.05371.

Urteaga, I. and C. Wiggins (2018). Variational inference for the multi-armed contextual bandit. In International Conference on Artificial Intelligence and Statistics, pp. 698–706. PMLR.

v. Neumann, J. (1928). Zur theorie der gesellschaftsspiele. Mathematische annalen 100(1), 295–320.

Van der Vaart, A. W. (2000). Asymptotic statistics, Volume 3. Cambridge university press.

Van Hasselt, H., A. Guez, and D. Silver (2016). Deep reinforcement learning with double q-learning. In Proceedings of the AAAI conference on artificial intelligence, Volume 30.

Vershynin, R. (2018). High-dimensional probability: An introduction with applications in data science, Volume 47. Cambridge university press.

Vieillard, N., T. Kozuno, B. Scherrer, O. Pietquin, R. Munos, et al. (2020). Leverage the average: an analysis of kl regularization in rl. arXiv preprint arXiv:2003.14089.

Vieillard, N., O. Pietquin, and M. Geist (2020). Munchausen reinforcement learning. Advances in Neural Information Processing Systems 33, 4235–4246.

Vinitsky, E., Y. Du, K. Parvate, K. Jang, P. Abbeel, et al. (2020). Robust reinforcement learning using adversarial populations. arXiv preprint arXiv:2008.01825.

Wainwright, M. J. (2019). Stochastic approximation with cone-contractive operators: Sharp ℓ_∞-bounds for Q-learning. arXiv preprint arXiv:1905.06265.

Wang, H. and X. Y. Zhou (2020). Continuous-time mean–variance portfolio selection: A reinforcement learning framework. Mathematical Finance 30(4), 1273–1308.

Wang, K., U. Gadot, N. Kumar, K. Levy, and S. Mannor (2023). Robust reinforcement learning via adversarial kernel approximation. arXiv preprint arXiv:2306.05859.

Wang, S., N. Si, J. Blanchet, and Z. Zhou (2023). A finite sample complexity bound for distributionally robust q-learning. arXiv preprint arXiv:2302.13203.

Wang, Y. and S. Zou (2021). Online robust reinforcement learning with model uncertainty. Advances in Neural Information Processing Systems 34.

Wang, Z., T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, et al. (2016). Dueling network architectures for deep reinforcement learning. In International conference on machine learning, pp. 1995–2003. PMLR.

Wiesemann, W., D. Kuhn, and B. Rustem (2013). Robust markov decision processes. Mathematics of Operations Research 38(1), 153–183.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning 8, 229–256.

Wolff, E. M., U. Topcu, and R. M. Murray (2012). Robust control of uncertain markov decision processes with temporal logic specifications. In 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), pp. 3372–3379. IEEE.

Wu, W., J. Yang, and C. Shen (2020). Stochastic linear contextual bandits with diverse contexts. In International Conference on Artificial Intelligence and Statistics, pp. 2392–2401. PMLR.

Xie, T., N. Jiang, H. Wang, C. Xiong, and Y. Bai (2021). Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. Advances in neural information processing systems 34.

Xu, H. and S. Mannor (2012). Distributionally robust Markov decision processes. Mathematics of Operations Research 37(2), 288–300.

Xu, P., Z. Wen, H. Zhao, and Q. Gu (2020). Neural contextual bandits with deep representation and shallow exploration. arXiv preprint arXiv:2012.01780.

Xu, P., H. Zheng, E. V. Mazumdar, K. Azizzadenesheli, and A. Anandkumar (2022). Langevin monte carlo for contextual bandits. In International Conference on Machine Learning, pp. 24830–24850. PMLR.

Xu, Z., K. Panaganti, and D. Kalathil (2023). Improved sample complexity bounds for distributionally robust reinforcement learning. In International Conference on Artificial Intelligence and Statistics, pp. 9728–9754. PMLR.

Yan, Y., G. Li, Y. Chen, and J. Fan (2022). The efficacy of pessimism in asynchronous Q-learning. arXiv preprint arXiv:2203.07368.

Yan, Y., G. Li, Y. Chen, and J. Fan (2023). The efficacy of pessimism in asynchronous q-learning. IEEE Transactions on Information Theory.

Yang, I. (2017). A convex optimization approach to distributionally robust markov decision processes with wasserstein distance. IEEE Control Systems Letters 1, 164–169.

Yang, K., L. Yang, and S. Du (2021). Q-learning with logarithmic regret. In International Conference on Artificial Intelligence and Statistics, pp. 1576–1584. PMLR.

Yang, W., H. Wang, T. Kozuno, S. M. Jordan, and Z. Zhang (2023). Avoiding model estimation in robust markov decision processes with a generative model. arXiv preprint arXiv:2302.01248.

Yang, W., L. Zhang, and Z. Zhang (2021). Towards theoretical understandings of robust markov decision processes: Sample complexity and asymptotics. arXiv preprint arXiv:2105.03863.

Yang, W., L. Zhang, and Z. Zhang (2022). Toward theoretical understandings of robust markov decision processes: Sample complexity and asymptotics. The Annals of Statistics 50(6), 3223–3248.

Yang, W. H. (1991). On generalized holder inequality.

Yin, M., Y. Bai, and Y.-X. Wang (2021). Near-optimal offline reinforcement learning via double variance reduction. arXiv preprint arXiv:2102.01748.

Ying, C., X. Zhou, H. Su, D. Yan, and J. Zhu (2021). Towards safe reinforcement learning via constraining conditional value-at-risk.

Yu, T., B. Kveton, Z. Wen, R. Zhang, and O. J. Mengshoel (2020). Graphical models meet bandits: A variational thompson sampling approach. In International Conference on Machine Learning, pp. 10902–10912. PMLR.

Yu, T., D. Quillen, Z. He, R. Julian, A. Narayan, et al. (2021). Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning.

Yu, W., C. K. Liu, and G. Turk (2018). Policy transfer with strategy optimization. International Conference On Learning Representations.

Zenati, H., A. Bietti, E. Diemert, J. Mairal, M. Martin, et al. (2022). Efficient kernelized ucb for contextual bandits. In International Conference on Artificial Intelligence and Statistics, pp. 5689–5720. PMLR.

Zhang, H., H. Chen, D. S. Boning, and C.-J. Hsieh (2021). Robust reinforcement learning on state observations with learned optimal adversary. In International Conference on Learning Representations.

Zhang, H., H. Chen, C. Xiao, B. Li, M. Liu, et al. (2020). Robust deep reinforcement learning against adversarial perturbations on state observations. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin (Eds.), Advances in Neural Information Processing Systems, Volume 33, pp. 21024–21037.

Zhang, J. and P. Weng. Safe distributional reinforcement learning.

Zhang, R., Y. Hu, and N. Li (2023). Regularized robust mdps and risk-sensitive mdps: Equivalence, policy gradient, and sample complexity. arXiv preprint arXiv:2306.11626.

Zhang, S., B. Liu, and S. Whiteson (2021). Mean-variance policy iteration for risk-averse reinforcement learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Volume 35, pp. 10905–10913.

Zhang, T. (2022). Feel-good thompson sampling for contextual bandits and reinforcement learning. SIAM Journal on Mathematics of Data Science 4(2), 834–857.

Zhang, W., D. Zhou, L. Li, and Q. Gu (2020). Neural thompson sampling. arXiv preprint arXiv:2010.00827.

Zhang, Z., Y. Zhou, and X. Ji (2020). Almost optimal model-free reinforcement learning via reference-advantage decomposition. Advances in Neural Information Processing Systems 33.

Zhao, C., O. Sigaud, F. Stulp, and T. M. Hospedales (2019). Investigating generalisation in continuous deep reinforcement learning. arXiv preprint arXiv:1902.07015.

Zhou, D., L. Li, and Q. Gu (2020). Neural contextual bandits with ucb-based exploration. In International Conference on Machine Learning, pp. 11492–11502. PMLR.

Zhou, Z., Q. Bai, Z. Zhou, L. Qiu, J. Blanchet, et al. (2021). Finite-sample regret bound for distributionally robust offline tabular reinforcement learning. In International Conference on Artificial Intelligence and Statistics, pp. 3331–3339. PMLR.

Ziegler, D. M., N. Stiennon, J. Wu, T. B. Brown, A. Radford, et al. (2019). Fine-tuning language models from human preferences. arXiv preprint arXiv:1909.08593.

Zouitine, A., D. Bertoin, P. Clavier, M. Geist, and E. Rachelson (2024a). Rrls: Robust reinforcement learning suite. arXiv preprint arXiv:2406.08406.

Zouitine, A., D. Bertoin, P. Clavier, M. Geist, and E. Rachelson (2024b). Time-constrained robust mdps.

Zouitine, A., E. Rachelson, and M. Geist (2023). Revisiting the static model in robust reinforcement learning. In Sixteenth European Workshop on Reinforcement Learning.

Appendix

Appendix of Chapter 2

1 Overview and useful inequalities

The appendix is organized as follows

- In Appendix 1.1, a comprehensive table with state-of-the-art complexity for every distance.
- In Appendix 1.2, we provide more details/explanations on the difference between our formulation on the one of Kumar et al. (2022) and Derman et al. (2021).
- In Appendix 1.3, we give more details about our algorithm : `DRVI Lp`.
- In Appendix 1.4, we give some useful inequalities frequently used in the proofs.
- In Appendix 2, we prove Theorem 2.4.1.
- In Appendix 3, we prove Theorem 2.5.1.

Finally, the proofs for the *s*-rectangular and *sa*-rectangular cases are often very similar. If this is true, we will combine them in a single proof with the two cases detailed when needed.

1.1 Table of sample Complexity

Table 9.1: Sample Complexity for different metric and *s*- or *sa* rectangular assumptions with σ the radius of uncertainty set, H the horizon factor, ϵ the precicion, $\bar{p}$, $\sigma_{0,p} = (1-\gamma)/(2\gamma S^{1/q})$. the smallest positive state transition probability of the nominal kernel visited by the optimal robust policy (see Yang et al. (2021)).

	Panaganti and Kalathil (2022a)	Yang et al. (2021)	Shi and Chi (2022)	Our $\sigma \geq 0$	Our $\sigma_{0,p} > \sigma > 0$	Shi et al. (2023) $\sigma > 1-\gamma$	Shi et al. (2023) $0 < \sigma < 1-\gamma$
TV (sa)	$\tilde{\mathcal{O}}\left(\frac{S^2AH^4}{\epsilon^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{S^2AH^4(2+\sigma)^2}{\epsilon^2\sigma^2}\right)$	$\times$	$\tilde{\mathcal{O}}\left(\frac{SAH^4}{\epsilon^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{SAH^3}{\epsilon^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{SAH^2}{\epsilon^2\sigma}\right)$	$\tilde{\mathcal{O}}\left(\frac{SAH^3}{\epsilon^2}\right)$
TV (s)	$\times$	$\tilde{\mathcal{O}}\left(\frac{S^2A^2H^4(2+\sigma)^2}{\epsilon^2\sigma^2}\right)$	$\times$	$\tilde{\mathcal{O}}\left(\frac{SAH^4}{\epsilon^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{SAH^3}{\epsilon^2}\right)$	$\times$	$\times$
L_p (sa)	$\times$	$\times$	$\times$	$\tilde{\mathcal{O}}\left(\frac{SAH^4}{\epsilon^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{SAH^3}{\epsilon^2}\right)$	$\times$	$\times$
L_p (s)	$\times$	$\times$	$\times$	$\tilde{\mathcal{O}}\left(\frac{SAH^4}{\epsilon^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{SAH^3}{\epsilon^2}\right)$	$\times$	$\times$
χ^2 (sa)	$\tilde{\mathcal{O}}\left(\frac{S^2A\sigma H^4}{\epsilon^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{\lvert\mathcal{S}\rvert^2\lvert\mathcal{A}\rvert(1+\sigma)^2H^4}{\varepsilon^2(\sqrt{1+\sigma}-1)^2}\right)$	$\times$	$\times$	$\times$	$\tilde{\mathcal{O}}\left(\frac{SA\sigma H^4}{\epsilon^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{SA\sigma H^4}{\epsilon^2}\right)$
χ^2 (s)	$\times$	$\tilde{\mathcal{O}}\left(\frac{\lvert\mathcal{S}\rvert^2\lvert\mathcal{A}^3\rvert(1+\sigma)^2H^4}{\varepsilon^2(\sqrt{1+\sigma}-1)^2}\right)$	$\times$	$\times$	$\times$		$\times$
KL (sa)	$\tilde{\mathcal{O}}\left(\frac{\lvert\mathcal{S}\rvert^2\lvert\mathcal{A}\rvert\exp(H)H^4}{\sigma^2\varepsilon^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{S^2AH^4}{\bar{p}^2\epsilon^2\sigma^2}\right)$	$\tilde{\mathcal{O}}\left(\frac{SAH^4}{\bar{p}\epsilon^2\sigma^4}\right)$	$\times$	$\times$	$\times$	$\times$
KL (s)	$\times$	$\tilde{\mathcal{O}}\left(\frac{S^2A^2H^4}{\bar{p}^2\epsilon^2\sigma^2}\right)$	$\times$	$\times$	$\times$	$\times$	$\times$

1.2 Relation with the work of Kumar et al. (2022) and Derman et al. (2021)

In the work of Derman et al. (2021) close forms for RMDPs with L_p norms are derived assuming the following uncertainty set :

Assumption 1.1. *(sa-rectangularity in Derman et al. (2021))*

$$\mathcal{U}^{sa,\sigma}_{\|.\|_p}(P^0) := (r_0 + \mathcal{R}) \times \left(P^0 + \mathcal{P}\right), \mathcal{R} = \times_{s\in\mathcal{S},a\in\mathcal{A}}\mathcal{R}_{s,a}, \mathcal{R}_{s,a} = \left\{r_{s,a} \in \mathbb{R} \mid \|r_{s,a}\|_p \leq \alpha_{s,a}\right\}$$
$$\mathcal{P} = \times_{s\in\mathcal{S},a\in\mathcal{A}}\mathcal{P}_{s,a}\mathcal{P}_{s,a} = \{P_{s,a} : \mathcal{S} \to \mathbb{R}, \|P_{s,a}\|_p \leq \sigma_{s,a}\}$$

Using these uncertainty sets leads to the following Bellman Operator :

Theorem 1.2 (Derman et al. (2021))**.** *The sa-rectangular Robust Bellman operator is equivalent to a regularized non-robust Bellman operator: for* $r^{s,a}_{V,\pi}(s,a) = -\left(\alpha_{s,a} + \gamma\sigma_{s,a}\|V\|_q\right) + r_0(s,a)$ *as we have*

$$\mathcal{T}^{\pi,\sigma}V(s) = \langle \pi_s, r^{s,a}_{V,\pi}(s,a) + \gamma \sum_{s'} P^0\left(s' \mid s,a\right) V\left(s'\right)\rangle_A$$

Using this formulation, they get a closed form for the inner minimization problem and for the Robust Bellman Operator

The work Kumar et al. (2022) modifies the work of Derman et al. (2021) using Kernel that sum to 1, $\sum_{s'} P_{s,a}(s') = 0$ in their definition, but using this uncertainty set, it is still possible to get a robust kernel out of the simplex. Using this formulation, they also get a closed form for the inner minimization problem and for the Robust Bellman Operator.

Assumption 1.3. *(sa-rectangularity in Kumar et al. (2022))*

$$\mathcal{U}^{sa,\sigma}_{\|.\|_p}(P^0) := (r_0 + \mathcal{R}) \times \left(P^0 + \mathcal{P}\right), \mathcal{R} = \times_{s\in\mathcal{S},a\in\mathcal{A}}\mathcal{R}_{s,a}, \mathcal{R}_{s,a} = \left\{r_{s,a} \in \mathbb{R} \mid \|r_{s,a}\|_p \leq \alpha_{s,a}\right\}$$
$$\mathcal{P} = \times_{s\in\mathcal{S},a\in\mathcal{A}}\mathcal{P}_{s,a}\mathcal{P}_{s,a} = \{P_{s,a} : \mathcal{S} \to \mathbb{R} \mid \sum_{s'} P_{s,a}\left(s'\right) = 0, \|P_{s,a}\|_p \leq \sigma_{s,a}\}$$

Using these uncertainty sets where robust Kernel may not belong anymore to the simplex as they do not assume $P^0 + P_{s,a} \geq 0$. This leads to the following Bellman Operator :

Theorem 1.4 (Kumar et al. (2022))**.** *The sa-rectangular Robust Bellman operator is equivalent to a regularized non-robust Bellman operator: for* $r^{s,a}_{V,\pi}(s,a) = -\left(\alpha_{s,a} + \gamma\sigma_{s,a}\mathrm{sp}_q(V)\right) + r_0(s,a)$*, as we have*

$$\mathcal{T}^{\pi,\sigma}V(s) = \langle \pi_s, r^{s,a}_{V,\pi}(s,a) + \gamma \sum_{s'} P^0\left(s' \mid s,a\right) V\left(s'\right)\rangle_A$$

where $\mathrm{sp}_q(V)$ in defined in Def. 2.3.1.These results are due to the following lemma.

Lemma 1.5 (Kumar et al. (2022). Duality for the minimization problem for *sa* rectangular case with L_p norm without simplex constrain)**.**

$$\inf_{P:\sum_{s'} P(s')=0 \left\|P-\hat{P}_{s,a}\right\|_p \leq \sigma_{s,a}} PV = \widehat{P}_{s,a}V - \sigma_{s,a}\mathrm{sp}_q(V)$$

Our analysis assumes the positivity of the kernel function, $P^0 + P_s \geq 0$ in s-rectangular or $P^0 + P_{s,a} \geq 0$ for *sa*-rectangular case. Using this more realistic assumption, we can not obtain a closed form of the robust Bellman operator. However, we are still able to compute a dual form for the inner minimization problem of RMDPs. With our definition of rectangularity in the simplex:

Assumption 1.6. *(sa-rectangularity) We define sa-rectangular L_p-constrained uncertainty set as*

$$\mathcal{U}^{sa,\sigma}_{\|.\|_p}(P^0) := (r_0 + \mathcal{R}) \times \left(P^0 + \mathcal{P}\right),$$
$$\mathcal{R} = \times_{s\in\mathcal{S},a\in\mathcal{A}}\mathcal{R}_{s,a}, \mathcal{P} = \times_{s\in\mathcal{S},a\in\mathcal{A}}\mathcal{P}_{s,a}, \mathcal{R}_{s,a} = \{r_{s,a} \in \mathbb{R} \mid |r_{s,a}| \leq \alpha_{s,a}\}$$
$$\mathcal{P}_{s,a} = \{P_{s,a} : \mathcal{S} \to \mathbb{R} \mid \sum_{s'} P_{s,a}(s') = 0, P_{0,s,a} + P_{s,a} \geq 0, , \|P_{s,a}\|_p \leq \sigma_{s,a}\}$$

and using $\kappa_{\mathcal{D}}(v) = \inf\left\{u^\top v : u \in \mathcal{D}\right\}$, we obtain :

Lemma 1.7 (Duality for the minimization problem for *sa* rectangular case with L_p norm)**.**

$$\kappa_{\hat{\mathcal{P}}_{s,a}}(V) = \max_{\mu\geq 0}\{\widehat{P}_{s,a}(V-\mu) - \sigma_{s,a}\mathrm{sp}_q(V-\mu)\}$$

Proof can be found on Appendix 2.5. Contrary to previous lemma in Kumar et al. (2022), there is an additional max operator in our dual formulation. Interestingly, their formulation is a relaxation of our Lemmas 2.3.3 as their formulation does not assume the positivity of the kernel. Their relaxation allows practical algorithms with close form, but still suffer from non-exact formulation of RMDPs with robust Kernel that are not in the simplex.

One crucial point in our analysis is that Bellman Operator for RMDPs is a γ- contraction for robust kernel in the simplex for any radius σ (see Iyengar (2005)). For Kumar et al. (2022) and Derman et al. (2021) the range of σ where their Robust Bellman Operator is a contraction is smaller than $\frac{1-\gamma}{\gamma S^{1/q}}$ (see Proposition 4 of Derman et al. (2021)) which is the range where we have minimax optimality in our Theorem 2.5.1. For $\sigma > \frac{1-\gamma}{\gamma S^{1/q}}$, there is no contraction anymore. In the following, we will assume that robust kernels belong to the simplex to use γ-contraction in our proof of sample complexity and ensure convergence of the following Distributionally Robust value Iteration for L_p norms for any σ Algoritm 11.

1.3 Model based DRVI L_p algorithm

Algorithm 10: DRVI L_p: Distributionally robust value iteration DRVI for L_P norms with *sa*−rectangular assuptions

input: empirical nominal transition kernel $\widehat{P}_0$; reward function r; uncertainty level σ.
initialization: $\widehat{Q}_0(s,a) = 0$, $\widehat{V}_0(s) = 0$ for all $(s,a) \in S \times A$.
for $t = 1, 2, \cdots, T$ **do**
  for $\forall s \in S, a \in A$ **do**
    Set $\widehat{Q}_t(s,a)$ according to (A.309) for *sa*−rectangular ;
  for $\forall s \in S$ **do**
    Set $\widehat{V}_t(s) = \max_a \widehat{Q}_t(s,a)$;
output: $\widehat{Q}_T$, $\widehat{V}_T$ and $\widehat{\pi}$ obeying $\widehat{\pi}(s) = \arg\max_a \widehat{Q}_T(s,a)$.

We propose Alg. 11 to solve robust MDPs in the case of L_P norms using value Iteration with *sa*- rectangularity assumptions. First, we can remark that directly solving classical RMDPs formulation is computationally costly as it requires an optimization over an S-dimensional probability simplex at each iteration, especially when the dimension of the state space S is large. However, using strong duality like Iyengar (2005) for the TV, one can also solve using

the dual problem of this formulation. The equivalence between the two formulations can be found in Lemma 2.3.3. Using the dual form, the optimization (A.2) reduces to a 2-dimensional optimization problem that can be solved efficiently using any 2−dimensional convex solver if there exists an analytic form of the span-semi norm. Then the iterates $\{\widehat{Q}_t\}_{t\geq 0}$ of DRVI for L_P norms converge linearly to the fixed point $\widehat{Q}^\star$, owing to the appealing γ-contraction property of robust MDPs in the simplex. From an initialization $\widehat{Q}_0 = 0$, the update rule at the t-th ($t \geq 1$) iteration can be formulated as for sa-rectangular case as:

$$\forall (s,a) \in S \times A: \quad \widehat{Q}_t(s,a) = r(s,a) + \max_{\mu \geq 0} \hat{P}(\hat{V}_{t-1} - \mu) - \sigma_{s,a}\mathrm{sp}_q(\hat{V}_{t-1} - \mu) \tag{A.1}$$

$$= r(s,a) + \max_{\alpha_{\hat{P}}^{\lambda,\omega} \in \mathrm{A}_{\hat{P}}^{\lambda,\omega}} \widehat{P}[\widehat{V}_{t-1}]_{\alpha_{\hat{P}}^{\lambda,\omega}} - \sigma_{s,a}\mathrm{sp}_q([\widehat{V}_{t-1}]_{\alpha_{\hat{P}}^{\lambda,\omega}}) \tag{A.2}$$

where the variational family $\mathrm{A}_{\hat{P}}^{\lambda,\omega}$ is a 2−dimensional variational family defined in (A.11). The specific form of the dual problem depends on the choice of the norm. In the case of L_1, L_2, or L_∞, span semi-norms involved in dual problems have closed form (respectively equals to median, variance, or span), and equation A.2 corresponds to a 2-D minimization problem.

But in general cases, one has to compute span-semi norms that can be easily computed using binary search solving

$$\sum_s \mathrm{sign}\left(v(s) - \omega_p(v)\right) |v(s) - \omega_p(v)|^{\frac{1}{p-1}} = 0$$

to compute ω_q and then setting the semi norm $\mathrm{sp}_q(v) = \|v - \omega_q\|$. Recall the q-variance function $\mathrm{sp}_q : \mathcal{S} \to \mathbb{R}$ and q-mean function $\omega_q : \mathcal{S} \to \mathbb{R}$ be defined as

$$\mathrm{sp}_q(v) := \min_{\omega \in \mathbb{R}} \|v - \omega \mathbf{1}\|_q, \quad \omega_q(v) := \arg\min_{\omega \in \mathbb{R}} \|v - \omega \mathbf{1}\|_q.$$

See Kumar et al. (2022) for discussion about computing span semi norms. So in the general case, we can also compute the maximum solving :

$$\forall (s,a) \in S \times A: \quad \widehat{Q}_t(s,a) = r(s,a) + \max_{\alpha_{\hat{P}}^{\lambda,\omega} \in \mathrm{A}_{\hat{P}}^{\lambda,\omega}} \widehat{P}[\widehat{V}_{t-1}]_{\alpha_{\hat{P}}^{\lambda,\omega}} - \sigma_{s,a} \left\| [\widehat{V}_{t-1}]_{\alpha_{\hat{P}}^{\lambda,\omega}} - w \right\|_q,$$

Using any 2−D convex optimization algorithm solves the problem as this problem is jointly concave in (λ, w) because $(\lambda, w) \to -\left\| [\widehat{V}_{t-1}]_{\alpha_{\hat{P}}^{\lambda,\omega}} - w \right\|_q$ is concave using norm property and $(\lambda, w) \to \widehat{P}[\widehat{V}_{t-1}]_{\alpha_{\hat{P}}^{\lambda,\omega}}$ also. Then the sum is concave.

Finally, in the sa-case we compute the best policy which is the greedy policy of the final Q-estimates $\widehat{Q}_T$ as the final policy $\widehat{\pi}$:

$$\forall s \in S: \quad \widehat{\pi}(s) = \arg\max_a \widehat{Q}_T(s,a).$$

1.4 Useful Inequalities and notations

Here we present some useful inequalities used frequently in the derivation. Consider any P a transition matrix and σ_s for s rectangular uncertain sets or σ_{sa} for sa- uncertainty sets, then for $\mathbf{1} = (1, 1, ..., 1)^\top$:

$$(1-\gamma P)^{-1}(\gamma\sigma_s)\mathbf{1} < \frac{\sigma}{1-\gamma}\mathbf{1} \text{ and } (1-\gamma P)^{-1}\mathbf{1} \leq \frac{1}{1-\gamma}\mathbf{1} \tag{A.3}$$

$$\forall q \in \mathbb{N}^*, \quad \mathrm{sp}_q(.) \leq 2\|.\|_q < 2S^{1/q}\|.\|_\infty, \quad \mathrm{sp}(.)_\infty \leq 2\|.\|_\infty \tag{A.4}$$

$$\mathrm{sp}_q(.) \leq 2\|.\|_q \leq 2\|.\|_q \tag{A.5}$$

Eq. (A.3) is true, taking the supremum norm of the left-hand side inequality. Eq. (A.4) and Eq. (A.5) come from properties of norms, see Eq. (1) from Scherrer (2013). Finally we denote the truncation operator for a vector $\alpha \in \mathbb{R}^S$,

$$[V]_\alpha := \begin{cases} \alpha(s), & \text{if } V(s) > \alpha(s) \\ V(s), & \text{otherwise.} \end{cases}$$

1.5 Robust Bellman Operator and robust Q values

This is proof of Lemma 2.3.5:

Lemma 1.8. *Robust Bellman Operator for sa− and s− rectangular are :*

$$\mathcal{T}^{\pi,\sigma}V(s) = \sum_a \pi(a|s)\Big(-\alpha_{s,a} + r_0(s,a) + \gamma\sum_{s'}P^0(s',s,a)v(s') + \gamma\min_{P\in\mathcal{P}_{s,a}}PV\Big)$$
$$\mathcal{T}^{\pi,\tilde{\sigma}}V(s) = -\|\pi_s\|_q\alpha_s + \gamma\min_{P^\pi\in\mathcal{P}_s}P^\pi V + \sum_a \pi(a|s)\Big(r_0(s,a) + \gamma P^0(s'|s,a)V(s')\Big)$$

Proof. For *sa*-rectangular: by rectangularity

$$\begin{aligned}\mathcal{T}^{\pi,\sigma}V(s) &= \sum_a \pi(a|s)\Big(-\alpha_{s,a} + r_0(s,a) + \gamma\min_{P\in\mathcal{P}^0_{s,a}}PV\Big)\\ &= \sum_a \pi(a|s)\Big(-\alpha_{s,a} + r_0(s,a) + \gamma\min_{P\in\mathcal{P}_{s,a}}PV + P_{0,s,a}V\Big)\end{aligned}$$

For $s-$rectangular case :

$$\begin{aligned}\mathcal{T}^{\pi,\tilde{\sigma}}V(s) &= \min_{P^\pi\in\mathcal{P}^0_s}\gamma PV + \min_{R\in\mathcal{R}^0_s}\sum_a \pi(a|s)R(s,a)\\ &= \sum_a \pi(a|s)r_0(s,a) + \min_{R\in\mathcal{R}_s}\sum_a \pi(a|s)R(s,a) + \sum_a \pi(a|s)\gamma\sum_{s'}P^0(s'|s,a)V(s')\\ &+ \min_{P^\pi\in\mathcal{P}_{s,}}\gamma P^\pi V\\ &\overset{(a)}{=} \sum_a \pi(a|s)\Big(r_0(s,a) + \sum_{s'}P^0(s'|s,a)V(s')\Big) - \alpha_s\|\pi_s\|_q + \min_{P^\pi\in\mathcal{P}_s}\gamma P^\pi V\end{aligned}$$

where (a) comes from Holder's inequality. □

Lemma 1.9. *For sa− and s− rectangular,*

$$Q^{\pi,\sigma}(s,a) = r_{Q^\pi}^{(s,a)} + \gamma P^0_{s,a}V^{\pi,\sigma}, \tag{A.6}$$
$$Q^{\pi,\tilde{\sigma}}(s,a) = r^s_{Q^\pi} + \gamma P^0_{s,a}V^{\pi,\tilde{\sigma}} \tag{A.7}$$

with

$$r^{(s,a)}_{Q^\pi} = r_0(s,a) - \alpha_{s,a} + \gamma \min_{P\in\mathcal{P}_{s,a}} PV^{\pi,\sigma} \tag{A.8}$$

$$r^{s}_{Q^\pi} = r_0(s,a) - \Big(\frac{\pi_s(a)}{\|\pi_s\|_q}\Big)^{q-1}\alpha_s + \gamma \min_{P^\pi\in\mathcal{P}_s} P^\pi V^{\pi,\tilde\sigma} \tag{A.9}$$

Proof. The result comes directly as for sa-rectangular the following relations hold,

$$V^{\pi,\sigma}(s) = \sum_a \pi(a|s) Q^{\pi,\sigma}(s,a) \quad \text{and}$$

and for s-rectangular case

$$V^{\pi,\tilde\sigma}(s) = \sum_a \pi(a|s) Q^{\pi,\tilde\sigma}(s,a).$$

Then using fixed point equation of Bellman operator: $\mathcal{T}^{\pi,\sigma}V^{\pi,\sigma}(s) = V^{\pi,\sigma}(s)$ or $\mathcal{T}^{\pi,\sigma}V^{\pi,\tilde\sigma}(s) = V^{\pi,\tilde\sigma}(s)$ and previous Lemma 1.8 for the expression of $\mathcal{T}^{\pi,\sigma}V^{\pi,\sigma}(s)$, we can identify the robust Q values that give the result

□

2 An H^4 bound for L_p-balls

To lighten notations, we remove superscript σ or $\tilde\sigma$ in most places and denote for example V^π instead of $V^{\pi,\sigma}$ for sa-rectangular sets.

Lemma 2.1 (Decomposition of the bound)**.**

$$\left\|Q^* - Q^{\hat\pi}\right\|_\infty \le \left\|Q^* - \hat Q^{\pi^*}\right\|_\infty + \left\|\hat Q^{\pi^*} - \hat Q^{\hat\pi}\right\|_\infty + \left\|\hat Q^{\hat\pi} - Q^{\hat\pi}\right\|_\infty$$

Proof.

$$\begin{aligned} 0 \le Q^* - Q^{\hat\pi} &= Q^* - \underbrace{\hat Q^*}_{\ge \hat Q^{\pi_*}} + \hat Q^* - \hat Q^{\hat\pi} + \hat Q^{\hat\pi} - Q^{\hat\pi} \\ &\le Q^* - \hat Q^{\pi_*} + \hat Q^* - \hat Q^{\hat\pi} + \hat Q^{\hat\pi} - Q^{\hat\pi} \\ \Rightarrow \|Q^* - Q^{\hat\pi}\|_\infty &\le \|Q^* - \hat Q^{\pi_*}\|_\infty + \|\hat Q^* - \hat Q^{\hat\pi}\|_\infty + \|\hat Q^{\hat\pi} - Q^{\hat\pi}\|_\infty \end{aligned}$$

□

This decomposition is the starting point of our proofs for both Theorems 2.4.1 and 2.5.1. In this decomposition, the second term satisfies $\|Q^* - \hat Q^{\pi^*}\|_\infty \le \epsilon_{\text{opt}}$ by definition. This term goes to 0 exponentially fast as the robust Bellman operator is a γ-contraction. The two last terms $\|Q^* - \hat Q^{\pi^*}\|_\infty$ and $\|\hat Q^{\hat\pi} - Q^{\hat\pi}\|_\infty$ need to be controlled using concentration inequalities between the true MDP and the estimated one. To do so, we need concentration inequalities such as the following Lemma 2.2.

Lemma 2.2 (Hoeffding's inequality for V)**.** *For any $V \in \mathbb{R}^{|\mathcal{S}|}$ with $\|V\|_\infty \le H$, with probability at least $1-\delta$, we have*

$$\max_{(s,a)} \left|P^0 V - \widehat{P} V\right| \le H\sqrt{\frac{\log(2|\mathcal{S}\|\mathcal{A}|/\delta)}{2N}}.$$

Proof. For any (s,a) pair, assume a discrete random variable taking value $V(i)$ with probability $P^0_{s,a}(i)$ for all $i \in \{1,2,\cdots,|\mathcal{S}|\}$. Using Hoeffding's inequality (Hoeffding 1994) and $\|V\|_\infty \leq H$:

$$\mathbb{P}\left(P^0V - \widehat{P}V \geq \varepsilon\right) \leq \exp\left(-N\varepsilon^2/(2H^2)\right) \quad \text{and} \quad \mathbb{P}\left(\widehat{P}V - P^0V \geq \varepsilon\right) \leq \exp\left(-N\varepsilon^2/(2H^2)\right).$$

Then, taking $\varepsilon = H\sqrt{\frac{2\log(2|\mathcal{S}||\mathcal{A}|/\delta)}{N}}$, we get

$$\mathbb{P}\left(\left|P^0V - \widehat{P}V\right| \geq H\sqrt{\frac{\log(2|\mathcal{S}||\mathcal{A}|/\delta)}{N}}\right) \leq \frac{\delta}{|\mathcal{S}||\mathcal{A}|}.$$

Finally, using a union bound:

$$\mathbb{P}\left(\max_{(s,a)}\left|P^0V - \widehat{P}V\right| \geq H\sqrt{\frac{2\log(2|\mathcal{S}||\mathcal{A}|/\delta)}{N}}\right) \leq \sum_{s,a}\mathbb{P}\left(\left|P^0V - \widehat{P}V\right| \geq H\sqrt{\frac{2\log(2|\mathcal{S}||\mathcal{A}|/\delta)}{N}}\right) \leq \delta.$$

□

This completes the concentration proof. Next we will look at the contraction argument of the robust Bellman operator.

Lemma 2.3 (Contraction of infimum operator)**.** *For $\mathcal{D} = \mathcal{P}_{s,a}$ or $\mathcal{P}_s$, the function*

$$\forall s,a, \quad v \mapsto \kappa_{\mathcal{D}}(v) = \inf\left\{u^\top v : u \in \mathcal{D}\right\}$$

is 1-Lipchitz.

Proof. We have that

$$\begin{aligned}\forall (s,a) \in \mathcal{S}\times\mathcal{A}, \quad \kappa_{\mathcal{P}_{s,a}}(V_2) - \kappa_{\mathcal{P}_{s,a}}(V_1) &= \inf_{p\in\mathcal{P}_{s,a}} p^\top V_2 - \inf_{\tilde{p}\in\mathcal{P}_{s,a}} \tilde{p}^\top V_1 = \inf_{p\in\mathcal{P}_{s,a}}\sup_{\tilde{p}\in\mathcal{P}_{s,a}} p^\top V_2 - \tilde{p}^\top V_1 \\ &\geq \inf_{p\in\mathcal{P}_{s,a}} p^\top (V_2 - V_1) = \kappa_{\mathcal{P}_{s,a}}(V_2 - V_1).\end{aligned}$$

Then $\forall \varepsilon > 0$, there exists $P_{s,a} \in \mathcal{P}_{s,a}$ such that

$$P_{s,a}^\top (V_2 - V_1) - \varepsilon \leq \kappa_{\mathcal{P}_{s,a}}(V_2 - V_1).$$

Using those two properties,

$$\kappa_{\mathcal{P}_{s,a}}(V_1) - \kappa_{\mathcal{P}_{s,a}}(V_2) \leq P_{s,a}^\top (V_1 - V_2) + \varepsilon \leq \|P_{s,a}\|_1 \|V_1 - V_2\| + \varepsilon = \|V_1 - V_2\| + \varepsilon,$$

where we used the Holder's inequality. Since ε is arbitrary small, we obtain, $\kappa_{\mathcal{P}_{s,a}}(V_1) - \kappa_{\mathcal{P}_{s,a}}(V_2) \leq \|V_1 - V_2\|$. Exchanging the roles of V_1 and V_2 give the result. The proof is similar for $\mathcal{P}_s$. □

Note that an immediate consequence is the already known γ- contraction of the robust Bellman operator.

Lemma 2.4 (Upper-bounds of $\left\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\right\|_\infty$ and $\left\|Q^* - \hat{Q}^{\pi^*}\right\|_\infty$)**.**

$$\begin{aligned}\left\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\right\|_\infty &\leq \frac{\gamma}{1-\gamma}\max_{s,a}\left|\kappa_{\hat{\mathcal{P}}_{s,a}}(\hat{V}^{\hat{\pi}}) - \kappa_{\mathcal{P}_{0,s,a}}(\hat{V}^{\hat{\pi}})\right|, \\ \left\|Q^* - \hat{Q}^{\pi^*}\right\|_\infty &\leq \frac{\gamma}{1-\gamma}\max_{s,a}\left|\kappa_{\widehat{\mathcal{P}}_{s,a}}(V^*) - \kappa_{\mathcal{P}_{0,s,a}}(V^*)\right|.\end{aligned}$$

Proof. For the first inequality, since we can rewrite the robust Q-function for any uncertainty sets on the dynamics as $Q^{\hat{\pi}}(s,a) = r - \alpha_{s,a} + \gamma\kappa_{\mathcal{P}_{0,s,a}}\left(V^{\hat{\pi}}\right)$ (see Eq. (2.3.5)), or replacing $\alpha_{s,a}$ by $\alpha_s\left(\frac{\hat{\pi}_s(a)}{\|\hat{\pi}_s\|_q}\right)^{q-1}$ in the s- rectangular case:

$$\begin{aligned} Q^{\hat{\pi}}(s,a) - \hat{Q}^{\hat{\pi}}(s,a) &\overset{(a)}{=} \gamma\kappa_{\mathcal{P}_{0,s,a}}\left(V^{\hat{\pi}}\right) - \gamma\kappa_{\hat{\mathcal{P}}_{s,a}}\left(\hat{V}^{\hat{\pi}}\right) \\ &= \gamma\left(\kappa_{\mathcal{P}_{0,s,a}}\left(V^{\hat{\pi}}\right) - \kappa_{\mathcal{P}_{0,s,a}}\left(\hat{V}^{\hat{\pi}}\right)\right) + \gamma\left(\kappa_{\mathcal{P}_{0,s,a}}\left(\hat{V}^{\hat{\pi}}\right) - \kappa_{\hat{\mathcal{P}}_{s,a}}\left(\hat{V}^{\hat{\pi}}\right)\right) \end{aligned}$$

with $\mathcal{P}_{s,a}$ defined in Assumption 2.3.1 and $\hat{\mathcal{P}}_{s,a}$ with the same definition but centered around the empirical MDP. Hence, taking the supremum norm $\|.\|_\infty$,

$$\begin{aligned} \left\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\right\|_\infty &= \max_{s,a}\left|\gamma\left(\kappa_{\mathcal{P}_{0,s,a}}\left(V^{\hat{\pi}}\right) - \kappa_{\mathcal{P}_{0,s,a}}\left(\hat{V}^{\hat{\pi}}\right)\right) + \gamma\left(\kappa_{\mathcal{P}_{0,s,a}}\left(\hat{V}^{\hat{\pi}}\right) - \kappa_{\hat{\mathcal{P}}_{s,a}}\left(\hat{V}^{\hat{\pi}}\right)\right)\right| \\ &\overset{(b)}{\leq} \gamma\left\|V^{\hat{\pi}} - \hat{V}^{\hat{\pi}}\right\|_\infty + \max_{s,a}\left|\gamma\left(\kappa_{\mathcal{P}_{0,s,a}}\left(\hat{V}^{\hat{\pi}}\right) - \kappa_{\hat{\mathcal{P}}_{s,a}}\left(\hat{V}^{\hat{\pi}}\right)\right)\right| \\ &\leq \gamma\left\|V^{\hat{\pi}} - \hat{V}^{\hat{\pi}}\right\|_\infty + \gamma\max_{s,a}\left|\kappa_{\hat{\mathcal{P}}_{s,a}}(\hat{V}^{\hat{\pi}}) - \kappa_{\mathcal{P}_{0,s,a}}(\hat{V}^{\hat{\pi}})\right| \\ &\overset{(c)}{\leq} \gamma\left\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\right\|_\infty + \gamma\max_{s,a}\left|\kappa_{\hat{\mathcal{P}}_{s,a}}(\hat{V}^{\hat{\pi}}) - \kappa_{\mathcal{P}_{0,s,a}}(\hat{V}^{\hat{\pi}})\right|. \end{aligned}$$

Line (a) comes from the rectangularity assumption, (b) uses the triangular inequality and the 1-contraction of the infimum in Lemma 2.3, (c) uses the fact that $\|V^\pi - \widehat{V}^\pi\|_\infty \leq \|Q^\pi - \widehat{Q}^\pi\|_\infty$ for any π. As $1-\gamma < 1$, we get the first stated result.

One can note that the proof is true for any policy, so it is also true for both $\hat{\pi}$ and π^* which concludes the proof. This proof is written for the sa-rectangular assumption, it is also true for the s-rectangular case with slightly different notations, replacing $\mathcal{D} = \mathcal{P}_{0,s,a}$ by $\mathcal{D} = \mathcal{P}_{0,s}$. Now we need to find new form for κ for both s and sa rectangular assumptions.

For the second claim,

$$\left\|Q^* - \hat{Q}^{\pi^*}\right\|_\infty \leq \frac{\gamma}{1-\gamma}\max_{s,a}\left|\kappa_{\widehat{\mathcal{P}}_{s,a}}(V^*) - \kappa_{\mathcal{P}_{0,s,a}}(V^*)\right|.$$

we are using a slightly different modification:

$$\begin{aligned} Q^*(s,a) - \hat{Q}^{\pi^*}(s,a) &\overset{(a)}{=} \gamma\kappa_{\mathcal{P}_{0,s,a}}(V^*) - \gamma\kappa_{\hat{\mathcal{P}}_{s,a}}\left(\hat{V}^{\pi^*}\right) \\ &= \gamma\kappa_{\mathcal{P}_{0,s,a}}(V^*) - \gamma\kappa_{\mathcal{P}_{0,s,a}}\left(\hat{V}^{\pi^*}\right) + \gamma\kappa_{\mathcal{P}_{0,s,a}}\left(\hat{V}^{\pi^*}\right) - \gamma\kappa_{\hat{\mathcal{P}}_{s,a}}\left(\hat{V}^{\pi^*}\right) \\ &\leq \gamma\left\|Q^* - \hat{Q}^{\pi^*}\right\|_\infty + \max_{s,a}\left|\kappa_{\widehat{\mathcal{P}}_{s,a}}(V^*) - \kappa_{\mathcal{P}_{0,s,a}}(V^*)\right| \end{aligned}$$

using the same arguments as in the first inequality. Solving gives the result. □

We denote $[V]_\alpha$ as its clipped version by some non-negative vector α, namely,

$$[V]_\alpha(s) := \begin{cases} \alpha(s), & \text{if } V(s) > \alpha(s), \\ V(s), & \text{otherwise.} \end{cases} \tag{A.10}$$

Defining the gradient of $P \mapsto \|P\|$ as $\nabla\|P\|$, $\lambda > 0$, a positive scalar and ω is the generalized mean defined as the argmin in the definition of the span semi norm in Def.2.3.1, we derive two optimization lemmas.

Lemma 2.5 (Duality for the minimization problem for *sa* rectangular case.)**.** *Denoting $\widehat{P}$ the vector $\widehat{P}_{s,a}$ or P^0 for $P_{0,s,a}$,*

$$\kappa_{\hat{\mathcal{P}}_{s,a}}(\hat{V}^{\hat{\pi}}) = \max_{\mu\geq 0}\{\widehat{P}(\hat{V}^{\hat{\pi}}-\mu)-\sigma_{s,a}\mathrm{sp}_q(\hat{V}^{\hat{\pi}}-\mu)\} = \max_{\mu^{\lambda,\omega}_{\widehat{P}}\in\mathcal{M}^{\lambda,\omega}_{\widehat{P}}}\{\widehat{P}(\hat{V}^{\hat{\pi}}-\mu^{\lambda,\omega}_{\widehat{P}})-\sigma_{s,a}\mathrm{sp}_q(\hat{V}^{\hat{\pi}}-\mu^{\lambda,\omega}_{\widehat{P}})\}$$
$$= \max_{\alpha^{\lambda,\omega}_{\widehat{P}}\in\mathrm{A}^{\lambda,\omega}_{\widehat{P}}} \widehat{P}[\hat{V}^{\hat{\pi}}]_{\alpha^{\lambda,\omega}_{\widehat{P}}}-\sigma_{s,a}\mathrm{sp}_q([\hat{V}^{\hat{\pi}}]_{\alpha^{\lambda,\omega}_{\widehat{P}}}).$$

$$\kappa_{\mathcal{P}_{0,s,a}}(V^*) = \max_{\mu\geq 0}\{P^0(V^*-\mu)-\sigma_{s,a}\mathrm{sp}_q(V^*-\mu)\} = \max_{\mu^{\lambda,\omega}_{P^0}\in\mu^{\lambda,\omega}_{P^0}}\{P^0(V^*-\mu^{\lambda,\omega}_{P^0})-\sigma_{s,a}\mathrm{sp}_q(V^*-\mu^{\lambda,\omega}_{P^0})\}$$
$$= \max_{\alpha^{\lambda,\omega}_{P^0}\in\mathrm{A}^{\lambda,\omega}_{P^0}} P^0[V^*]_{\alpha^{\lambda,\omega}_{P^0}}-\sigma_{s,a}\mathrm{sp}_q([V^*]_{\alpha^{\lambda,\omega}_{P^0}}).$$

where

$$\mathrm{A}^{\lambda,\omega}_P = \{\alpha^{\lambda,\omega}_P : \alpha^{\lambda,\omega}_P(s) = \omega+\lambda|\nabla\|P\|_p(s) : \lambda>0, w>0, P\in\Delta(S), \alpha^{\lambda,\omega}_P\in\left[0,\frac{1}{1-\gamma}\right]^S\} \tag{A.11}$$

$$\mathcal{M}^{\lambda,\omega}_P = \{\mu^{\lambda,\omega}_P = V-\alpha^{\lambda,\omega}_P, \lambda,\omega\in\mathbb{R}^+, P\in\Delta(S), \mu\in\mathbb{R}^S_+, \mu^{\lambda,\omega}_P = \left[0,\frac{1}{1-\gamma}\right]^S\} \tag{A.12}$$

(A.13)

and with $[V]_\alpha := \begin{cases}\alpha(s), & \text{if } V(s)>\alpha(s)\\ V(s), & \text{otherwise.}\end{cases}$

For L_1 or TV, case , the vector $\alpha^{\lambda,\omega}_P$ reduces to a 1 dimensional scalar such as $\alpha\in[0,1/(1-\gamma)]$.

Proof. First, we will show that

$$\kappa_{\hat{\mathcal{P}}_{s,a}}(\hat{V}^{\hat{\pi}}) = \max_{\mu\geq 0}\{\widehat{P}(\hat{V}^{\hat{\pi}}-\mu)-\sigma_{s,a}\mathrm{sp}_q(\hat{V}^{\hat{\pi}}-\mu)\}$$

The second equation of this lemma is the same as the first one, replacing the center of the ball constrain $\widehat{P}_{s,a}$ by $P^0_{s,a}$ and $\hat{\pi}$ by π^*. By definition,

$$\kappa_{\hat{\mathcal{P}}_{s,a}}(\hat{V}^{\hat{\pi}}) = \min_{P\in\Delta_s,\left\|P-\widehat{P}\right\|_p\leq\sigma_{s,a}}\sum_{s'}P(s')\hat{V}^{\hat{\pi}}(s') = \widehat{P}_{s,a}\hat{V}^{\hat{\pi}} + \min_{y,\|y\|_p\leq\sigma_{s,a},1y=0,y\geq-\hat{P}}\sum_{s'}y(s')\hat{V}^{\hat{\pi}}(s')$$

where we use the change of variable $y(s') = P(s')-\hat{P}(s')$. Then writing the Lagrangian we get for $\mu\in\mathbb{R}^S_+,\gamma\in\mathbb{R}$ the Lagrangian variables:

$$\widehat{P}\hat{V}^{\hat{\pi}} + \max_{\mu\geq 0,\nu\in\mathbb{R}}\min_{y:\|y\|_p\leq\sigma_{s,a}} -\sum_{s'}\mu(s)\hat{P}(s')+\sum_{s'}(y(s')(\hat{V}^{\hat{\pi}}(s')-\mu(s')-\nu) \tag{A.14}$$

$$\overset{(a)}{=} \widehat{P}\hat{V}^{\hat{\pi}} + \max_{\mu\geq 0,\nu\in\mathbb{R}} -\sum_{s'}\mu(s')\hat{P}(s')-\sigma_{s,a}\left\|(\hat{V}^{\hat{\pi}}(s')-\mu(s')-\nu)\right\|_q \tag{A.15}$$

$$\overset{(b)}{=} \max_{\mu\geq 0}\widehat{P}(\hat{V}^{\hat{\pi}}-\mu)-\sigma_{s,a}\mathrm{sp}_q(\hat{V}^{\hat{\pi}}-\mu) \tag{A.16}$$

where (a) is true using the equality case of Holder's inequality and (b) is the definition of the span semi-norm (see Def. 2.3.1). The value that maximizes the inner maximization problem in A.15 in ν is the q-mean (see Def. 2.3.1) by definition denoted ω. Now the aim is to prove that

$$\max_{\mu\geq 0}\{\widehat{P}(\hat{V}^{\hat{\pi}}-\mu)-\sigma_{s,a}\mathrm{sp}_q(\hat{V}^{\hat{\pi}}-\mu)\}=\max_{\mu^{\lambda,\omega}_{\widehat{P}}\in\mathcal{M}^{\lambda,\omega}_{\widehat{P}}}\{\widehat{P}(\hat{V}^{\hat{\pi}}-\mu^{\lambda,\omega}_{\widehat{P}})-\sigma_{s,a}\mathrm{sp}_q(\hat{V}^{\hat{\pi}}-\mu^{\lambda,\omega}_{\widehat{P}})\}.$$

First, as the norm is differentiable (which true for L_p, $p\geq 2$), we have that the equality (a) comes from the generalized Holder's inequality for arbitrary norms Yang (1991), namely, defining $z=(\hat{V}^{\hat{\pi}}-\mu-\omega)$, it satisfies

$$z=\|z\|_q\nabla\|y\|_p \tag{A.17}$$

The quantity ν is replaced by the generalized mean for equality in (b) while (A.93) comes from Yang (1991). Using complementary slackness Karush (2013)we define $\mathcal{B}=\{s\in\mathcal{S}:\mu(s)>0\}$

$$\forall s\in\mathcal{B}:\quad y^*(s)=-\widehat{P}(s), \tag{A.18}$$

which leads to the following equality by plugging the previous (A.18) in (A.93) and defining $z^*=\hat{V}^{\hat{\pi}}-\mu^*-\omega$:

$$\forall s\in\mathcal{B},\quad z^*(s)=\|z^*\|_q\nabla\left\|\widehat{P}\right\|_p(s) \tag{A.19}$$

or

$$\forall s\in\mathcal{B},\quad \hat{V}^{\hat{\pi}}(s)-\mu^*(s)=\omega+\lambda\nabla\left\|\widehat{P}\right\|_p(s)\hat{=}\alpha^{\lambda,\omega}_{\widehat{P}} \tag{A.20}$$

by letting $\lambda=\|z^*\|_q\in\mathbb{R}^+$. Note that for $s\in\mathcal{B}$, $\nabla\|y\|_p=\nabla\|P\|_p$ only depends on $P(s)$ and not on other coordinates due to definition of L_p norm.

We can remark that $v-\mu^*$ is P dependent, but if P is known, the best μ^* is only determined by one 2 dimensional parameters $\lambda=\|v-\mu^*-\nu\|_q$ and $\omega\in\mathbb{R}^+$. Moreover, when $\widehat{P}$ is fixed, the scalar ω is a constant is fully determined by P, v and μ^*. This is why the quantity defined $\alpha^{\lambda}_{\widehat{P}}$ varies through 2 parameter λ and ω. Given this observation, we can rewrite the optimization problem as :

$$\max_{\mu\geq 0}\{\widehat{P}(\hat{V}^{\hat{\pi}}-\mu)-\sigma_{s,a}\mathrm{sp}_q(\hat{V}^{\hat{\pi}}-\mu)\}=\max_{\mu^{\lambda,\omega}_{\widehat{P}}\in\mathcal{M}^{\lambda,\omega}_{\widehat{P}}}\{\widehat{P}(\hat{V}^{\hat{\pi}}-\mu^{\lambda,\omega}_{\widehat{P}})-\sigma_{s,a}\mathrm{sp}_q(\hat{V}^{\hat{\pi}}-\mu^{\lambda,\omega}_{\widehat{P}})\} \tag{A.21}$$

$$=\max_{\alpha^{\lambda,\omega}_{\widehat{P}}\in\mathrm{A}^{\lambda,\omega}_{\widehat{P}}}\widehat{P}[\hat{V}^{\hat{\pi}}]_{\alpha^{\lambda,\omega}_{\widehat{P}}}-\sigma_{s,a}\mathrm{sp}_q([\hat{V}^{\hat{\pi}}]_{\alpha^{\lambda,\omega}_{\widehat{P}}}) \tag{A.22}$$

where we defined the maximization problem on μ not in $\mathbb{R}^S$ but at the optimal in the variational family denote $\mathcal{M}^{\lambda,\omega}_P=\{\mu^{\lambda,\omega}_P=\hat{V}^{\hat{\pi}}-\alpha^{\lambda,\omega}_P,\lambda,\omega\in\mathbb{R}^+,P\in\Delta(S),\mu\in\mathbb{R}^S_+,\mu^{\lambda,\omega}_P=\left[0,\frac{1}{1-\gamma}\right]^S\}$.

We can rewrite the optimization problem in terms of α_P with

$$[V]_{\alpha^{\lambda,\omega}_{\widehat{P}}}(s):=\begin{cases}\alpha^{\lambda,\omega}_{\widehat{P}},\text{if } V(s)\geq\alpha^{\lambda,\omega}_{\widehat{P}}\\ V(s), \qquad\qquad\qquad\text{otherwise.}\end{cases}$$

Note that for TV or L_1, this lemma holds, but the vector $\alpha^{\lambda,\omega}_{\widehat{P}}$ reduces to a positive scalar denoted α which is equal to $\left\|\hat{V}^{\hat{\pi}}-\mu^*\right\|_\infty$ according to Iyengar (2005). The thing which is of capital importance is that the second part of the equation $\mathrm{sp}_q([\hat{V}^{\hat{\pi}}]_\alpha)$ does not depend on $\widehat{P}$.

□

Lemma 2.6 (Duality for the minimization problem for s rectangular case.)**.** *Considering a projection matrix associated with a given policy π such that $P_s^\pi(s') = \sum_a \pi(a|s) P_{s,a}(s')$ and denoting $\widehat{P}^\pi \in \mathbb{R}^s$ the vector $\widehat{P}_s^\pi(.)$ or $P^{0,\pi}$ for $P_s^{0,\pi}(.)$, we have:*

$$\kappa_{\hat{\mathcal{P}}_s}(\hat{V}^{\hat{\pi}}) = \sum_a \hat{\pi}(a|s) \max_{\alpha^{\lambda,\omega}_{\hat{P}_{s,a}} \in \mathrm{A}^{\lambda,\omega}_{\hat{P}_{s,a}}} \left(\left(\hat{P}_{s,a}[\hat{V}^{\hat{\pi}}]_{\alpha^{\lambda,\omega}_{\hat{P}_{s,a}}} - \sigma_s \left\|\pi_s\right\|_q \mathrm{sp}_q([\hat{V}^{\hat{\pi}}]_{\alpha^{\lambda,\omega}_{\hat{P}_{s,a}}}) \right) \right)$$

$$\kappa_{\mathcal{P}_{0,s}}(V^*) = \sum_a \pi(a|s) \max_{\alpha^{\lambda,\omega}_{P_{0,s,a}} \in \mathrm{A}^{\lambda,\omega}_{P_{0,s,a}}} \left(\left(P_{0,s,a}[V^*]_{\alpha^{\lambda,\omega}_{P_{0,s,a}}} - \sigma_s \left\|\pi_s\right\|_q \mathrm{sp}_q([V^*]_{\alpha^{\lambda,\omega}_{P_{0,s,a}}}) \right) \right))$$

with $[V]_\alpha(s) := \begin{cases} \alpha(s), & \text{if } V(s) > \alpha \\ V(s), & \text{otherwise.} \end{cases}$

Proof. The second equation is the same replacing the center of the ball constrain $\widehat{P}_s^\pi$ by $P^{0,\pi}$ and $\hat{\pi}$ by π^*. By definition,

$$\begin{aligned} \kappa_{\hat{\mathcal{P}}_s}(\hat{V}^{\hat{\pi}})(s) &= \min_{P_s^{\hat{\pi}} \in (\Delta_s), P_s^{\hat{\pi}} \in \hat{\mathcal{P}}_s} P_s^{\hat{\pi}} \hat{V}^{\hat{\pi}}(s) \\ &\overset{(a)}{=} \sum_a \hat{\pi}(a|s) \hat{P}_{s,a} \hat{V}^{\hat{\pi}} + \min_{\|\sigma_{s,a}\|_p \le \sigma_s} \sum_a \hat{\pi}(a|s) \min_{y, \|y\|_p \le \sigma_{s,a}, 1y=0, y \ge -\hat{P}} \sum_{s'} y(s') \hat{V}^{\hat{\pi}} \end{aligned}$$

where we use the change of variable $y(s') = P_{s,a}(s') - \hat{P}_{s,a}(s')$ in (a). Then we case use the previous lemma for *sa* rectangular assumption, Lemma 2.3.3. Then,

$$\begin{aligned} &\min_{\|\sigma_{s,a}\|_p \le \sigma_s} \sum_a \hat{\pi}(a|s) \min_{y, \|y\|_p \le \sigma_{s,a}, 1y=0, y \ge -\hat{P}_{s,a}} \sum_{s'} y(s') \hat{V}^{\hat{\pi}} \\ &= \min_{\|\sigma_{s,a}\|_p \le \sigma_s} \sum_a \hat{\pi}(a|s) \max_{\mu \ge 0} \Big(- \widehat{P}_{s,a}\mu - \sigma_{s,a} \mathrm{sp}_q(\hat{V}^{\hat{\pi}} - \mu) \Big) \\ &= \sum_a \max_{\mu \ge 0} \left(\hat{\pi}(a|s)(-\widehat{P}_{s,a}\mu) - \max_{\|\sigma_{s,a}\|_p \le \sigma_s} \sum_a \hat{\pi}(a|s) \sigma_{s,a} \mathrm{sp}_q(\hat{V}^{\hat{\pi}} - \mu) \right) \\ &= \sum_a \max_{\mu \ge 0} \left(\hat{\pi}(a|s)(-\widehat{P}_{s,a}\mu) - \sigma_s \left\|\pi_s\right\|_q \mathrm{sp}_q(\hat{V}^{\hat{\pi}} - \mu) \right) \end{aligned}$$

we can exchange the min and the max as we get concave-convex problems in $\sigma_{s,a}$ and μ , ((v. Neumann 1928)) in the second line and using Holder's inequality in the last line. Finally, we obtain:

$$\begin{aligned} \kappa_{\hat{\mathcal{P}}_s}(\hat{V}^{\hat{\pi}}) &= \sum_a \max_{\mu \ge 0} \left(\hat{\pi}(a|s)(\hat{P}_{s,a}(\hat{V}^{\hat{\pi}} - \mu) - \sigma_s \left\|\pi_s\right\|_q \mathrm{sp}_q(\hat{V}^{\hat{\pi}} - \mu)) \right) \\ &\overset{(a)}{=} \sum_a \hat{\pi}(a|s) \max_{\alpha^{\lambda,\omega}_{\hat{P}_{s,a}} \in \mathrm{A}^{\lambda,\omega}_{\hat{P}_{s,a}}} \left(\left(\hat{P}_{s,a}[\hat{V}^{\hat{\pi}}]_{\alpha^{\lambda,\omega}_{\hat{P}_{s,a}}} - \sigma_s \left\|\pi_s\right\|_q \mathrm{sp}_q([\hat{V}^{\hat{\pi}}]_{\alpha^{\lambda,\omega}_{\hat{P}_{s,a}}}) \right) \right)) \end{aligned}$$

where in (a) we use Lemma 2.3.3. Second claim is the same replacing $\hat{V}^{\hat{\pi}}$ by V^*, $\hat{\pi}$ by π^* and $\hat{P}$ by P^0. Then we derive a new decomposition of the difference the two minimum.

□

Lemma 2.7. *For s and sa rectangular assumptions,*

$$\left|\kappa_{\hat{\mathcal{P}}_{s,a}}(\hat{V}^{\hat{\pi}}) - \kappa_{\mathcal{P}_{0,s,a}}(\hat{V}^{\hat{\pi}})\right| \leq \max\Big\{ \underbrace{\max_{s,a} \left| \max_{\mu \in \mu^{\lambda,\omega}_{P^0_{s,a}}} \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)(\hat{V}^{\hat{\pi}} - \mu^{\lambda,\omega}_{P^0_{s,a}}) \right|}_{=:g_{s,a}(\alpha^{\lambda,\omega}_P, \hat{V}^{\hat{\pi}})}, \tag{A.23}$$

$$\underbrace{\max_{s,a} \left| \max_{\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{\hat{P}^0_{s,a}}} \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)(\hat{V}^{\hat{\pi}} - \mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}) \right|}_{=:g_{s,a}(\alpha^{\lambda,\omega}_{\hat{P}}, \hat{V}^{\hat{\pi}})} \Big\} \tag{A.24}$$

$$\left|\kappa_{\hat{\mathcal{P}}_s}(V^*) - \kappa_{\mathcal{P}_{0,s}}(V^*)\right| \leq \max\Big\{ \underbrace{\max_{s,a} \left| \max_{\mu \in \mu^{\lambda,\omega}_{P^0_{s,a}}} \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)(V^* - \mu^{\lambda,\omega}_{P^0_{s,a}}) \right|}_{=:g_{s,a}(\alpha^{\lambda,\omega}_P, V^*)}, \tag{A.25}$$

$$\underbrace{\max_{s,a} \left| \max_{\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{\hat{P}^0_{s,a}}} \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)(V^* - \mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}) \right|}_{=:g_{s,a}(\alpha^{\lambda,\omega}_{\hat{P}}, V^*)} \Big\} \tag{A.26}$$

Proof.

$$\left|\kappa_{\hat{\mathcal{P}}_{s,a}}(V^*) - \kappa_{\mathcal{P}_{0,s,a}}(V^*)\right| \tag{A.27}$$

$$= \Big| \max_{\mu^{\lambda,\omega}_{P^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{P^0_{s,a}}} \left\{ P^0_{s,a}(V^* - \mu) - \sigma_{s,a}\left(\mathrm{sp}((V^* - \mu))_*\right) \right\}$$

$$- \max_{\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{\hat{P}^0_{s,a}}} \left\{ \widehat{P}^0_{s,a}(V^* - \mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}) - \sigma_{s,a}\left(\mathrm{sp}((V^* - \mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}))_*\right) \right\} \Big|$$

$$\leq \max\Big\{ \Big| \max_{\mu^{\lambda,\omega}_{P^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{P^0_{s,a}}} \left\{ P^0_{s,a}(V^* - \mu^{\lambda,\omega}_{P^0_{s,a}}) - \sigma_{s,a}\left(\mathrm{sp}((V^* - \mu^{\lambda,\omega}_{P^0_{s,a}}))_*\right) \right\}$$

$$- \max_{\mu^{\lambda,\omega}_{P^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{P^0_{s,a}}} \left\{ \widehat{P}^0_{s,a}(V^* - \mu^{\lambda,\omega}_{P^0_{s,a}}) - \sigma_{s,a}\left(\mathrm{sp}((V^* - \mu^{\lambda,\omega}_{P^0_{s,a}}))_*\right) \right\} \Big|; \tag{A.28}$$

$$\Big| \max_{\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{\hat{P}^0_{s,a}}} \left\{ \widehat{P}^0_{s,a}(V^* - \mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}) - \sigma_{s,a}\left(\mathrm{sp}((V^* - \mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}))_*\right) \right\} \tag{A.29}$$

$$- \max_{\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{\hat{P}^0_{s,a}}} \left\{ P^0_{s,a}(V^* - \mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}) - \sigma_{s,a}\left(\mathrm{sp}((V^* - \mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}))_*\right) \right\} \quad \Big| \Big\}$$

$$\leq \max\Big\{ \underbrace{\left| \max_{\mu \in \mu^{\lambda,\omega}_{P^0_{s,a}}} \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)(V^* - \mu^{\lambda,\omega}_{P^0_{s,a}}) \right|}_{=:g_{s,a}(\alpha^{\lambda,\omega}_P, V^*)}, \underbrace{\left| \max_{\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{\hat{P}^0_{s,a}}} \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)(V^* - \mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}) \right|}_{=:g_{s,a}(\alpha^{\lambda,\omega}_{\hat{P}}, V^*)} \Big\} \tag{A.30}$$

where in the first equality we use Lemma 2.5. The final inequality is a consequence of the 1-Lipschitzness of the max operator. Taking the supremum over s, a gives the result. Replacing V^* by $\hat{V}^{\hat{\pi}}$ gives the other inequality. The result for s rectangular are the same as

$$\sum_a \pi(a|s) \max\Big\{ \underbrace{\Big| \max_{\mu \in \mu^{\lambda,\omega}_{P^0_{s,a}}} \Big(P^0_{s,a} - \widehat{P}^0_{s,a}\Big)(V^* - \mu^{\lambda,\omega}_{P^0_{s,a}}) \Big|}_{=: g_{s,a}(\alpha^{\lambda,\omega}_P, V^*)}, \tag{A.31}$$

$$\underbrace{\Big| \max_{\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{\hat{P}^0_{s,a}}} \Big(P^0_{s,a} - \widehat{P}^0_{s,a}\Big)(V^* - \mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}) \Big|}_{=: g_{s,a}(\alpha^{\lambda,\omega}_{\hat{P}}, V^*)} \Big\} \tag{A.32}$$

$$\leq \max\Big\{ \max_{s,a} \underbrace{\Big| \max_{\mu \in \mu^{\lambda,\omega}_{P^0_{s,a}}} \Big(P^0_{s,a} - \widehat{P}^0_{s,a}\Big)(V^* - \mu^{\lambda,\omega}_{P^0_{s,a}}) \Big|}_{=: g_{s,a}(\alpha^{\lambda,\omega}_P, V^*)}, \tag{A.33}$$

$$\max_{s,a} \underbrace{\Big| \max_{\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{\hat{P}^0_{s,a}}} \Big(P^0_{s,a} - \widehat{P}^0_{s,a}\Big)(V^* - \mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}) \Big|}_{=: g_{s,a}(\alpha^{\lambda,\omega}_{\hat{P}}, V^*)} \Big\} \tag{A.34}$$

Note that at this point, quantities for s and sa rectangular is the same as the part with span semi norms cancelled. Now, note that the main problem is that we can not apply classical Hoeffding's inequality as $\widehat{P}$ is dependent of data as $\hat{V}^{\hat{\pi}}$. We need to decouple $\hat{V}^{\hat{\pi}}$ using s absorbing MDPS as in Agarwal et al. (2020) but using Hoeffding arguments. First, we will use a concentration for V^*.

□

Lemma 2.8. *For sa and s-rectangular, with probability* $1-\delta$*, it holds:*

$$\Big|\kappa_{\hat{\mathcal{P}}_{s,a}}(V^*) - \kappa_{\mathcal{P}_{0,s,a}}(V^*)\Big| \leq 2\sqrt{\frac{L}{2N(1-\gamma)^2}} + \frac{2L|S|^{1/q}\|1_S\|_q(p-1)}{N(1-\gamma)}$$

with $L = \log\Big(18\|1\|_q SAN/\delta\Big)$

Proof. First, we can use previous Lemma 2.7

$$\Big|\kappa_{\hat{\mathcal{P}}_{s,a}}(V^*) - \kappa_{\mathcal{P}_{0,s,a}}(V^*)\Big| \tag{A.35}$$

$$\leq \max\Big\{ \underbrace{\Big| \max_{\mu \in \mu^{\lambda,\omega}_{P^0_{s,a}}} \Big(P^0_{s,a} - \widehat{P}^0_{s,a}\Big)(V^* - \mu^{\lambda,\omega}_{P^0_{s,a}}) \Big|}_{=: g_{s,a}(\alpha^{\lambda,\omega}_P, V^*)}, \underbrace{\Big| \max_{\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{\hat{P}^0_{s,a}}} \Big(P^0_{s,a} - \widehat{P}^0_{s,a}\Big)(V^* - \mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}) \Big|}_{=: g_{s,a}(\alpha^{\lambda,\omega}_{\hat{P}}, V^*)} \Big\} \tag{A.36}$$

First, we control $g_{s,a}(\alpha^{\lambda,\omega}_P, V^*)$. To do so, we use for a fixed $\alpha^{\lambda,\omega}_P$ and any vector V^* that is independent with $\widehat{P}^0$, the Hoeffding's inequality, one has with probability at least $1-\delta$ with *sa*-rectangular notations,

$$g_{s,a}(\alpha_P^{\lambda,\omega}, V^*) = \left| \left(P_{s,a}^0 - \widehat{P}_{s,a}^0 \right) [V^*]_{\alpha_P^{\lambda,\omega}} \right| \leq \sqrt{\frac{\log\left(\frac{2}{\delta}\right)}{(1-\gamma)^2 2N}} \tag{A.37}$$

Once pointwise concentration derived, we will use uniform concentration to yield this lemma. First, union bound, is obtained noticing that $g_{s,a}(\alpha_P^{\lambda,\omega}, V^*)$ is 1-Lipschitz w.r.t. λ and ω as it is linear in λ and ω. Moreover, $\lambda^* = \|V^* - \mu^* - \omega\|_q$ obeying $\lambda^* \leq \frac{\|1\|_q}{1-\gamma}$. The quantity $\omega \in [0, 1/(1-\gamma)]$ as it is always smaller that V^* by definition. We construct then a 2-dimensional a ε_1-net N_{ε_1} over $\lambda^* \in [0, \frac{\|1\|_q}{1-\gamma}]$ and $\omega \in [0, 1/(1-\gamma)]$ whose size satisfies $|N_{\varepsilon_1}| \leq \left(\frac{3\|1\|_q}{\varepsilon_1(1-\gamma)}\right)^2$ (Vershynin 2018). Using union bound and (A.194), it holds with probability at least $1 - \frac{\delta}{SA}$ that for all $\lambda \in N_{\varepsilon_1}$,

$$g_{s,a}(\alpha_P^{\lambda}, V^*) \leq \sqrt{\frac{2\log\left(\frac{SA|N_{\varepsilon_1}|}{\delta}\right)}{2N(1-\gamma)^2}}. \tag{A.38}$$

Using the previous equation and also (A.193), it results in using notation $\log\left(\frac{18SAN}{\delta}\right) = L$,

$$\begin{aligned}
g_{s,a}(\alpha_P^{\lambda}, V^*) &\overset{(a)}{\leq} \sup_{\alpha_P^{\lambda} \in N_{\varepsilon_1}} \left| \left(P_{s,a}^0 - \widehat{P}_{s,a}^0 \right) [V^*]_{\alpha_P^{\lambda}} \right| + \varepsilon_1 \\
&\overset{(b)}{\leq} \sqrt{\frac{\log\left(\frac{SA|N_{\varepsilon_1}|}{\delta}\right)}{2(1-\gamma)^2 N}} + \varepsilon_1 \\
&\overset{(c)}{\leq} \sqrt{\frac{\log\left(\frac{2SA|N_{\varepsilon_1}|}{\delta}\right)}{2N(1-\gamma)^2}} + \frac{\log\left(\frac{2SA|N_{\varepsilon_1}|}{\delta}\right)}{3N(1-\gamma)} \\
&\overset{(d)}{\leq} \sqrt{\frac{L}{2N(1-\gamma)2}} + \frac{L}{3N(1-\gamma)} \\
&\leq 2\sqrt{\frac{L}{2(1-\gamma)^2 N}}
\end{aligned} \tag{A.39, A.40}$$

where (a) is because the optimal α^* falls into the ε_1-ball centered around some point inside N_{ε_1} and $g_{s,a}(\alpha_P^{\lambda}, V^*)$ is 1-Lipschitz with regard to λ and ω, (b) is due to Eq. (A.38), (c) arises from taking $\varepsilon_1 = \frac{\log\left(\frac{2SA|N_{\varepsilon_1}|}{\delta}\right)}{3N(1-\gamma)}$, (d) is verified by $|N_{\varepsilon_1}| \leq \left(\frac{3\|1\|_q}{\varepsilon_1(1-\gamma)}\right)^2 \leq 9N\|1\|_q$ and that variance of a ceiling function of a vector is smaller than the variance of non-ceiling vector.

For L_p with $p \geq 2$, contrary to the previous term, the second term $g_{s,a}(\alpha_{\widehat{P}}^{\lambda}, V)$ is more difficult as we need concentration, but there is an extra dependency in the data thought the parameter $\alpha_{\widehat{P}}^{\lambda}$. Note that this term does not exist as α is a constant for TV. We need to decouple this problem using absorbing MDPs. Then it leads to

$$g_{s,a}(\alpha^{\lambda,\omega}_{\hat{P}}, V^*) \tag{A.41}$$

$$= |\max_{\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{\hat{P}^0_{s,a}}} \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)(V^* - \mu^{\lambda,\omega}_{\hat{P}^0_{s,a}})| \tag{A.42}$$

$$= |\max_{\mu \in \mathcal{M}^{\lambda,\omega}_{\hat{P}^0_{s,a}}} \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)(V^* - \mu^{\lambda,\omega}_{P^0_{s,a}}) + \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)(\mu^{\lambda,\omega}_{P^0_{s,a}} - \mu^{\lambda,\omega}_{\hat{P}^0_{s,a}})| \tag{A.43}$$

$$\leq |\max_{\mu^{\lambda,\omega}_{P^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{P^0_{s,a}}} \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)(V^* - \mu^{\lambda,\omega}_{P^0_{s,a}}) \tag{A.44}$$

$$+ \max_{\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{\hat{P}^0_{s,a}}} \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)(\mu^{\lambda,\omega}_{P^0_{s,a}} - \mu^{\lambda,\omega}_{\hat{P}^0_{s,a}})| \tag{A.45}$$

In the first equality, we add the term $\mu^{\lambda,\omega}_{P^0_{s,a}}$ to retrieve the previous concentration problem, fixing $P^0_{s,a}$ and optimizing λ, ω. In the second, we extend the max using triangular inequality. The first term in the last equality is exactly the term we have controlled previously, while the second one needs more attention. We decouple the dependency of the data, and then controlling the difference between the μ. Then using the characterization of the optimal μ from equation (A.96):

$$\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)(\mu^{\lambda,\omega}_{P^0_{s,a}} - \mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}) = \sum_{s'} \lambda \left(P^0_{s,a}(s') - \widehat{P}^0_{s,a}(s')\right)(\nabla \left\|P^0_{s,a}\right\|_p(s') - \nabla \left\|\hat{P}^0_{s,a}(s')\right\|_p)$$

As the norm is C^2 for $p \geq 2$, using Mean value theorem, we know that

$$\left\|(\nabla \left\|P^0_{s,a}\right\|_p - \nabla \left\|\hat{P}^0_{s,a}\right\|_p)\right\|_2 \leq \sup_{x \in \Delta(S)} \left\|\nabla^2 \|x\|_p\right\|_2 \left\|(P^0_{s,a} - \hat{P}^0_{s,a})\right\|_2.$$

For $L_p = \|x\|_p$ norms, $p \geq 2$, we have simple taking derivative twice:

$$\nabla^2 \|x\|_p = \frac{p-1}{L_p}\left(\mathcal{A}^{p-2} - g_p g_p^T\right)$$

with

$$\mathcal{A} = \text{Diag}\left(\frac{\text{abs}(x)}{L_p}\right) \quad g_p = \mathcal{A}^{p-2}\left(\frac{x}{L_p}\right).$$

and L_p the norm, where Diag is the diagonal matrix. However, as $x \leq L_p$, $\mathcal{A} \leq I$, we get

$$H \leq \frac{p-1}{\|x\|_p} \leq (p-1)|S|^{1/q} \tag{A.46}$$

where the $1/L_p$ is minimized for the uniform distribution. Then using Cauchy-Swartz inequality, it holds

$$\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)(\mu^{\lambda,\omega}_{P^0_{s,a}} - \mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}) \leq (p-1)\lambda|S|^{1/q}\left\|\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)\right\|_2^2. \tag{A.47}$$

Then the question is how to bound the quantity $\left\|\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)\right\|_2^2$. To do so, we will use Mac Diarmid inequality.

Definition 2.1. *Bounded difference property*

A function $f : \mathcal{X}_1 \times \dots \mathcal{X}_n \to \mathbb{R}$ satisfies the bounded difference property if for each $i = 1, \dots, n$ the change of coordinate from s_i to s_i' may change the value of the function at most on c_i

$$\forall i \in [n] : \sup_{x_i' \in \mathcal{X}_i} \left| f\left(x_1, \dots, x_i, \dots, x_n\right) - f\left(x_1, \dots, x_i', \dots, x_n\right) \right| \leq c_i$$

In our case, we consider $f\left(X_1, \dots, X_n\right) = \left\| \sum_{k=1}^n X_k \right\|_2$. Then we can notice that by triangle inequality for any $x_1, \dots, x_n$ and x_k' with $X_{i,s'} = P^i_{0,s,a}(s') - P^0_{s,a}(s')$ (index i holds for index of sample generated from the generative model) that

$$\begin{aligned} & f\left(x_1, \dots, x_k, \dots, x_n\right) = \left\| x_1 + \dots + x_n \right\|_2 \leq \left\| x_1 + \dots + x_n - x_k + x_k' \right\|_2 + \left\| x_k - x_k' \right\|_2 \\ & \leq f\left(x_1, \dots, x_k', \dots, x_n\right) + 2 \end{aligned}$$

Theorem 2.9. *(McDiarmid's inequality). McDiarmid et al. (1989) Let $f : \mathcal{X}_1 \times \dots \mathcal{X}_n \to \mathbb{R}$ be a function satisfying the bounded difference property with bounds $c_1, \dots, c_n$. Consider independent random variables $X_1, \dots, X_n, X_i \in \mathcal{X}_i$ for all i. Then for any $t > 0$*

$$\mathbb{P}\left[f\left(X_1, \dots, X_n\right) - \mathbb{E}\left[f\left(X_1, \dots, X_n\right) \right] \geq t \right] \leq \exp\left(- \frac{2t^2}{\sum_{i=1}^n c_i^2} \right)$$

Using McDiarmid's inequality and union bound, we can bound the term as here

$$\left(\left\| \left(P^0_{s,a} - \widehat{P}^0_{s,a} \right) \right\|_2 - \mathbb{E}[\left\| \left(P^0_{s,a} - \widehat{P}^0_{s,a} \right) \right\|_2] \right)^2 \leq \frac{2N \log(|S||A|/\delta))}{N^2}$$

with probability $1 - \delta/(|S||A|)$. Moreover, the additional term can be bounded as follows:

$$\mathbb{E}[\left\| \left(P^0_{s,a} - \widehat{P}^0_{s,a} \right) \right\|_2^2] = \mathbb{E}[\sum_{s'} (P^0_{s,a}(s') - P^0_{s,a}(s'))^2 = \mathbb{E}[\sum_{s'} (\frac{1}{N} \sum_i^N X_{i,s'})^2]$$

with $X_{i,s'} = P^i_{0,s,a}(s') - P^0_{s,a}(s')$ is one sample sampled from the generative model. Then

$$\begin{aligned} \mathbb{E}[\left\| \left(P^0_{s,a} - \widehat{P}^0_{s,a} \right) \right\|_2^2] = \frac{1}{N^2} \sum_{s'} \mathsf{Var}(\sum_i^N X_{i,s}) &\overset{a}{=} \frac{1}{N^2} \sum_i^N \sum_{s'} \mathsf{Var}(X_{i,s}) \\ &= \frac{1}{N^2} \sum_i^N \mathbb{E}(\sum_{s'} X_{i,s}^2) \leq \frac{4}{N} \end{aligned}$$

where (a) the last equality comes from the independence of the random variables and where the last inequality comes from the fact the maximum of two elements in the simplex is bounded by 2. Finally, regrouping all the terms, we obtain with probability $1 - \delta/(|S||A|)$:

$$\begin{aligned} & \left\| \left(P^0_{s,a} - \widehat{P}^0_{s,a} \right) \right\|_2^2 = \left(\left\| \left(P^0_{s,a} - \widehat{P}^0_{s,a} \right) \right\|_2 - \mathbb{E}[\left\| \left(P^0_{s,a} - \widehat{P}^0_{s,a} \right) \right\|_2] \right)^2 + \mathbb{E}[\left\| \left(P^0_{s,a} - \widehat{P}^0_{s,a} \right) \right\|_2^2] \\ & + 2\mathbb{E}\left[\left\| \left(P^0_{s,a} - \widehat{P}^0_{s,a} \right) \right\|_2 \right]^2 \left(\left\| \left(P^0_{s,a} - \widehat{P}^0_{s,a} \right) \right\|_2 - \mathbb{E}[\left\| \left(P^0_{s,a} - \widehat{P}^0_{s,a} \right) \right\|_2] \right) \\ & \leq \frac{2N \log(|S||A|/(\delta)))}{N^2} + \frac{4}{N} + \frac{\sqrt{\frac{4}{N}} \sqrt{2N \log(|S||A|/(\delta)))}}{N} \\ & \leq \frac{10 \log(|S||A|/(\delta))}{N} = \frac{L'}{N} \end{aligned}$$

with $L' = 10\log(|S||A|/(\delta))$. Finally, plugging the previous equation in (A.204):

$$\max_{\mu\in\mu^{\lambda}_{\widehat{P}^0_{s,a}}} \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)(\mu^{\lambda}_{P^0_{s,a}} - \mu)| \leq \max_{\lambda} \left\|\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)\right\|_2^2 S^{1/q}(p-1)\lambda.$$

This term can be easily controlled by taking the supremum over λ which is a 1 dimensional parameter. Then we can bound $\lambda \in [0, H\|1_S\|_q]$. Indeed,

$$\lambda^* = \|V^* - \mu^* - \omega\|_q \leq \|V^*\|_q \leq H\|1_S\|_q.$$

Finally, we obtain:

$$\max_{\lambda} \left\|\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)\right\|_2^2 S^{1/q}\lambda \leq \frac{L'|S|^{1/q}\|1_S\|_q(p-1)}{N(1-\gamma)}.$$

Regrouping all terms:

$$\begin{aligned} g_{s,a}(\alpha^{\lambda}_{\hat{P}}, V^*) \leq |\max_{\mu^{\lambda}_{P^0_{s,a}} \in \mathcal{M}^{\lambda}_{P^0_{s,a}}} & \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)(V^* - \mu^{\lambda}_{P^0_{s,a}}) \\ + \max_{\mu^{\lambda}_{\widehat{P}^0_{s,a}} \in \mathcal{M}^{\lambda}_{\widehat{P}^0_{s,a}}} & \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)(\mu^{\lambda}_{P^0_{s,a}} - \mu^{\lambda}_{\widehat{P}^0_{s,a}})| \qquad \text{(A.48)} \\ \leq 2\sqrt{\frac{L}{2N(1-\gamma)^2}} + \frac{L'|S|^{1/q}\|1_S\|_q(p-1)}{N(1-\gamma)} & \leq 2\sqrt{\frac{L}{2N(1-\gamma)^2}} + \frac{2L|S|^{1/q}\|1_S\|_q(p-1)}{N(1-\gamma)} \qquad \text{(A.49)} \\ & \qquad \text{(A.50)} \end{aligned}$$

For the specific case of TV which is not C^2 smooth, this lemma still holds as in (A.193), we only need to control one term without the dependency on data in the supremum as α^{λ}_P reduces to a scalar α which does not depend on P. Then extra decomposition using smoothness of the norm is not needed, as the only remaining term in the max in (A.193) is the left hand side term.

□

Lemma 2.10 (s-absorbing MDPs for Hoeffding's concentration Inequalities)**.**

As in Agarwal paper Agarwal et al. (2020), we define for a state s and a scalar u, the MDP called $M_{s,u}$ such that: $M_{s,u}$ is identical to M except that state s is absorbing in $M_{s,u}$, i.e. $P_{M_{s,u}}(s \mid s, a) = 1$ for all a, and the reward at state s in $M_{s,u}$ is $(1-\gamma)u$. The remainder of the transition model and reward function are identical to those in M. In the following, we will use $V^{\pi}_{s,u}$ to denote the value function $V^{\pi}_{M_{s,u}}$ and correspondingly for Q and reward and transition functions to avoid notational clutter. Then, we have that for all policies π :

$$V^{\pi}_{s,u}(s) = u$$

because s is absorbing with reward $(1-\gamma)u$. For some state s, we will only consider the MDP $M_{s,u}$ for u in a finite set U_s with

$$U_s \subset [V^{\star}(s) - \Delta_{\delta,N} V^{\star}(s) + \Delta_{\delta,N}]\,.$$

with $\Delta_{\delta,N} := \frac{\gamma}{(1-\gamma)^2}\left(2\sqrt{\frac{L}{2N}} + \frac{2L|S|^{1/q}\|1_S\|_q(p-1)}{N}\right)$ The set U_s consists of evenly spaced elements in this interval, where we set the size of $|U_s|$ appropriately later on. As before, we let $\widehat{M}_{s,u}$

denote the MDP that uses the empirical model $\widehat{P}$ instead of P, at all non-absorbing states and abbreviate the value functions in $\widehat{M}_{s,u}$ as $\widehat{V}^{\pi}_{s,u}$. Then we have for a fix a state s, action a, a finite set U_s, and $\delta \geq 0$, that for all $u \in U_s$: with probability greater than $1-\delta$, it holds :

$$|(\widehat{P}_{s,a} - P^0_{s,a})[V^{\hat{\pi}}_u]_{\alpha_P^{\lambda,\omega}}| \leq \tag{A.51}$$

$$\frac{1}{(1-\gamma)}\Big(2\sqrt{\frac{\log\Big(\frac{18SAN|U_s|\|1\|_q}{\delta}\Big)}{2N}} + \frac{2\log\Big(\frac{18SAN|U_s|\|1\|_q}{\delta}\Big)|S|^{1/q}\|1_S\|_q(p-1)}{N}\Big) \tag{A.52}$$

This is exactly 2.8 in equation (A.193) to the finite set U_s as now $V^{\hat{\pi}}_u$ and $\widehat{P}_{s,a}$ are now independent.

Lemma 2.11 (Agarwal et al. (2020), Lemma 7)**.** *Let $u^* = V^{\star}_M(s)$ and $u^{\pi} = V^{\pi}_M(s)$. We have*

$$V^{\star}_M = V^{\star}_{s,u^{\star}}, \quad \text{and for all policies } \pi, \quad V^{\pi}_M = V^{\pi}_{M^{\pi}_{s,u^{\pi}}}$$

Proof can be found in Agarwal et al. (2020), Lemma 7.

Lemma 2.12. *For any u, u', s and policy π:*

$$\left\|Q^{\pi}_{s,u} - Q^{\pi}_{s,u'}\right\|_{\infty} \leq |u - u'|$$

Proof. To obtain the result in our robust MDP setting, we need a similar stability property like in Lemma 8 of Agarwal et al. (2020), but for the robust value functions. It turns out that this a direct consequence of the property for classical MDP. Agarwal in Agarwal et al. (2020) show equation A.53 for classical MPDs, then we have for RMDPs:

$$|Q^{\pi}_{M_{s,u}}(s,a) - Q^{\pi}_{M_{s,u'}}(s,a)| \leq \frac{1}{1-\gamma}|u-u'| \tag{A.53}$$

$$\Rightarrow |\inf_M Q^{\pi}_{M_{s,u}}(s,a) - \inf_M Q^{\pi}_{M_{s,u}}(s,a)| \leq \frac{1}{1-\gamma}|u-u'| \tag{A.54}$$

$$\Rightarrow |\sup_{\pi}\inf_M Q^{\pi}_{M_{s,u}}(s,a) - \sup_{\pi}\inf_M Q^{\pi}_{M_{s,u}}(s,a)| \leq \frac{1}{1-\gamma}|u-u'|. \tag{A.55}$$

which concludes the proof for RMDPs. □

Lemma 2.13 (Hoeffding's Concentration for dependent variables)**.** *Removing s, a notations for kernels,*

$$\left|\left(P^0 - \widehat{P}\right)\cdot[\widehat{V}^{\star}]_{\alpha_P^{\lambda,\omega}}\right| \leq \frac{1}{(1-\gamma)}\Big(2\sqrt{\frac{\log\Big(\frac{18SAN|U_s|\|1\|_q}{\delta}\Big)}{2N}} + \frac{2\log\Big(\frac{18SAN|U_s|\|1\|_q}{\delta}\Big)|S|^{1/q}\|1_S\|_q(p-1)}{N}\Big) \tag{A.56}$$

$$+ 2\min_{u\in U_s}\left|\widehat{V}^{\star}(s) - u\right| \tag{A.57}$$

Proof.

$$\left|\left(P^0-\widehat{P}\right)\cdot[\widehat{V}^\star]_{\alpha_P^{\lambda,\omega}}\right| \tag{A.58}$$

$$=\left|\left(P^0-\widehat{P}\right)\cdot\left([\widehat{V}^\star]_{\alpha_P^{\lambda,\omega}}-[V^\star_{s,u}]_{\alpha_P^{\lambda,\omega}}+[V^\star_{s,u}]_{\alpha_P^{\lambda,\omega}}\right)\right| \tag{A.59}$$

$$\leq\left|\left(P^0-\widehat{P}\right)\cdot\left([\widehat{V}^\star]_{\alpha_P^{\lambda,\omega}}-[V^\star_{s,u}]_{\alpha_P^{\lambda,\omega}}\right)\right|+\left|\left(P^0-\widehat{P}\right)\cdot\left([V^\star_{s,u}]_{\alpha_P^{\lambda,\omega}}\right)\right| \tag{A.60}$$

$$\overset{(a)}{\leq}\frac{1}{(1-\gamma)}\Big(2\sqrt{\frac{\log\left(\frac{18SAN|U_s|\|1\|_q}{\delta}\right)}{2N}}+\frac{2\log\left(\frac{18SAN|U_s|\|1\|_q}{\delta}\right)|S|^{1/q}\|1_S\|_q(p-1)}{N}\Big) \tag{A.61}$$

$$+2\left\|\widehat{V}^\star-V^\star_{s,u}\right\|_\infty \tag{A.62}$$

$$\overset{(b)}{\leq}\frac{1}{(1-\gamma)}\Big(2\sqrt{\frac{\log\left(\frac{18SAN|U_s|\|1\|_q}{\delta}\right)}{2N}}+\frac{2\log\left(\frac{18SAN|U_s|\|1\|_q}{\delta}\right)|S|^{1/q}\|1_S\|_q(p-1)}{N}\Big) \tag{A.63}$$

$$+2\left|\widehat{V}^\star(s)-u\right| \tag{A.64}$$

$$\tag{A.65}$$

where (a) is A.51 or Hoeffding's inequality for s-absorbing MDPs. By Lemmas 2.11 and 2.12,

$$\left\|[\widehat{V}^\star]_{\alpha_P^{\lambda,\omega}}-[V^\star_{s,u}]_{\alpha_P^{\lambda,\omega}}\right\|_\infty\leq\left\|[\widehat{V}^\star-V^\star_{s,u}]_{\alpha_P^{\lambda,\omega}}\right\|_\infty\leq\left\|\widehat{V}^\star-V^\star_{s,u}\right\|_\infty=\left\|\widehat{V}^\star_{s,\widehat{V}^\star(s)}-V^\star_{s,u}\right\|_\infty\leq\left|\widehat{V}^\star(s)-u\right|.$$

which is point (b). The last min operator in the result comes from the fact that the previous equation holds for all $u\in U_s$, we take the best possible choice, which completes the proof of the first claim. □

Lemma 2.14 (Crude bound for Robust MDPs)**.** *This lemma is needed for next Lemma 2.15 but the proof differs from the classical MDP setting. For s and sa rectangular assumptions,*

$$\left\|Q^*-\hat{Q}^{\pi^*}\right\|_\infty\leq\Delta_{\delta,N}\ \text{and}\ \left\|Q^*-\hat{Q}^*\right\|_\infty\leq\Delta_{\delta,N} \tag{A.66}$$

$$\text{with}\quad\Delta_{\delta,N}=\frac{\gamma}{(1-\gamma)^2}\Big(2\sqrt{\frac{L}{2N}}+\frac{2L|S|^{1/q}\|1_S\|_q(p-1)}{N}\Big) \tag{A.67}$$

Proof. For the first claim :

$$\begin{aligned}\left\|Q^\pi-\hat{Q}^\pi\right\|_\infty&=\max_{s,a}\left|\gamma\left(\kappa_{\mathcal{P}_{0,s,a}}(V^\pi)-\kappa_{\hat{\mathcal{P}}_{s,a}}(V^\pi)\right)+\gamma\left(\kappa_{\hat{\mathcal{P}}_{s,a}}(V^\pi)-\kappa_{\hat{\mathcal{P}}_{s,a}}\left(\hat{V}^\pi\right)\right)\right|\\&\overset{(b)}{\leq}\max_{s,a}\left|\gamma\left(\kappa_{\mathcal{P}_{0,s,a}}(V^\pi)-\kappa_{\hat{\mathcal{P}}_{s,a}}(V^\pi)\right)\right|+\gamma\left\|V^\pi-\hat{V}^\pi\right\|_\infty\\&\overset{(b)}{\leq}\gamma\max_{s,a}\left|\kappa_{\hat{\mathcal{P}}_{s,a}}(V^\pi)-\kappa_{\mathcal{P}_{0,s,a}}(V^\pi)\right|+\gamma\left\|Q^\pi-\hat{Q}^\pi\right\|_\infty.\end{aligned}$$

□

where we use contraction of κ, lemma 2.3 in (a) and $\left\|Q^\pi-\hat{Q}^\pi\right\|_\infty\leq\left\|V^\pi-\hat{V}^\pi\right\|_\infty$ in (c) for any π. Solving we get :

$$\left\|Q^\pi-\hat{Q}^\pi\right\|_\infty\leq\frac{\gamma}{1-\gamma}\max_{s,a}\left|\kappa_{\hat{\mathcal{P}}_{s,a}}(V^\pi)-\kappa_{\mathcal{P}_{0,s,a}}(V^\pi)\right|$$

Then using Lemma 2.7, we obtain :

$$\left\|Q^{\pi}-\hat{Q}^{\pi}\right\|_{\infty} \leq \frac{\gamma}{1-\gamma}\max_{s,a}\left|\kappa_{\hat{\mathcal{P}}_{s,a}}(V^{\pi})-\kappa_{\mathcal{P}_{0,s,a}}(V^{\pi})\right|$$

Taking $\pi=\pi^*$, V^{π^*} is independent of the data and we can use Lemma 2.8. Finally, we have

$$\left\|Q^{*}-\hat{Q}^{\pi^*}\right\|_{\infty} \leq \frac{\gamma}{1-\gamma}\left\|(\hat{P}-P^0)V^{\pi}\right\|_{\infty} \leq \frac{\gamma}{1-\gamma}\Big(2\sqrt{\frac{L}{2N(1-\gamma)^2}}+\frac{2L|S|^{1/q}\|1_S\|_q(p-1)}{N(1-\gamma)}\Big)$$

For the second point, using s or sa rectangular assumptions,

$$\begin{aligned}
\left\|Q^{*}-\hat{Q}^{*}\right\|_{\infty} &\leq \left\|\mathcal{T}^{\pi^*}_{\mathcal{U}^{sa}_p}Q^{*}-\hat{\mathcal{T}}^{\hat{\pi}^*}_{\mathcal{U}^{sa}_p}Q^{*}+\hat{\mathcal{T}}^{\hat{\pi}^*}_{\mathcal{U}^{sa}_p}Q^{*}-\hat{\mathcal{T}}^{\hat{\pi}^*}_{\mathcal{U}^{sa}_p}\hat{Q}^{*}\right\|_{\infty}\\
&\leq \left\|\mathcal{T}^{\pi^*}_{\mathcal{U}^{sa}_p}Q^{*}-\hat{\mathcal{T}}^{\hat{\pi}^*}_{\mathcal{U}^{sa}_p}Q^{*}\right\|_{\infty}+\left\|\hat{\mathcal{T}}^{\hat{\pi}^*}_{\mathcal{U}^{sa}_p}Q^{*}-\hat{\mathcal{T}}^{\hat{\pi}^*}_{\mathcal{U}^{sa}_p}\hat{Q}^{*}\right\|_{\infty}\\
&\overset{(a)}{\leq} \left\|\mathcal{T}^{\pi^*}_{\mathcal{U}^{sa}_p}Q^{*}-\hat{\mathcal{T}}^{\hat{\pi}^*}_{\mathcal{U}^{sa}_p}Q^{*}\right\|_{\infty}+\gamma\left\|Q^{*}-\hat{Q}^{*}\right\|_{\infty}\\
&\overset{(b)}{\leq} \left\|\kappa_{\hat{\mathcal{P}}_{s,a}}(V^{*})-\kappa_{\mathcal{P}_{0,s,a}}(V^{*})\right\|_{\infty}+\gamma\left\|Q^{*}-\hat{Q}^{*}\right\|_{\infty}
\end{aligned}$$

Then using Lemma 2.7, and solving we get :

$$\left\|Q^{*}-\hat{Q}^{*}\right\|_{\infty}\frac{\gamma}{1-\gamma}\left\|\kappa_{\hat{\mathcal{P}}_{s,a}}(V^{*})-\kappa_{\mathcal{P}_{0,s,a}}(V^{*})\right\|_{\infty}$$

Finally using Lemma 2.8, we obtain

$$\left\|Q^{*}-\hat{Q}^{*}\right\|_{\infty} \leq \frac{\gamma}{(1-\gamma)^2}\Big(2\sqrt{\frac{L}{2N}}+\frac{2L|S|^{1/q}\|1_S\|_q(p-1)}{N}\Big)$$

which concludes the proof.

Lemma 2.15 (Similar to Agarwal, Agarwal et al. (2020) lemma 9 but for RMPDs)**.** *With probability* $1-\delta$*, we have:*

$$\min_{u\in U_s}\left|\widehat{V}^{\star}(s)-u\right| \leq 4\gamma\Big(2\sqrt{\frac{L}{2N}}+\frac{2L|S|^{1/q}\|1_S\|_q(p-1)}{N}\Big)$$

Proof. The proof can be found in Agarwal et al. (2020) and is similar for RMDs than for classical MPDs and consists in choosing U_s to be the evenly spaced elements in the interval $\Big[V^{\star}(s)-\Delta_{\delta/2,N}V^{\star}(s)+\Delta_{\delta/2,N}\Big]$, then finally the size of U_s is chosen to be $|U_s|=\frac{1}{(1-\gamma)^2}$. Using lemma , with probability greater than $1-\delta/2$, we have $\widehat{V}^{\star}(s)\in\Big[V^{\star}(s)-\Delta_{\delta/2,N}V^{\star}(s)+\Delta_{\delta/2,N}\Big]$ for all s according to Lemma 2.14. This implies using that that $\hat{V}^{\pi^*}$ will land in one of $|U_s|-1$evenly sized sub-intervals of length $2\Delta_{\delta/2,N}$:

$$\begin{aligned}
&\min_{u\in U_s}\left|\widehat{V}^{\star}(s)-u\right| \leq \frac{2\Delta_{\delta/2,N}}{|U_s|-1}=\frac{2}{|U_s|-1}\frac{\gamma}{(1-\gamma)^2}\Big(2\sqrt{\frac{L}{2N}}+\frac{2L|S|^{1/q}\|1_S\|_q}{N}\Big)\\
&\leq 4\gamma\Big(2\sqrt{\frac{L}{2N}}+\frac{2L|S|^{1/q}\|1_S\|_q(p-1)}{N}\Big)
\end{aligned}$$

□

Lemma 2.16 (Relation between concentration of robust and non-robust MDPs)**.** *With probability* $1-\delta$*, we get:*

$$\max_{s,a}\left|\kappa_{\hat{\mathcal{P}}_{s,a}}(V^{\hat{\pi}})-\kappa_{\mathcal{P}_{0,s,a}}(V^{\hat{\pi}})\right|\leq\frac{10}{(1-\gamma)}\Big(\sqrt{\frac{L''}{2N}}+\frac{L''|S|^{1/q}\|1_S\|_q(p-1)}{N}\Big)+2\epsilon_{opt}.$$
$$\max_{s,a}\left|\kappa_{\widehat{\mathcal{P}}_{s,a}}(V^{*})-\kappa_{\mathcal{P}_{0,s,a}}(V^{*})\right|\leq\frac{10}{(1-\gamma)}\Big(\sqrt{\frac{L''}{2N}}+\frac{L''|S|^{1/q}\|1_S\|_q(p-1)}{N}\Big).$$

with $L''=\log\Big(\frac{32SAN\|1\|_q}{\delta(1-\gamma)}\Big)$

Proof. Using Lemma 2.7, we directly have the first inequality equality part of the first statement:

$$\max_{s,a}\left|\kappa_{\hat{\mathcal{P}}_{s,a}}(\hat{V}^{\hat{\pi}})-\kappa_{\mathcal{P}_{0,s,a}}(\hat{V}^{\hat{\pi}})\right|$$

is bounded by either by

$$\max_{(s,a)}\max_{\alpha_P^{\lambda,\omega}\in\mathrm{A}_P^{\lambda,\omega}}\left|\Big(P^0-\widehat{P}\Big)\,[\hat{V}^{\hat{\pi}}]_{\alpha_P^{\lambda,\omega}}\right|$$

or

$$\max_{(s,a)}\max_{\alpha_{\hat{P}}^{\lambda,\omega}\in\mathrm{A}_{\hat{P}}^{\lambda,\omega}}\left|\Big(P^0-\widehat{P}\Big)\,[\hat{V}^{\hat{\pi}}]_{\alpha_{\hat{P}}^{\lambda,\omega}}\right|.$$

We know that in both cases that

$$\max_{(s,a)}\left|\Big(P^0-\widehat{P}\Big)\,[\hat{V}^{\hat{\pi}}]_{\alpha_P^{\lambda,\omega}}\right|\leq\max_{(s,a)}|(P^0-\widehat{P})([\hat{V}^{\hat{\pi}}]_{\alpha_P^{\lambda,\omega}}-[\hat{V}^{*}]_{\alpha_P^{\lambda,\omega}})|+\max_{(s,a)}|(P^0-\widehat{P})[\hat{V}^{*}]_{\alpha_P^{\lambda,\omega}}|,$$

using $|[\hat{V}^{\hat{\pi}}]_{\alpha_P^{\lambda,\omega}}|-|[\hat{V}^{*}]_{\alpha_P^{\lambda,\omega}}|\leq|([\hat{V}^{\hat{\pi}}-\hat{V}^{*}]_{\alpha_P^{\lambda,\omega}})|\leq|(\hat{V}^{\hat{\pi}}-\hat{V}^{*})|$ and combining Lemma 2.13 and 2.15, for $|U_s|=\frac{1}{(1-\gamma)^2}$, with probability $1-\delta$, we have :

$$\begin{aligned}|\Big(P^0-\widehat{P}\Big)\,[\hat{V}^{\hat{\pi}}]_{\alpha_P^{\lambda,\omega}}|\leq&4\gamma\Big(2\sqrt{\frac{L''}{2N}}+\frac{2LS^{1/q}\|1_S\|_q}{N}\Big)+\frac{1}{(1-\gamma)}\Big(2\sqrt{\frac{L''}{2N}}+\frac{2L''S^{1/q}\|1_S\|_q}{N}\Big)+2\epsilon_{opt}.\\ \leq&\frac{10}{(1-\gamma)}\Big(\sqrt{\frac{L''}{2N}}+\frac{L''|S|^{1/q}\|1_S\|_q(p-1)}{N}\Big)+2\epsilon_{opt}.\end{aligned}$$

The proof is exactly the same by replacing $\hat{\pi}$ by π^* but without the $2\epsilon_{opt}$, which gives the second stated result. Again, this proof is written for the *sa*-rectangular assumption, it is also true for the *s*-rectangular case with slightly different notations, replacing $\mathcal{D}=\mathcal{P}_{0,s,a}$ by $\mathcal{D}=\mathcal{P}_{0,s}$. □

These two inequalities are the core of our proof, as the closed form solution of the min problem in the robust setting only depends on α,σ and the current value function.

Theorem 2.17. *Suppose* $\delta>0$*,* $\epsilon>0$ *and* $\sigma>0$*, let* $\widehat{\pi}$ *be any* ϵ_{opt} *-optimal policy for* $\widehat{M}$*, i.e.* $\left\|\widehat{Q}^{\widehat{\pi}}-\widehat{Q}^{\star}\right\|_\infty\leq\epsilon_{opt}$ *. If*

$$N\geq\frac{C\gamma^2L''}{(1-\gamma)^4\epsilon^2},$$

we get

$$\left\|Q^* - Q^{\hat{\pi}}\right\|_\infty \leq \epsilon + \frac{3\gamma\epsilon_{opt}}{1-\gamma}$$

with probability at least $1-\delta$*, where* C *is an absolute constant. Finally, for* $N_{total} = N|\mathcal{S}||\mathcal{A}|$ *and* $H = 1/(1-\gamma)$*, we get an overall complexity of*

$$N_{total} = \tilde{\mathcal{O}}\left(\frac{H^4 SA}{\epsilon^2}\right).$$

Proof.

$$\begin{aligned}
\left\|Q^* - Q^{\hat{\pi}}\right\|_\infty &\overset{(a)}{\leq} \left\|Q^* - \hat{Q}^*\right\|_\infty + \left\|\hat{Q}^* - \hat{Q}^{\hat{\pi}}\right\|_\infty + \left\|\hat{Q}^{\hat{\pi}} - Q^{\hat{\pi}}\right\|_\infty \\
&\overset{(b)}{\leq} \epsilon_{\text{opt}} + \frac{\gamma}{(1-\gamma)}\left(\max_{s,a}\left|\kappa_{\hat{\mathcal{P}}_{s,a}}\left(V^*\right) - \kappa_{\mathcal{P}_{s,a}}\left(V^*\right)\right| + \max_{s,a}\left|\kappa_{\mathcal{P}_{s,a}}\left(V^{\hat{\pi}}\right) - \kappa_{\mathcal{P}_{s,a}}\left(V^{\hat{\pi}}\right)\right|\right) \\
&\overset{(c)}{\leq} \frac{20\gamma}{(1-\gamma)^2}\Big(\sqrt{\frac{L''}{2N}} + \frac{L''|S|^{1/q}\|1_S\|_q(p-1)}{N}\Big) + \epsilon_{\text{opt}} + \frac{2\gamma\epsilon_{\text{opt}}}{1-\gamma} \\
&\leq \frac{20\gamma}{(1-\gamma)^2}\Big(\sqrt{\frac{L''}{2N}} + \frac{L''|S|^{1/q}\|1_S\|_q(p-1)}{N}\Big) + \epsilon_{\text{opt}} + \frac{2\gamma\epsilon_{\text{opt}}}{1-\gamma} \\
&\overset{(d)}{\leq} \epsilon + \frac{3\gamma\epsilon_{\text{opt}}}{1-\gamma}
\end{aligned}$$

Inequality (a) is due to Lemma 2.1. Inequality (b) comes from Lemma 2.4. Finally, inequality (c) comes from Lemma 2.16 and inequality (d) from the form of N in the theorem. For $N \geq H^4 SA$, the second term proportional to $1/N$ is very small compared to the asymptotic term in $1/\sqrt{N}$ for small ϵ. Note that $S^{1/q}\|1_S\|_q = |S|$ for L_2 norm for example. This proof holds for both s- and sa-rectangular assumptions. □

3 Towards minimax optimal bounds

We start from the same decomposition as the proof of Theorem 2.4.1 proved in Lemma 2.1:

$$\left\|Q^* - Q^{\hat{\pi}}\right\|_\infty \leq \left\|Q^* - \hat{Q}^{\pi^*}\right\|_\infty + \left\|\hat{Q}^{\pi^*} - \hat{Q}^{\hat{\pi}}\right\|_\infty + \left\|\hat{Q}^{\hat{\pi}} - Q^{\hat{\pi}}\right\|_\infty.$$

However, we need tighter concentration arguments for this proof.

In the following, we will frequently use the fact that, for any policy π, written below for the s-rectangular case (a similar expression can be obtained for the sa-rectangular case, adapting the regularized reward),

Recall, the fix point equation for Q^π can be written as :

$$Q^\pi = \left(I - \gamma P^{0,\pi}\right)^{-1}\left(r_0 - \alpha_s\left(\pi_s / \|\pi_s\|_q\right)^{q-1} + \gamma \inf_{P^\pi \in \mathcal{P}_s} P^\pi V^\pi\right) \tag{A.68}$$

It will be applied notably to $\hat{\pi}$ and π^* (recall that $Q^* = Q^{\pi^*}$), in the RMDP but also in the empirical one.

Lemma 3.1. *For s-rectangular we have*

$$\begin{aligned}\left(I-\gamma P^{0,\pi}\right)^{-1} r^s_{\hat{Q}^\pi} - \left(I-\gamma \widehat{P}^{\pi}\right)^{-1} r^s_{\hat{Q}^\pi} &\overset{(a)}{=} \left(I-\gamma P^{0,\pi}\right)^{-1}\left(\left(I-\gamma \widehat{P}^{\pi}\right)-\left(I-\gamma P^{0,\pi}\right)\right)\widehat{Q}^\pi_s \\ &= \gamma\left(I-\gamma P^{0,\pi}\right)^{-1}\left(P^{0,\pi}-\widehat{P}^{\pi}\right)\widehat{Q}^\pi_s \\ &= \gamma\left(I-\gamma P^{0,\pi}\right)^{-1}(P^0-\widehat{P})\widehat{V}^\pi_s\end{aligned}$$

and for optimal policy

$$\left(I-\gamma P^{0,\pi^*}\right)^{-1} r^s_{\hat{Q}^{\pi^*}_s} - \left(I-\gamma \hat{P}^{\pi^*}\right)^{-1} r^s_{\hat{Q}^{\pi^*}_s} = \gamma\left(I-\gamma P^{0,\pi^*}\right)^{-1}(P^0-\widehat{P})\hat{V}^{\pi^*}_s \tag{A.69}$$

$$\left(I-\gamma P^{0,\hat{\pi}}\right)^{-1} r^s_{\hat{Q}^{\hat{\pi}}_s} - \left(I-\gamma \hat{P}^{\hat{\pi}}\right)^{-1} r^s_{\hat{Q}^{\hat{\pi}}_s} = \gamma\left(I-\gamma P^{0,\hat{\pi}}\right)^{-1}(P^0-\widehat{P})\hat{V}^{\hat{\pi}}_s \tag{A.70}$$

The solution is a bit different as $r^s_{\hat{Q}^\pi}$ is the regularized form of the L_p optimization problem with simplex constraints which correspond to $r^s_{\hat{Q}^\pi} = r_0 - \left(\frac{\pi^*_s}{\|\pi^*_s\|_q}\right)^{q-1}\alpha_s + \gamma \inf_{P^\pi\in\mathcal{P}_s} P^\pi \hat{V}^\pi$ or for *sa* case : $r^{(s,a)}_{\hat{Q}^\pi} = r_0 - \alpha_{\text{sa}} + \gamma \inf_{P^\pi\in\mathcal{P}_s} P^\pi \hat{V}^\pi$

Indeed, even without close form, we can write the problem with an expectation over the nominal and the infimum problem.

Lemma 3.2 (Upper bound on $Q^* - \hat{Q}^{\pi^*}$ and on $Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}$, all Q values are now with robust under simplex constraints.)**.**

$$\begin{aligned}\left\|Q^* - \hat{Q}^{\pi^*}\right\|_\infty &\leq \gamma \left\|(I-\gamma P^{0,\pi^*})^{-1}(P^0-\widehat{P})\hat{V}^{\pi^*}\right\|_\infty + \frac{2\gamma\sigma S^{1/q}}{1-\gamma}\left\|Q^* - \hat{Q}^{\pi^*}\right\|_\infty \\ \left\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\right\|_\infty &\leq \gamma \left\|\left(I-\gamma P^{0,\hat{\pi}}\right)^{-1}(P^0-\widehat{P})\hat{V}^{\hat{\pi}}\right\|_\infty + \frac{2\gamma\sigma S^{1/q}}{1-\gamma}\left\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\right\|_\infty\end{aligned}$$

Proof.

$$\begin{aligned}&Q^* - \hat{Q}^{\pi^*} \\ &= \left(I-\gamma P^{0,\pi^*}\right)^{-1}\left(r_0 - \left(\frac{\pi^*_s}{\|\pi^*_s\|_q}\right)^{q-1}\alpha_s + \inf_{P^\pi\in\mathcal{P}_s} P^\pi V^*\right) \\ &\quad - \left(I-\gamma \hat{P}^{\pi^*}\right)^{-1}\left(r_0 - \left(\frac{\pi^*_s}{\|\pi^*_s\|_q}\right)^{q-1}\alpha_s + \inf_{P^\pi\in\mathcal{P}_s} P^\pi \hat{V}^{\pi^*}\right) \\ &= \left(I-\gamma P^{0,\pi^*}\right)^{-1}\left(r_0 - \left(\frac{\pi^*_s}{\|\pi^*_s\|_q}\right)^{q-1}\alpha_s + \gamma\inf_{P^\pi\in\mathcal{P}_s} P^\pi V^*\right) \\ &\quad - \left(I-\gamma P^{0,\pi^*}\right)^{-1}\left(r_0 - \left(\frac{\pi^*_s}{\|\pi^*_s\|_q}\right)^{q-1}\alpha_s + \gamma\inf_{P^\pi\in\mathcal{P}_s} P^\pi \hat{V}^{\pi^*}\right) \\ &\quad + \left(I-\gamma P^{0,\pi^*}\right)^{-1}\left(r_0 - \left(\frac{\pi^*_s}{\|\pi^*_s\|_q}\right)^{q-1}\alpha_s + \gamma\inf_{P^\pi\in\mathcal{P}_s} P^\pi \hat{V}^{\pi^*}\right) \\ &\quad - \left(I-\gamma \hat{P}^{\pi^*}\right)^{-1}\left(r_0 - \left(\frac{\pi^*_s}{\|\pi^*_s\|_q}\right)^{q-1}\alpha_s + \gamma\inf_{P^\pi\in\mathcal{P}_s} P^\pi \hat{V}^{\pi^*}\right) \\ &\overset{(a)}{=} \gamma\left(I-\gamma P^{0,\pi^*}\right)^{-1}(P^0-\widehat{P})\hat{V}^{\pi^*} + \left(I-\gamma P^{0,\pi^*}\right)^{-1}\gamma\left(\inf_{P^\pi\in\mathcal{P}_s} P^\pi V^* - \inf_{P^\pi\in\mathcal{P}_s} P^\pi \hat{V}^{\pi^*}\right)\end{aligned}$$

where in (a) we use previous Lemma 3.1. □

Hence, taking the supremum norm $\|.\|_\infty$,

$$
\begin{aligned}
&\left\|Q^* - \hat{Q}^{\pi^*}\right\|_\infty = \\
&\left\|\gamma\left(I - \gamma P^{0,\pi^*}\right)^{-1}(P^0 - \widehat{P})\hat{V}^{\pi^*} + \left(I - \gamma P^{0,\pi^*}\right)^{-1}\gamma\left(\inf_{P^\pi \in \mathcal{P}_s} P^\pi V^* - \inf_{P^\pi \in \mathcal{P}_s} P^\pi \hat{V}^{\pi^*}\right)\right\|_\infty \\
&\overset{(b)}{\leq} \left\|\gamma\left(I - \gamma P^{0,\pi^*}\right)^{-1}(P^0 - \widehat{P})\hat{V}^{\pi^*}\right\|_\infty + \left\|\left(I - \gamma P^{0,\pi^*}\right)^{-1}\gamma\left(\inf_{P^\pi \in \mathcal{P}_s} P^\pi V^* - \inf_{P^\pi \in \mathcal{P}_s} P^\pi \hat{V}^{\pi^*}\right)\right\|_\infty \\
&\overset{(c)}{\leq} \left\|\gamma\left(I - \gamma P^{0,\pi^*}\right)^{-1}(P^0 - \widehat{P})\hat{V}^{\pi^*}\right\|_\infty + \frac{\gamma}{1-\gamma} \mid \inf_{P^\pi \in \mathcal{P}_s} P^\pi V^* - \inf_{P^\pi \in \mathcal{P}_s} P^\pi \hat{V}^{\pi^*} \mid \\
&\overset{(d)}{\leq} \left\|\gamma\left(I - \gamma P^{0,\pi^*}\right)^{-1}(P^0 - \widehat{P})\hat{V}^{\pi^*}\right\|_\infty + \frac{\gamma}{1-\gamma} \sup_{P^\pi \in \mathcal{P}_s} P^\pi \mid V^* - \hat{V}^{\pi^*} \mid \\
&\overset{(e)}{\leq} \left\|\gamma\left(I - \gamma P^{0,\pi^*}\right)^{-1}(P^0 - \widehat{P})\hat{V}^{\pi^*}\right\|_\infty + \frac{\gamma}{1-\gamma} \sup_{P:\|P\|_p \leq \sigma_s, \sum_s P(s)=0} P \mid V^* - \hat{V}^{\pi^*} \mid \\
&\overset{(f)}{\leq} \left\|\gamma\left(I - \gamma P^{0,\pi^*}\right)^{-1}(P^0 - \widehat{P})\hat{V}^{\pi^*}\right\|_\infty - \frac{\gamma}{1-\gamma} \inf_{P:\|P\|_p \leq \sigma_s, \sum_s P(s)=0} -P \mid V^* - \hat{V}^{\pi^*} \mid \\
&\overset{(g)}{\leq} \left\|\gamma\left(I - \gamma P^{0,\pi^*}\right)^{-1}(P^0 - \widehat{P})\hat{V}^{\pi^*}\right\|_\infty + \frac{\gamma\sigma S^{1/q}}{1-\gamma}\mathrm{sp}_{q,\pi^*}(Q^* - \hat{Q}^{\pi^*}) \\
&\overset{(h)}{\leq} \left\|\gamma\left(I - \gamma P^{0,\pi^*}\right)^{-1}(P^0 - \widehat{P})\hat{V}^{\pi^*}\right\|_\infty + \frac{2\gamma\sigma S^{1/q}}{1-\gamma}\left\|Q^* - \hat{Q}^{\pi^*}\right\|_\infty
\end{aligned}
$$

where (b) is the triangular inequality, (c) Eq. (A.3), (d) is the triangular inequality for seminorms, (d) is $|\inf_A f - \inf_A g| \leq \sup_A |f - g|$., (e) is a relaxation (f) is the relation between sup and inf, (g) is lemma 1 of Kumar et al. (2022)), (h) is inequality for seminorms and norms (A.4).

For brevity in the remaining analysis, let us define the shorthand:

$$L = \log(8|\mathcal{S}||\mathcal{A}|/((1-\gamma)\delta)).$$

Recall, slightly abusing the notation, for $V \in \mathbb{R}^S$, we define the vector $\mathrm{Var}_P(V) \in \mathbb{R}^{\mathcal{S}\times A}$ as $\mathrm{Var}_P(V) = P(V)^2 - (PV)^2$.

Lemma 3.3 (Agarwal et al. (2020), Lemma 9)**.** *With probability greater than* $1-\delta$*,*

$$
\begin{aligned}
\left|(P^0 - \widehat{P})\widehat{V}^\star\right| &\leq \sqrt{\frac{8L}{N}}\sqrt{\mathrm{Var}_{P^0}\left(\widehat{V}^\star\right)} + \Delta'_{\delta,N}\mathbf{1} \\
\left|(P^0 - \widehat{P})\widehat{V}^{\pi^\star}\right| &\leq \sqrt{\frac{8L}{N}}\sqrt{\mathrm{Var}_{P^0}\left(\widehat{V}^{\pi^\star}\right)} + \Delta'_{\delta,N}\mathbf{1} \\
\textit{where } \Delta'_{\delta,N} &= \sqrt{\frac{cL}{N}} + \frac{cL}{(1-\gamma)N} \textit{ and } c \textit{ is a universal constant smaller than } 16.
\end{aligned}
$$

Proof. The proof of Agarwal et al. (2020) holds for classical MDP but can be adapted to the robust setting using all lemmas proved for the bound in H^4 previously. Lemma 2.11,2.12 ,2.14,2.15,A.53 are needed but the main difference is that we are using Berstein's inequality and not Hoeffding's inequality. The idea is first, as in the previous proof, to apply Berstein's inequality to independent variables using s absorbing MDPs then using Lemma 2.15.

Proof. Similar to Agarwal et al. (2020), we first show that

$$\left|\left(P^0-\widehat{P}\right)\cdot\widehat{V}^\star\right| \leq \sqrt{\frac{2\log\left(4\left|U_s\right|/\delta\right)}{N}}\sqrt{\mathrm{Var}_{P^0}\left(\widehat{V}^\star\right)} + \min_{u\in U_s}\left|\widehat{V}^\star(s)-u\right|\left(1+\sqrt{\frac{2\log\left(4\left|U_s\right|/\delta\right)}{N}}\right)+\frac{2\log\left(4\left|U_s\right|/\delta\right)}{(1-\gamma)3N}$$

$$\left|\left(P^0-\widehat{P}\right)\cdot\widehat{V}^{\pi^\star}\right| \leq \sqrt{\frac{2\log\left(4\left|U_s\right|/\delta\right)}{N}}\sqrt{\mathrm{Var}_{P^0}\left(\widehat{V}^{\pi^\star}\right)} + \min_{u\in U_s}\left|\widehat{V}^{\pi^\star}(s)-u\right|\left(1+\sqrt{\frac{2\log\left(4\left|U_s\right|/\delta\right)}{N}}\right)+\frac{2\log\left(4\left|U_s\right|/\delta\right)}{(1-\gamma)3N}$$

First, with probability greater than $1-\delta$, we have that for all $u\in U_s$.

$$\begin{aligned}
&\left|\left(P^0-\widehat{P}\right)\cdot\widehat{V}^\star\right| = \left|\left(P^0-\widehat{P}\right)\cdot\left(\widehat{V}^\star-V^\star_{s,u}+V^\star_{s,u}\right)\right| \\
&\overset{(a)}{\leq} \left|\left(P^0-\widehat{P}\right)\cdot\left(\widehat{V}^\star-V^\star_{s,u}\right)\right| + \left|\left(P^0-\widehat{P}\right)\cdot\left(V^\star_{s,u}\right)\right| \\
&\overset{(b)}{\leq} \left\|\widehat{V}^\star-V^\star_{s,u}\right\|_\infty + \sqrt{\frac{2\log\left(4\left|U_s\right|/\delta\right)}{N}}\sqrt{\mathrm{Var}_{P^0}\left(V^\star_{s,u}\right)} + \frac{2\log\left(4\left|U_s\right|/\delta\right)}{(1-\gamma)3N} \\
&\overset{(c)}{=} \left\|\widehat{V}^\star-V^\star_{s,u}\right\|_\infty + \sqrt{\frac{2\log\left(4\left|U_s\right|/\delta\right)}{N}}\sqrt{\mathrm{Var}_{P^0}\left(\widehat{V}^\star-V^\star_{s,u}-\widehat{V}^\star\right)} + \frac{2\log\left(4\left|U_s\right|/\delta\right)}{(1-\gamma)3N} \\
&\overset{(d)}{\leq} \left\|\widehat{V}^\star-V^\star_{\widehat{M}_{s,u}}\right\|_\infty\left(1+\sqrt{\frac{2\log\left(4\left|U_s\right|/\delta\right)}{N}}\right) + \sqrt{\frac{2\log\left(4\left|U_s\right|/\delta\right)}{N}}\sqrt{\mathrm{Var}_{P^0}\left(\widehat{V}^\star\right)} + \frac{2\log\left(4\left|U_s\right|/\delta\right)}{(1-\gamma)3N}
\end{aligned}$$

using the triangle inequality in (a), (b) classical Berstein's inequality, (d) for variance and Lemmas 2.11 and 2.12 such as

$$\left\|\widehat{V}^\star-V^\star_{s,u}\right\|_\infty = \left\|\widehat{V}^\star_{s,\widehat{V}^\star(s)}-V^\star_{s,u}\right\|_\infty \leq \left|\widehat{V}^\star(s)-u\right|.$$

It is true for $u\in U_s$, so we take the best possible choice, which completes the proof of the first claim. The proof of the second claim is similar. Then using Lemma 2.15 gives the final concentration theorem. □

□

Lemma 3.4 (Azar et al. (2013a), Lemma 7)**.** *This is an adaptation of Azar et al. (2013a) to RMDPs. For any policy π,*

$$\left\|\left(I-\gamma P^{0,\pi}\right)^{-1}\sqrt{\mathrm{Var}_{P^0}\left(V^\pi\right)}\right\|_\infty \leq \sqrt{\frac{2}{(1-\gamma)^3}},$$

where P^0 is the nominal transition model of M.

Proof. This proof is exactly the same for Robust and non robust MDPs, as it uses only standard computations such as the Jensen inequality and no robust form which are specific to this problem. The main difference is that we are doing the proof on the nominal of our robust set P^0, considering the regularized robust Bellman operator and associated regularized reward functions.

Azar et al. (2013a) introduce the variance of the sum of discounted rewards starting at state-action (s, a),

$$\Sigma^{\pi}(s,a) := \mathbb{E}[|\sum_{t\geq 0}\gamma^t r_0(s_t,a_t) - Q^{\pi}(s,a)|^2 | s_0 = s, a_0 = a],$$

and we defined the same variance for robust MDPs using robust rewards $r^{(s,a)}_{Q^\pi}$ and $r^{s}_{Q^\pi}$ and using robust Q-function instead of classical Q-function in the definition of Σ. Then, in their Lemma 6 they show that, for any π:

$$\Sigma^{\pi} = \mathrm{Var}_{P^0}\left(V^{\pi}\right) + \gamma^2 P^{0,\pi}\Sigma^{\pi},$$

which is, in fact, a Bellman equation for the variance. The proof is exactly the same for RMDPs considering our robust reward $r^{(s,a)}_{Q^\pi}$ or $r^{s}_{Q^\pi}$ and not classical r_0. Note that this is thanks to the regularized form of robust RMDPs. Finally, Lemma 3.4 is the same as their Lemma 7 considering robust rewards. This lemma is usually called the total variance lemma. This completes the proof. □

Lemma 3.5. *The following upper bound holds with probability* $1-\delta$*:*

$$\left\|Q^{\widehat{\pi}} - \widehat{Q}^{\widehat{\pi}}\right\|_{\infty} < (C_N + C_\sigma)\left\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\right\|_{\infty} + \gamma 4\sqrt{\frac{L}{N(1-\gamma)^3}} + \frac{\gamma\Delta'_{\delta,N}}{1-\gamma} + \frac{\gamma\epsilon_{\text{opt}}}{1-\gamma}\left(2 + \sqrt{\frac{8L}{N}}\right) \quad \text{(A.71)}$$

with $C_N = \frac{\gamma}{1-\gamma}\sqrt{\frac{8L}{N}}$ *and* $C_\sigma = \frac{2\gamma\sigma S^{1/q}}{1-\gamma}$.

Proof.

$$
\begin{aligned}
&\left\|Q^{\widehat{\pi}} - \widehat{Q}^{\widehat{\pi}}\right\|_\infty \\
&\overset{(a)}{\leq} \gamma \left\| \left(I - \gamma P^{0,\widehat{\pi}}\right)^{-1} (P^0 - \widehat{P}) \widehat{V}^{\widehat{\pi}} \right\|_\infty + \frac{2\gamma\sigma S^{1/q}}{1-\gamma} \left\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\right\|_\infty \\
&\overset{(b)}{\leq} \gamma \left\| \left(I - \gamma P^{\widehat{\pi}}\right)^{-1} (P^0 - \widehat{P}) \widehat{V}^{\star} \right\|_\infty + \gamma \left\| \left(I - \gamma P^{0,\pi}\right)^{-1} (P^0 - \widehat{P}) \left(\widehat{V}^{\widehat{\pi}} - \widehat{V}^{\star}\right) \right\|_\infty \\
&+ \frac{2\gamma\sigma S^{1/q}}{1-\gamma} \left\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\right\|_\infty \\
&\overset{(c)}{\leq} \gamma \left\| \left(I - \gamma P^{0,\widehat{\pi}}\right)^{-1} (P^0 - \widehat{P}) \widehat{V}^{\star} \right\|_\infty + \frac{2\gamma\epsilon_{\text{opt}}}{1-\gamma} + \frac{2\gamma\sigma S^{1/q}}{1-\gamma} \left\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\right\|_\infty \\
&\overset{(d)}{\leq} \gamma \left\| \left(I - \gamma P^{0,\widehat{\pi}}\right)^{-1} \left|(P^0 - \widehat{P}) \widehat{V}^{\star}\right| \right\|_\infty + \frac{2\gamma\epsilon_{\text{opt}}}{1-\gamma} + \frac{2\gamma\sigma S^{1/q}}{1-\gamma} \left\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\right\|_\infty \\
&\overset{(e)}{\leq} \gamma \sqrt{\frac{8L}{N}} \left\| \left(I - \gamma P^{0,\widehat{\pi}}\right)^{-1} \sqrt{\operatorname{Var}_{P^0}\left(\widehat{V}^{\star}\right)} \right\|_\infty + 2\frac{\gamma\Delta'_{\delta,N}}{1-\gamma} + \frac{2\gamma\epsilon_{\text{opt}}}{1-\gamma} + \frac{2\gamma\sigma S^{1/q}}{1-\gamma} \left\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\right\|_\infty \\
&\overset{(f)}{\leq} \gamma \sqrt{\frac{8L}{N}} \left\| \left(I - \gamma P^{0,\widehat{\pi}}\right)^{-1} \left(\sqrt{\operatorname{Var}_{P^0}\left(V^{\widehat{\pi}}\right)} + \sqrt{\operatorname{Var}_{P^0}\left(V^{\widehat{\pi}} - \widehat{V}^{\widehat{\pi}}\right)} + \sqrt{\operatorname{Var}_{P^0}\left(\widehat{V}^{\widehat{\pi}} - \widehat{V}^{\star}\right)} \right) \right\|_\infty \\
&\quad + \frac{\gamma\Delta'_{\delta,N}}{1-\gamma} + \frac{2\gamma\epsilon_{\text{opt}}}{1-\gamma} + \frac{2\gamma\sigma}{1-\gamma} \left\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\right\|_\infty \\
&\overset{(g)}{\leq} \gamma \sqrt{\frac{8L}{N}} \left(\sqrt{\frac{2}{(1-\gamma)^3}} + \frac{\sqrt{\left\|V^{\widehat{\pi}} - \widehat{V}^{\widehat{\pi}}\right\|_\infty^2}}{1-\gamma} + \frac{2\epsilon_{\text{opt}}}{1-\gamma} \right) + \frac{\gamma\Delta'_{\delta,N}}{1-\gamma} + \frac{2\gamma\epsilon_{\text{opt}}}{1-\gamma} + \frac{2\gamma\sigma}{1-\gamma} \left\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\right\|_\infty \\
&\overset{(h)}{\leq} \gamma \sqrt{\frac{8L}{N}} \left(\sqrt{\frac{2}{(1-\gamma)^3}} + \frac{\left\|Q^{\widehat{\pi}} - \widehat{Q}^{\widehat{\pi}}\right\|_\infty}{1-\gamma} + \frac{2\epsilon_{\text{opt}}}{1-\gamma} \right) + \frac{\gamma\Delta'_{\delta,N}}{1-\gamma} + \frac{2\gamma\epsilon_{\text{opt}}}{1-\gamma} + \frac{2\gamma\sigma S^{1/q}}{1-\gamma} \left\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\right\|_\infty \\
&= \gamma \sqrt{\frac{8L}{N}} \left(\sqrt{\frac{2}{(1-\gamma)^3}} + \frac{\left\|Q^{\widehat{\pi}} - \widehat{Q}^{\widehat{\pi}}\right\|_\infty}{1-\gamma} \right) + \frac{\gamma\Delta'_{\delta,N}}{1-\gamma} + \frac{\gamma\epsilon_{\text{opt}}}{1-\gamma} \left(2 + \sqrt{\frac{8L}{N}}\right) \\
&+ \frac{2\gamma\sigma S^{1/q}}{1-\gamma} \left\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\right\|_\infty \\
&= (C_N + C_\sigma) \left\|Q^{\hat{\pi}} - \hat{Q}^{\hat{\pi}}\right\|_\infty + 4\gamma \sqrt{\frac{L}{N(1-\gamma)^3}} + \frac{\gamma\Delta'_{\delta,N}}{1-\gamma} + \frac{\gamma\epsilon_{\text{opt}}}{1-\gamma} \left(2 + \sqrt{\frac{8L}{N}}\right)
\end{aligned}
$$

with $C_N = \frac{\gamma}{1-\gamma}\sqrt{\frac{8L}{N}}$ and $C_\sigma = \frac{2\gamma\sigma S^{1/q}}{1-\gamma}$.

We have that (a) is true by Lemma 3.2, (b) is by the triangular inequality using $\widehat{V}^{\widehat{\pi}} = \widehat{V}^{\widehat{\pi}} + \widehat{V}^{\star} - \widehat{V}^{\star}$, (c) is from the definition of ϵ_{opt} and Eq. (A.3), (d) is by positivity of the classic horizon inverse matrix, that is $(I - \gamma P)^{-1} = \sum_{t>0} \gamma^t P^t > 0$, (e) is by Lemma 3.3, (f) is by the triangular inequality for the variance (which is, in fact, a seminorm) and decomposing $\widehat{V}^{\star} = \widehat{V}^{\star} + \widehat{V}^{\widehat{\pi}} - \widehat{V}^{\widehat{\pi}} + V^{\widehat{\pi}} - V^{\widehat{\pi}}$, (g) is by Lemma 3.4, uses the definition of ϵ_{opt} and takes the sup over (s, a) of the variance in the second term, and eventually (h) is because we have that $\|V^{\pi} - \widehat{V}^{\pi}\|_\infty \leq \|Q^{\pi} - \widehat{Q}^{\pi}\|_\infty$ for any π.

□

Lemma 3.6. *The following upper bound holds with probability* $1-\delta$*:*

$$\left\|Q^* - \hat{Q}^{\pi^*}\right\|_\infty < (C_N + C_\sigma)\left\|Q^* - \hat{Q}^{\pi^*}\right\|_\infty + \gamma 4\sqrt{\frac{L}{N(1-\gamma)^3}} + \frac{\gamma\Delta'_{\delta,N}}{1-\gamma}. \tag{A.72}$$

with $C_N = \frac{\gamma}{1-\gamma}\sqrt{\frac{8L}{N}}$ *and* $C_\sigma = \frac{2\gamma\sigma S^{1/q}}{1-\gamma}$.

Proof.

$$\begin{aligned}
\left\|Q^* - \hat{Q}^{\pi^*}\right\|_\infty &\overset{(a)}{\leq} \gamma\left\|\left(I - \gamma P^{0,\pi^*}\right)^{-1}(P^0 - \widehat{P})\hat{V}^{\pi^*}\right\|_\infty + \frac{2\gamma\sigma S^{1/q}}{1-\gamma}\left\|Q^* - \hat{Q}^{\pi^*}\right\|_\infty \\
&\overset{(b)}{\leq} \gamma\left\|\left(I - \gamma P^{0,\pi^*}\right)^{-1}\left|(P^0 - \widehat{P})\hat{V}^{\pi^*}\right|\right\|_\infty + \frac{2\gamma\sigma S^{1/q}}{1-\gamma}\left\|Q^* - \hat{Q}^{\pi^*}\right\|_\infty \\
&\overset{(c)}{\leq} \gamma\sqrt{\frac{8L}{N}}\left\|\left(I - \gamma P^{0,\pi^*}\right)^{-1}\sqrt{\mathrm{Var}_{P^0}\left(\hat{V}^{\pi^*}\right)}\right\|_\infty + 2\frac{\gamma\Delta'_{\delta,N}}{1-\gamma} + \frac{2\gamma\sigma S^{1/q}}{1-\gamma}\left\|Q^* - \hat{Q}^{\pi^*}\right\|_\infty \\
&\overset{(d)}{\leq} \gamma\sqrt{\frac{8L}{N}}\left\|\left(I - \gamma P^{0,\pi^*}\right)^{-1}\left(\sqrt{\mathrm{Var}_{P^0}\left(V^*\right)} + \sqrt{\mathrm{Var}_{P^0}\left(V^* - \hat{V}^{\pi^*}\right)}\right)\right\|_\infty \\
&\quad + \frac{\gamma\Delta'_{\delta,N}}{1-\gamma} + \frac{2\gamma\sigma S^{1/q}}{1-\gamma}\left\|Q^* - \hat{Q}^{\pi^*}\right\|_\infty \\
&\overset{(e)}{\leq} \gamma\sqrt{\frac{8L}{N}}\left(\sqrt{\frac{2}{(1-\gamma)^3}} + \frac{\sqrt{\left\|V^* - \hat{V}^{\pi^*}\right\|_\infty^2}}{1-\gamma}\right) + \frac{\gamma\Delta'_{\delta,N}}{1-\gamma} + \frac{2\gamma\sigma S^{1/q}}{1-\gamma}\left\|Q^* - \hat{Q}^{\pi^*}\right\|_\infty \\
&\leq \gamma\sqrt{\frac{8L}{N}}\left(\sqrt{\frac{2}{(1-\gamma)^3}} + \frac{\left\|Q^* - \hat{Q}^{\pi^*}\right\|_\infty}{1-\gamma}\right) + \frac{\gamma\Delta'_{\delta,N}}{1-\gamma} + \frac{2\gamma\sigma S^{1/q}}{1-\gamma}\left\|Q^* - \hat{Q}^{\pi^*}\right\|_\infty \\
&= (C_N + C_\sigma)\left\|Q^* - \hat{Q}^{\pi^*}\right\|_\infty + 4\gamma\sqrt{\frac{L}{N(1-\gamma)^3}} + \frac{\gamma\Delta'_{\delta,N}}{1-\gamma}
\end{aligned}$$

with $C_N = \frac{\gamma}{1-\gamma}\sqrt{\frac{8L}{N}}$ and $C_\sigma = \frac{2\gamma\sigma S^{1/q}}{1-\gamma}$.

We have that (a) is true by Lemma 3.2, (b) is by the positivity of the classic horizon inverse matrix, (c) is by Lemma (3.3), (d) is by the triangular inequality for the variance (which is a seminorm), (e) is by Lemma 3.4 and taking the sup over (s,a) of the variance in the second term, and eventually (h) is because $\|V^\pi - \widehat{V}^\pi\|_\infty \leq \|Q^\pi - \widehat{Q}^\pi\|_\infty$ for any π.

□

As the event on which $\Delta'_{\delta,N}$ is the same in the two previous Lemma 3.5 and Lemma 3.6, we can obtain the following.

Theorem 3.7. *For* $0 < C_\sigma \leq 1/2$ *and* $0 < C_N + C_\sigma < 1$*, with probability* $1-\delta$*, we get:*

$$\left\|Q^* - Q^{\hat{\pi}}\right\|_\infty < \frac{1}{1-(C_N + C_\sigma)}\left(8\gamma\sqrt{\frac{L}{N(1-\gamma)^3}} + \frac{2\gamma\Delta'_{\delta,N}}{1-\gamma} + \frac{\gamma\epsilon_{\text{opt}}}{1-\gamma}\left(2 + \sqrt{\frac{8L}{N}}\right)\right) + \epsilon_{\text{opt}}.$$

Proof. This result is obtained by combining the two previous Lemmas 3.5 and 3.6 and passing the term in $(C_N + C_\sigma)$ to the left-hand side. □

Note that $C_\sigma + C_N < 1$ implies $C_\sigma = \frac{2\gamma\sigma S^{1/q}}{1-\gamma} < 1$ and hence $\sigma < \frac{1-\gamma}{2\gamma S^{1/q}}$. Now we need to pick $C_N < 1 - C_\sigma$. Let $C_N \leq 1 - C_\sigma - \eta$, for any $0 < \eta < 1 - C_\sigma$ the previous inequality becomes

$$\left\|Q^* - Q^{\hat{\pi}}\right\|_\infty < \frac{8}{\eta}\gamma\sqrt{\frac{L}{N(1-\gamma)^3}} + \frac{2\gamma\Delta'_{\delta,N}}{\eta(1-\gamma)} + \frac{\gamma\epsilon_{\text{opt}}}{\eta(1-\gamma)}\left(2 + \sqrt{\frac{8L}{N}}\right) + \epsilon_{\text{opt}}.$$

As $\Delta'_{\delta,N} = \sqrt{\frac{cL}{N}} + \frac{cL}{(1-\gamma)N}$, the term in $1/\sqrt{N}$ is given by $\frac{8\gamma\sqrt{L}H^{3/2}}{\eta\sqrt{N}}\left(1 + 1/4\sqrt{c/H}\right)$ and is smaller than ϵ whenever

$$N \geq \frac{64\gamma^2 LH^3(1 + 1/4\sqrt{c/H})^2}{\eta^2\epsilon^2}.$$

We will use $c < 16$ and $H \geq 1$ and use the stronger constraint

$$N \geq \frac{256\gamma^2 LH^3}{\eta^2\epsilon^2}.$$

Along the same line, the term in $1/N$ is $\frac{2\gamma cLH^2}{\eta N}$ which is smaller than ϵ whenever

$$N \geq \frac{2\gamma cLH^2}{\epsilon}.$$

Now, $C_N < 1 - \eta - C_\sigma$ means

$$\frac{\gamma}{1-\gamma}\sqrt{\frac{8L}{N}} < 1 - \eta - C_\sigma$$

hence

$$N > \frac{8L\gamma^2 H^2}{(1 - \eta - C_\sigma)^2}.$$

We deduce that whenever

$$\begin{aligned} N &\geq \max\left(\frac{256\gamma^2 LH^3}{\eta^2\epsilon^2}, \frac{2\gamma cLH^2}{\epsilon}, \frac{8L\gamma^2 H^2}{(1-\eta-C_\sigma)^2}\right) \\ &= \frac{256\gamma^2 LH^3}{\eta^2}\max\left(\frac{1}{\epsilon^2}, \frac{c\eta}{128H\gamma\epsilon}, \frac{\eta^2}{64H(1-\eta-C_\sigma)^2}\right) \end{aligned}$$

the error is smaller than 2ϵ up to the ϵ_{opt} terms.

This bounds reduces to

$$N \geq \frac{C\gamma^2 LH^3}{\epsilon^2}$$

with $C = 256/\eta^2$ if

$$\epsilon \leq \min\left(\frac{128H}{\eta}, \sqrt{64H}\frac{1-\eta-C_\sigma}{\eta}\right).$$

Note that $\epsilon \in [0, H)$ and $\eta < 1$ so that the previous condition simplifies to

$$\epsilon \leq \sqrt{64H}\frac{1-\eta-C_\sigma}{\eta} = \epsilon_0.$$

If we want to obtain an arbitrary ϵ_0, it suffices thus to take η arbitrarily small leading to the constant $C = 256/\eta^2$ to be arbitrarily large.

Note that if $\epsilon_0 \geq O(H^{1/2+\delta})$ then $1/\eta > O(H^\delta)$ which adds a $H^{2\delta}$ factor to the bound on N.

However, for any $\kappa\sqrt{H}$ and for any C_σ, it exists an η independent of H so that $\epsilon_0 = 8\sqrt{H}\frac{1-\eta-C_\sigma}{\eta} = \kappa\sqrt{H}$, hence the result stated in Theorem 2.5.1. Now, as $L = \log(8|\mathcal{S}||\mathcal{A}|/((1-\gamma)\delta))$, the previous condition can be summarized by

$$N_{\text{total}} = N|\mathcal{S}||\mathcal{A}| = \tilde{\mathcal{O}}\left(\frac{H^3|\mathcal{S}||\mathcal{A}|}{\epsilon^2}\right)$$

provided $\epsilon < \epsilon_0$. Finally, taking $\sigma_0 = \frac{1-\gamma}{8\gamma}$ which gives $C_\sigma = 1/4$ and $\eta = 1/2$ so that $C_N \leq 1/4$, we obtain $C = 1024$ and $\epsilon_0 = \sqrt{16H}$.

Appendix of Chapter 3

4 Other related works

Here we provide additional discussion of related work that could not be fit into the main paper due to space considerations. We limit our discussions to the tabular setting with finite state and action spaces provable RL algorithms.

Classical reinforcement learning with finite-sample guarantees. A recent surge in attention for RL has leveraged the methodologies derived from high-dimensional probability and statistics to analyze RL algorithms in non-asymptotic scenarios. Substantial efforts have been devoted to conducting non-asymptotic sample analyses of standard RL in many settings. Illustrative instances encompass investigations employing Probably Approximately Correct (PAC) bonds in the context of *generative model* settings (Kearns and Singh 1999, Beck and Srikant 2012, Li et al. 2022, Chen et al. 2020, Azar et al. 2013b, Sidford et al. 2018, Agarwal et al. 2020, Li et al. 2023; 2020, Wainwright 2019) and the *online setting* via both in PAC-base or regret-based analyses (Jin et al. 2018, Bai et al. 2019, Li et al. 2021, Zhang et al. 2020, Dong et al. 2019, Jin et al. 2020, Li et al. 2023, Jafarnia-Jahromi et al. 2020, Yang et al. 2021) and finally *offline setting* (Rashidinejad et al. 2021, Xie et al. 2021, Yin et al. 2021, Shi et al. 2022, Li et al. 2022, Jin et al. 2021, Yan et al. 2022).

Robustness in reinforcement learning. Reinforcement learning has had notable achievements but has also exhibited significant limitations, particularly when the learned policy is susceptible to deviations in the deployed environment due to perturbations, model discrepancies, or structural modifications. To address these challenges, the idea of robustness in RL algorithms has been studied. Robustness could concern uncertainty or perturbations across different Markov Decision Processes (MDPs) components, encompassing reward, state, action, and the transition kernel. Moos et al. (2022) gives a recent overview of the different work in this field.

The distributionally robust MDP (RMDP) framework has been proposed (Iyengar 2005) to enhance the robustness of RL has been proposed. In addition to this work, various other research efforts, including, but not limited to, Zhang et al. (2020; 2021), Han et al. (2022), Clavier et al. (2022), Qiaoben et al. (2021), explore robustness regarding state uncertainty. In these scenarios, the agent's policy is determined on the basis of perturbed observations generated from the state, introducing restricted noise, or undergoing adversarial attacks. Finally, robustness considerations extend to uncertainty in the action domain. Works such as Tessler et al. (2019), Tan et al. (2020) consider the robustness of actions, acknowledging potential distortions introduced by an adversarial agent.

Given the focus of our work, we provide a more detailed background on progress related to distributionally robust RL. The idea of distributionally robust optimization has been explored within the context of supervised learning (Rahimian and Mehrotra 2019, Gao 2020, Duchi and Namkoong 2018, Blanchet and Murthy 2019) and has also been extended to distributionally robust dynamic programming and Distributionally Robust Markov Decision Processes (DRMDPs) such as in

(Iyengar 2005, Xu and Mannor 2012, Wolff et al. 2012, Kaufman and Schaefer 2013, Ho et al. 2018, Smirnova et al. 2019a, Ho et al. 2021, Goyal and Grand-Clement 2022, Derman and Mannor 2020, Tamar et al. 2014, Badrinath and Kalathil 2021). Despite the considerable attention received, both empirically and theoretically, most previous theoretical analyses in the context of RMDPs adopt an asymptotic perspective (Roy et al. 2017) or focus on planning with exact knowledge of the uncertainty set (Iyengar 2005, Xu and Mannor 2012, Tamar et al. 2014). Many works have focused on the finite-sample performance of verifiable robust Reinforcement Learning (RL) algorithms. These investigations encompass various data generation mechanisms and uncertainty set formulations over the transition kernel. Closely related to our work, various forms of uncertainty sets have been explored, showcasing the versatility of approaches. Divergence such as Kullback-Leibler (KL) divergence is another prevalent choice, extensively studied by Yang et al. (2021), Panaganti and Kalathil (2022b), Zhou et al. (2021), Shi and Chi (2022), Xu et al. (2023), Wang et al. (2023), Blanchet et al. (2023), who investigated the sample complexity of both model-based and model-free algorithms in simulator or offline settings. Xu et al. (2023) considered various uncertainty sets, including those associated with the Wasserstein distance. The introduction of an R-contamination uncertainty set Wang and Zou (2021), has been proposed to tackle a robust Q-learning algorithm for the online setting, with guarantees analogous to standard RL. Finally, the finite-horizon scenario has been studied by Xu et al. (2023), Dong et al. (2022) with finite-sample complexity bounds for (RMDPs) using TV and χ^2 divergence. More broadly, other related topics have been explored, such as the iteration complexity of policy-based methods (Li et al. 2022, Kumar et al. 2023), and regularization-based robust RL (Yang et al. 2023). Finally, Badrinath and Kalathil (2021) examined a general *sa*-rectangular form of the uncertainty set, proposing a model-free algorithm for the online setting with linear function approximation to address large state spaces.

5 Further discussions of Theorem 3.4.1 and Theorem 3.4.3

- *What norms are included in the Definition 3.2.1?* In our upper bound result Theorems 3.4.3 and 3.4.1, we upper bound the sample complexity for C^2 norms and TV. The set of C^2 smooth norm is very large as it includes all, L_p norm, weighted, rescaled L_p norms for $p \geq 2$. Weighted norms can be useful in practice, to get more weights on dangerous specific states in Robust MDPs formulation such as in Russel et al. (2019). Moreover, note that our result can generalize to metric or pseudo metric (which are not homogeneous ie $\|\lambda\| = |\lambda|\|x\| \forall x \in \mathbb{R}^n, \lambda \in \mathbb{R}$) with norms of the form $x \mapsto \phi^{-1}(\sum_{k=1}^n, \phi(|x_k|))$ with ϕ a convex incising function such as the norm is still positive, definite positive. Choosing $\phi(x) = x^p$ leads to the L_p norms.

- *Assumptions on γ in Theorems 3.4.1 and 3.4.3, and Assumptions on γ for lower bound.* When γ is small (e.g., $\gamma \in (0, \frac{1}{2}]$ leads to the effective horizon length is at most 2), the sequential structure almost disappears and is much less of interest for RL community. So people Li et al. (2020) Yan et al. (2023) usually focus on reasonable range $\gamma \in (c, 1)$ for some small positive constant c, such as $\gamma \in [\frac{1}{2}, 1)$. However, the theorems can be directly extended to a broader range of $\gamma \in (c, 1)$ along with c as small as desired so that almost cover the full range $(0, 1)$.

- *Why final results on s depend on $\hat{\pi}$?*

  Theorem 3.4.3 is $\hat{\pi}$ data dependent which is randomness-dependent measure. However, taking the minimum of this quantity leads to the same bound as is *sa*-rectangular, so to illustrate that it is possible to get tighter bounds for *s*-rectangular with instance-dependent RMDPs, we decide to write also randomness-dependent quantity, while the less tight upper bound is written also in the theorem, taking the first term in the min operator in (3.21).

- *Why our results are still true for TV ?* Theorems 3.4.1 and 3.4.3 are stated for C^2 smooth norms, however, our result is still true for TV which is not C^2 as in this specific case, the dual of the optimization problem becomes a $1-$dimensional problem. In this case in the main concentration lemma 7.4, the additional term involving smoothness term denoted C_S is not present and the bound is simpler as is not required this additional term.

- *Why burn-in or sufficiently small ϵ condition is not too restrictive?* The burn-in term in Th. 3.4.1 and 3.4.3 is proportional to $1/\epsilon$ where the "sample complexity" term is proportional to $1/\epsilon^2$. The smooth term depending on C_S or burn-in is then not too large for sufficiently small ϵ compared to the other term, which will give final sample complexity.

- *Why this is not extendable to f-divergence currently?* The f-divergence as a distinct family of divergence is beyond the scope of this paper. Current proof for arbitrary norms cannot be directly extended since the key phenomenon of shrinking range of the robust value function has not been verified for f-divergence yet, while it is promising as an interesting future direction.

6 Preliminaries

These quantities appear in the dual formulation of the robust optimization problem and more preciously the dual span semi norm $\mathrm{sp}(.)_*$ note that for L_2, we retrieve the classical mean with the definition of ω) With slight abuse of notation, we denote 0 (resp. 1) as the all-zero (resp. all-one) vector. We then introduce the notation $[T] \coloneqq \{1, \cdots, T\}$ for any positive integer $T > 0$. Then, for all $1 \leq i \leq n$, for two vectors $x = [x_i]_{1 \leq i \leq n}$ and $y = [y_i]_{1 \leq i \leq n}$, the notation $x \leq y$ (resp. $x \geq y$) means $x_i \leq y_i$ (resp. $x_i \geq y_i$) . Finally, for any vector x, the notation is overloaded by letting $x^{\circ 2} = [x(s,a)^2]_{(s,a) \in \mathcal{S} \times \mathcal{A}}$ (resp. $x^{\circ 2} = [x(s)^2]_{s \in \mathcal{S}}$), Finally, we drop the subscript $\|.\|$ to write $\mathcal{U}^{\sigma}_{\|.\|}(\cdot) = \mathcal{U}^{\sigma}(\cdot)$ for both sa- and s- rectangular assumptions such that we write uncertainty set in the for sa-rectangular case $\mathcal{U}^{\mathsf{sa},\sigma}(.)$ or $\mathcal{U}^{\mathsf{s},\widetilde{\sigma}}(.)$ in the s-rectangular assumptions.

Matrix and Vector Notations. We define the following notation.

- $r \in \mathbb{R}^{SA}$ the reward function, such that $r_{(s,a)} = r(s,a)$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$.
- $P^0 \in \mathbb{R}^{SA \times S}$ the nominal transition kernel matrix using $P^0_{s,a}$ as the (s,a)-th row.
- $\widehat{P}^0 \in \mathbb{R}^{SA \times S}$ the estimated nomimal transition kernel matrix with $\widehat{P}^0_{s,a}$ as the (s,a)-th row.
- $\Pi^\pi \in \{0,1\}^{S \times SA}$ the projection matrix associated with a policy π

$$\Pi^\pi = \begin{pmatrix} 1^\top_{\pi(1)} & 0^\top & \cdots & 0^\top \\ 0^\top & 1^\top_{\pi(2)} & \cdots & 0^\top \\ \vdots & \vdots & \ddots & \vdots \\ 0^\top & 0^\top & \cdots & 1^\top_{\pi(S)} \end{pmatrix}, \tag{A.73}$$

where $1^\top_{\pi(1)}, 1^\top_{\pi(2)}, \ldots, 1^\top_{\pi(S)} \in \mathbb{R}^A$ are simplex vector such as

$$1^\top_{\pi(1)} = (\pi(a_1|s_1), \pi(a_2|s_1), ..., \pi(a_A|s_1)).$$

- The two matrices $P^V \in \mathbb{R}^{SA\times S}$, $\widehat{P}^V \in \mathbb{R}^{SA\times S}$ represent the probability transition kernel in the uncertainty set that leads to the worst-case value for any vector $V \in \mathbb{R}^S$. Moreover, the quantities $P^V_{s,a}$ (resp. $\widehat{P}^V_{s,a}$) stands for the (s,a)-th row of the transition matrix P^V (resp. $\widehat{P}^V$). In *sa*-rectangular case , the (s,a)-th rows of these transition matrices are defined as

$$P^V_{s,a} = \text{argmin}_{\mathcal{P}\in\mathcal{U}^{\text{sa},\sigma}(P^0_{s,a})}\mathcal{P}V, \qquad \text{and} \qquad \widehat{P}^V_{s,a} = \text{argmin}_{\mathcal{P}\in\mathcal{U}^{\text{sa},\sigma}(\widehat{P}^0_{s,a})}\mathcal{P}V. \tag{A.74a}$$

  Moreover, the shorthand notation defined below is used

$$P^{\pi,V}_{s,a} := P^{V^{\pi,\sigma}}_{s,a} = \text{argmin}_{\mathcal{P}\in\mathcal{U}^{\text{sa},\sigma}(P^0_{s,a})}\mathcal{P}V^{\pi,\sigma}, \tag{A.74b}$$

$$P^{\pi,\widehat{V}}_{s,a} := P^{\widehat{V}^{\pi,\sigma}}_{s,a} = \text{argmin}_{\mathcal{P}\in\mathcal{U}^{\text{sa},\sigma}(P^0_{s,a})}\mathcal{P}\widehat{V}^{\pi,\sigma}, \tag{A.74c}$$

$$\widehat{P}^{\pi,V}_{s,a} := \widehat{P}^{V^{\pi,\sigma}}_{s,a} = \text{argmin}_{P\in\mathcal{U}^{\text{sa},\sigma}(\widehat{P}^0_{s,a})}PV^{\pi,\sigma}, \tag{A.74d}$$

$$\widehat{P}^{\pi,\widehat{V}}_{s,a} := \widehat{P}^{\widehat{V}^{\pi,\sigma}}_{s,a} = \text{argmin}_{P\in\mathcal{U}^{\text{sa},\sigma}(\widehat{P}^0_{s,a})}P\widehat{V}^{\pi,\sigma}. \tag{A.74e}$$

  In the following, we define the corresponding probability transition matrices which are denoted by $P^{\pi,V} \in \mathbb{R}^{SA\times S}$, $P^{\pi,\widehat{V}} \in \mathbb{R}^{SA\times S}$, $\widehat{P}^{\pi,V} \in \mathbb{R}^{SA\times S}$ and $\widehat{P}^{\pi,\widehat{V}} \in \mathbb{R}^{SA\times S}$.

- Using the projection over π, the matrices $P^\pi \in \mathbb{R}^{S\times S}$, $\widehat{P}^\pi \in \mathbb{R}^{S\times S}$, $\underline{P}^{\pi,V} \in \mathbb{R}^{S\times S}$, $\underline{P}^{\pi,\widehat{V}} \in \mathbb{R}^{S\times S}$, $\underline{\widehat{P}}^{\pi,V} \in \mathbb{R}^{S\times S}$ and $\underline{\widehat{P}}^{\pi,\widehat{V}} \in \mathbb{R}^{S\times S}$ represent probability transition matrices w.r.t. policy π.

$$P^\pi := \Pi^\pi P^0, \qquad \widehat{P}^\pi := \Pi^\pi \widehat{P}^0, \qquad \underline{P}^{\pi,V} := \Pi^\pi P^{\pi,V}, \qquad \underline{P}^{\pi,\widehat{V}} := \Pi^\pi P^{\pi,\widehat{V}},$$
$$\underline{\widehat{P}}^{\pi,V} := \Pi^\pi \widehat{P}^{\pi,V}, \qquad \text{and} \qquad \underline{\widehat{P}}^{\pi,\widehat{V}} := \Pi^\pi \widehat{P}^{\pi,\widehat{V}}. \tag{A.75}$$

  For s-rectangular, we will use the same notation for these transition matrices. Finally, we denote P^π_s as the s-th row of the transition matrix P^π.

- $r_\pi \in \mathbb{R}^S$ is the reward function restricted to the actions chosen by π, $r_\pi = \Pi^\pi r$.

- $\text{Var}_P(V) \in \mathbb{R}^{SA}$ is the variance for a given transition kernel $P \in \mathbb{R}^{SA\times S}$ and vector $V \in \mathbb{R}^S$, we denote the (s,a)-th row of $\text{Var}_P(V)$ as

$$\mathsf{Var}_P(s,a) := \text{Var}_{P_{s,a}}(V). \tag{A.76}$$

6.1 Additional definitions and basic facts

For any norm smooth $\|.\|$ introduced in 3.2.1, we define the span semi norm as

Definition 6.1 (Span semi norm)**.** *Given any norm $\|\cdot\|$, we define the span semi norm as:* $\text{sp}(x) = \min_{\omega\in\mathbb{R}}\|v - \omega\mathbf{1}\|$ *and the generalized mean as* $\omega(x) := \arg\min_{\omega\in\mathbb{R}}\|x - \omega\mathbf{1}\|$.

Let vector $P \in \mathbb{R}^{1\times S}$ and vector $V \in \mathbb{R}^S$, we define the variance

$$\text{Var}_P(V) := P(V \circ V) - (PV) \circ (PV). \tag{A.77}$$

The following lemma bounds the Lipschitz constant of the variance function.

Lemma 6.1. *(Shi et al. (2023) , Lemma 2) Assuming* $0 \le V_1, V_2 \le \frac{1}{1-\gamma}$ *which obey* $\|V_1 - V_2\|_\infty \le x$*, then for* $P \in \Delta(S)$*, one has*

$$|\text{Var}_P(V_1) - \text{Var}_P(V_2)| \le \frac{2x}{(1-\gamma)}. \tag{A.78}$$

Lemma 6.2. *(Panaganti and Kalathil 2022b, Lemma 6) Consider any $\delta \in (0,1)$. For any fixed policy π and fixed value vector $V \in \mathbb{R}^S$, one has with probability at least $1-\delta$,*

$$\left|\sqrt{\mathrm{Var}_{\widehat{P}^\pi}(V)} - \sqrt{\mathrm{Var}_{P^\pi}(V)}\right| \leq \sqrt{\frac{2\|V\|_\infty^2 \log\left(\frac{2SA}{\delta}\right)}{N}} 1.$$

6.2 Empirical robust MDP $\widehat{\mathcal{M}}_{\mathsf{rob}}$ Bellman equations

We define the robust MDP $\widehat{\mathcal{M}}_{\mathsf{rob}} = \{\mathcal{S}, \mathcal{A}, \gamma, \mathcal{U}^\sigma(\widehat{P}^0), r\}$ based on the estimated nominal distribution $\widehat{P}^0$ in (3.11). Then, we denote the associated robust value function (resp. robust Q-function) are $\widehat{V}^{\pi,\sigma}$ (resp. $\widehat{Q}^{\pi,\sigma}$) qnd we can notice that that $\widehat{Q}^{\star,\sigma}$ is the unique-fixed point of $\widehat{\mathcal{T}}^\sigma(\cdot)$ (see Lemma 6.3), the empirical robust Bellman operator constructed using $\widehat{P}^0$. Finally, similarly to (3.9), for $\widehat{\mathcal{M}}_{\mathsf{rob}}$, the Bellman's optimality principle gives the following *robust Bellman consistency equation* (resp. *robust Bellman optimality equation*) for *sa*-rectangular assumptions:

$$\widehat{Q}^{\pi,\sigma}(s,a) = r(s,a) + \gamma \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}(\widehat{P}^0_{s,a})} \mathcal{P}\widehat{V}^{\pi,\sigma}, \tag{A.79a}$$

$$\widehat{Q}^{\star,\sigma}(s,a) = r(s,a) + \gamma \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}(\widehat{P}^0_{s,a})} \mathcal{P}\widehat{V}^{\star,\sigma}. \tag{A.79b}$$

Using matrix notation, we can write the robust Bellman consistency equations as

$$Q^{\pi,\sigma} = r + \gamma \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}(P^0)} \mathcal{P}V^{\pi,\sigma} \quad \text{and} \quad \widehat{Q}^{\pi,\sigma} = r + \gamma \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}(\widehat{P}^0)} \mathcal{P}\widehat{V}^{\pi,\sigma}, \tag{A.80}$$

which imply

$$\begin{aligned} V^{\pi,\sigma} &= r_\pi + \gamma \Pi^\pi \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}(P^0)} \mathcal{P}V^{\pi,\sigma} \overset{\text{(i)}}{=} r_\pi + \gamma \underline{P}^{\pi,V} V^{\pi,\sigma}, \\ \widehat{V}^{\pi,\sigma} &= r_\pi + \gamma \Pi^\pi \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}(\widehat{P}^0)} \mathcal{P}\widehat{V}^{\pi,\sigma} \overset{\text{(ii)}}{=} r_\pi + \gamma \underline{\widehat{P}}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma}, \end{aligned} \tag{A.81}$$

where (i) and (ii) hold by the definitions in (A.73), (A.74) and (A.75). For *s*-rectangular, we can define the same notation, removing a subscript:

$$\begin{aligned} V^{\pi,\sigma} &= r_\pi + \gamma \Pi^\pi \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{s},\sigma}(P^0)} \mathcal{P}V^{\pi,\sigma} \overset{\text{(i)}}{=} r_\pi + \gamma \underline{P}^{\pi,V} V^{\pi,\sigma}, \\ \widehat{V}^{\pi,\sigma} &= r_\pi + \gamma \Pi^\pi \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{s},\sigma}(\widehat{P}^0)} \mathcal{P}\widehat{V}^{\pi,\sigma} \overset{\text{(ii)}}{=} r_\pi + \gamma \underline{\widehat{P}}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma}, . \end{aligned} \tag{A.82}$$

6.3 Properties of the robust Bellman operator and dual representation

The robust Bellman operator (cf. (3.10)) shares the γ-contraction property of the standard Bellman operator as:

(Iyengar 2005, Theorem 3.2) Given $\gamma \in [0,1)$, the robust Bellman operator $\mathcal{T}^\sigma(\cdot)$ (cf. (3.10)) is a γ-contraction w.r.t. $\|\cdot\|_\infty$. More formally, for any $Q_1, Q_2 \in \mathbb{R}^{SA}$ s.t. $Q_1(s,a), Q_2(s,a) \in [0, \frac{1}{1-\gamma}]$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, one has

$$\|\mathcal{T}^\sigma(Q_1) - \mathcal{T}^\sigma(Q_2)\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty . \tag{A.83}$$

It can be also shown that, $Q^{\star,\sigma}$ is the unique fixed point of $\mathcal{T}^{\sigma}(\cdot)$ obeying $0 \leq Q^{\star,\sigma}(s,a) \leq \frac{1}{1-\gamma}$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$.

One of the main contributions is to derive the dual form of optimization problem using arbitrary norms. These lemma take ideas from Iyengar (2005) and are adapted to arbitrary norms and not only TV distance.

Dual equivalence of the robust Bellman operator. Fortunately, the robust Bellman operator can be evaluated efficiently by resorting to its dual formulation, and this idea is central in all proofs for RMPDs. Dual formulation of RMDPs have been introduced in (Iyengar 2005) but the proof was done uniquely for the TV and the χ^2 case. Before continuing, for any $V \in \mathbb{R}^S$, we denote $[V]_\alpha$ as its clipped version by some non-negative vector α, namely,

$$[V]_\alpha(s) \coloneqq \begin{cases} \alpha, & \text{if } V(s) > \alpha(s), \\ V(s), & \text{otherwise.} \end{cases} \tag{A.84}$$

Defining the gradient of $P \mapsto \|P\|$ as $\nabla\|P\|$, $\lambda > 0$, a positive scalar and ω is the generalized mean defined as the argmin in the definition of the span semi norm in Def.6.1, we derive two optimization lemmas.

Lemma 6.3 (Strong duality using norm $\|\cdot\|$ in the *sa*-rectangular case.)**.** *Consider any probability vector $P \in \Delta(\mathcal{S})$ and any fixed uncertainty level σ, we abbreviate the notation of the uncertainty set $\mathcal{U}^{\mathsf{sa},\sigma}_{\|\cdot\|}(P)$ (cf. (3.3)) as $\mathcal{U}^{\mathsf{sa},\sigma}(P)$. For any vector $V \in \mathbb{R}^S$ obeying $V \geq 0$, recalling the definition of $[V]_\alpha$ in (A.84), one has*

$$\inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}(P)} \mathcal{P}V = \max_{\mu_P^{\lambda,\omega} \in \mathcal{M}_P^{\lambda,\omega}} \left\{ P(V - \mu_P^{\lambda,\omega}) - \sigma \left(\mathrm{sp}((V - \mu_P^{\lambda,\omega}))_* \right) \right\}. \tag{A.85}$$

$$= \max_{\alpha_P^{\lambda,\omega} \in \mathrm{A}_P^{\lambda,\omega}} \left\{ P\,[V]_{\alpha_P^{\lambda,\omega}} - \sigma \left(\mathrm{sp}([V]_{\alpha_P^{\lambda,\omega}})_* \right) \right\} \tag{A.86}$$

where $\mathrm{sp}()_*$ is defined in Def..6.1. Here, the two auxiliary variational family $\mathrm{A}_P^{\lambda,\omega}, \mathcal{M}_P^{\lambda,\omega}$ are defined as below:

$$\mathrm{A}_P^{\lambda,\omega} = \{\alpha_P^{\lambda,\omega} : \alpha_P^{\lambda,\omega}(s) = \omega + \lambda|\nabla\|P\|(s) : \lambda > 0, w > 0, P \in \Delta(S), \alpha_P^{\lambda,\omega} \in \left[0, \frac{1}{1-\gamma}\right]^S \} \tag{A.87}$$

$$\mathcal{M}_P^{\lambda,\omega} = \{\mu_P^{\lambda,\omega} = V - \alpha_P^{\lambda,\omega}, \lambda, \omega \in \mathbb{R}^+, P \in \Delta(S), \mu \in \mathbb{R}_+^S, \mu_P^{\lambda,\omega} = \left[0, \frac{1}{1-\gamma}\right]^S \}. \tag{A.88}$$

$$\tag{A.89}$$

For L_1 or TV, case , the vector $\alpha_P^{\lambda,\omega}$ reduces to a 1 dimensional scalar such as $\alpha \in [0, 1/(1-\gamma)]$.

Proof.

$$\begin{aligned} \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}(P)} \mathcal{P}V &= \inf_{\{\mathcal{P}: \mathcal{P} \in \Delta_s, \|\mathcal{P} - P\| \leq \sigma\}} \sum_{s'} \mathcal{P}(s')V(s') \\ &= PV + \inf_{\{y: \|y\| \leq \sigma, 1y = 0, y \geq -P\}} \sum_{s'} y(s')V(s') \end{aligned}$$

where we use the change of variable $y(s') = \mathcal{P}(s') - P(s')$ for all $s' \in \mathcal{S}$. Then the Lagrangian function of the above optimization problem can be written as follows:

$$\inf_{\mathcal{P}\in\mathcal{U}^{\sigma}_{s,a}(P)} \mathcal{P}V = PV + \sup_{\mu\geq 0,\nu\in\mathbb{R}} \inf_{\{y:\|y\|\leq\sigma\}} -\sum_{s'}\mu(s)P(s') + \sum_{s'}(y(s')(V(s') - \mu(s') - \nu) \tag{A.90}$$

$$\overset{(a)}{=} PV + \sup_{\mu\geq 0,\nu\in\mathbb{R}} -\sum_{s'}\mu(s')P(s') - \sigma\left\|(V(s') - \mu(s') - \nu\mathbf{1})\right\|_* \tag{A.91}$$

$$\overset{(b)}{=} \sup_{\mu\geq 0} P(V-\mu) - \sigma \mathrm{sp}(V-\mu)_* \tag{A.92}$$

where $\mu \in \mathbb{R}^{\mathcal{S}}_+$, $\nu \in \mathbb{R}$ are Lagrangian variables, (a) is true using the equality case of Cauchy-Swartz inequality for dual norm Yang (1991), and (b) is due to is the definition of the span semi-norm (see (6)). The value that maximizes the inner maximization problem in (A.91) in $\omega(V,\mu)$ is the generalized-mean by definition denoted with abbreviate notation ω. If the norm is differentiable, then we have that the equality (a) comes from the generalized Holder's inequality for arbitrary norms Yang (1991), namely, defining $z = (V - \mu - \omega)$, it satisfies

$$z = \|z\|_* \nabla\|y\| \tag{A.93}$$

The quantity ν is replaced by the generalized mean for equality in (b) while (A.93) comes from Yang (1991). Using complementary slackness Karush (2013)stackness let $\mathcal{B} = \{s \in \mathcal{S} : \mu(s) > 0\}$

$$\forall s \in \mathcal{B}: \quad y^*(s) = -P(s), \tag{A.94}$$

which leads to the following equality by plugging the previous (A.94) in (A.93) and defining $z^* = V - \mu^* - \omega$:

$$\forall s \in \mathcal{B}, \quad z^*(s) = \|z^*\|_* \nabla\|P\|(s) \tag{A.95}$$

or

$$\forall s \in \mathcal{B}, \quad V(s) - \mu^*(s) = \omega + \lambda\nabla\|P\|(s) \hat{=} \alpha_P^{\lambda,\omega} \tag{A.96}$$

by letting $\lambda = \|z^*\|_* \in \mathbb{R}^+$. Note that here the hypothesis of 3.2.1 are use and especially separability is needed to ensure that for $s \in \mathcal{B}$, $\nabla\|y\| = \nabla\|P\|$ only depend on $P(s)$ and not on other coordinates, which is true form generalized L_p norms. We can remark that $v - \mu^*$ is P dependent, but if P is known, the best μ^* is only determined by one 2 dimensional parameters $\lambda = \|v - \mu^* - \nu\|_*$ and $\omega \in \mathbb{R}^+$. Moreover, when P is fixed, the scalar ω is a constant is fully determined by P, v and μ^*. This is why the quantity defined α_P^{λ} varies through 2 parameter λ and ω. Given this observation, we can rewrite the optimization problem as :

$$\sup_{\mu\geq 0} P(V-\mu) - \sigma\mathrm{sp}(V-\mu)_* = \sup_{\mu_P^{\lambda,\omega}\in\mathcal{M}_P^{\lambda,\omega}} P(V-\mu_P^{\lambda,\omega}) - \sigma\mathrm{sp}((V-\mu_P^{\lambda,\omega}))_* \tag{A.97}$$

$$= \sup_{\alpha_P^{\lambda,\omega}\in \mathrm{A}_P^{\lambda,\omega}} P[V]_{\alpha_P^{\lambda,\omega}} - \sigma\mathrm{sp}([V]_{\alpha_P^{\lambda,\omega}})_* \tag{A.98}$$

where we defined the maximization problem on μ not in $\mathbb{R}^S$ but at the optimal in the variational family denote $\mathcal{M}_P^{\lambda,\omega} = \{v - \alpha_P^{\lambda,\omega}, (\lambda,\omega) \in \mathbb{R}^2_+, P \in \Delta(S)\}$. We can rewrite the optimization problem in terms of α_P with $[V]_{\alpha_P^{\lambda,\omega}}$ defined in A.84. Contrary to the TV case, α is not a scalar but $\alpha_P^{\lambda,\omega}$ belongs to a variational family only determined by two parameter. Note that this lemma is still true writing subgradient and not gradient of P. As we assume C^2-regularity on norms, the subgradient space of the norm reduce to the singleton of the gradient in our case. C^2 smoothness will be needed in concentration part while it is possible to be more general in optimization lemmas. Note that for TV or L_1, this lemma holds, but the vector $\alpha_P^{\lambda,\omega}$ reduces to a positive scalar denoted α which is equal to $\|v - \mu^*\|_\infty$ according to Iyengar (2005).

□

Lemma 6.4 (Strong duality for the distance induced by the norm $\|\|\|$ in the s-rectangular case.)**.** *Consider any probability vector $P^\pi := \Pi^\pi P \in \Delta_s$ for $P \in \Delta(S)^{\mathcal{A}}$, any fixed uncertainty level $\tilde{\sigma}$ and the uncertainty set $\mathcal{U}^{\mathbf{s},\widetilde{\sigma}}_{\|.\|}(P)$, we abbreviate the subscript to use $\mathcal{U}^{\mathbf{s},\widetilde{\sigma}}(P) := \mathcal{U}^{\mathbf{s},\widetilde{\sigma}}_{\|.\|}(P)$. Then for any vector $V \in \mathbb{R}^S$ obeying $V \geq 0$, recalling the definition of $[V]_\alpha$ in (A.84), one has*

$$\inf_{\mathcal{P}\in\mathcal{U}^{\mathbf{s},\widetilde{\sigma}}(P)} \mathcal{P}^\pi V = \sum_a \pi(a|s)\Big(\max_{\alpha^{\lambda,\omega}_{P_{sa}} \in \mathrm{A}^{\lambda,\omega}_{P_{sa}}} P_{sa}[V]_{\alpha^{\lambda,\omega}_{P_{sa}}} - \tilde{\sigma}\,\|\pi_s\|_*\,\mathrm{sp}([V]_{\alpha^{\lambda,\omega}_{P_{sa}}})_*\Big). \tag{A.99}$$

with the definition of $\mathrm{sp}()_*$ *in 6 and where the variational family* $\mathrm{A}^{\lambda,\omega}_P$ *is defined as :*

$$\mathrm{A}^{\lambda,\omega}_P = \{\alpha \in \Big[0, 1/(1-\gamma)\Big]^S, \alpha = \omega + \lambda|\nabla\|P\|\| := \alpha^{\lambda,\omega}_P\} \tag{A.100}$$

$$\tag{A.101}$$

with ω is the generalized mean defined as the argmin in the definition of the span semi norm in 6.1 and λ, ω a positive scalar. Moreover, for L_1 or TV, case, the vector $\alpha^{\lambda,\omega}_P$ reduces to a 1 dimensional scalar such as $\alpha \in [0, 1/(1-\gamma)]$.

In the proof of the previous lemma, we decompose this problem s-rectangular radius $\tilde{\sigma}$ into sa-rectangular sub-problem with respectively radius σ_{sa}.

Proof.

$$\begin{aligned}
\inf_{\mathcal{P}^\pi\in\mathcal{U}^{\mathbf{s},\widetilde{\sigma}}(P^\pi)} \mathcal{P}^\pi V &= \inf_{\{\sigma_{sa}:\|\sigma_{sa}\|\leq\tilde{\sigma}\}} \inf_{\mathcal{P}'\in\mathcal{U}^{\mathbf{sa},\sigma}(P_{sa})} \sum_a \pi(a|s)\mathcal{P}'V \\
&\overset{(a)}{=} \sum_a \pi(a|s)P_{sa}V + \min_{\{\sigma_{sa}:\|\sigma_{sa}\|\leq\tilde{\sigma}\}} \sum_a \pi(a|s) \min_{\{y:\|y\|\leq\sigma_{sa}, 1y=0, y\geq -P_{sa}\}} \sum_{s'} y(s')V
\end{aligned}$$

where we use the change of variable $y(s') = \mathcal{P}_{sa}(s') - P_{sa}(s')$ in (a). Then we case use the previous lemma for sa rectangular assumption, Lemma 6.3. Then,

$$\begin{aligned}
&\min_{\{\sigma_{sa}:\|\sigma_{sa}\|\leq\tilde{\sigma}\}} \sum_a \pi(a|s) \min_{\{y,\|y\|\leq\sigma_{s,a}, 1y=0, y\geq -P_{sa}\}} \sum_{s'} y(s')V \\
&= \min_{\{\sigma_{sa}:\|\sigma_{sa}\|\leq\tilde{\sigma}\}} \sum_a \pi(a|s) \max_{\mu\geq 0} \Big(- P_{sa}\mu - \sigma_{sa}\mathrm{sp}(V-\mu)_*\Big) \\
&= \left(\sum_a \pi(a|s) \max_{\mu\geq 0} \Big\{(-P_{sa}\mu) - \max_{\{\sigma_{sa}:\|\sigma_{sa}\|\leq\tilde{\sigma}\}} \sum_a \pi(a|s)\sigma\mathrm{sp}(V-\mu)_*\Big\}\right) \\
&= \sum_a \pi(a|s) \max_{\mu\geq 0} \Big\{(-P_{sa}\mu) - \tilde{\sigma}\,\|\pi_s\|_*\,\mathrm{sp}(V-\mu)_*\Big\}.
\end{aligned}$$

We can exchange the min and the max as we get concave-convex problems in σ and μ in the second line according to minimax theorem (v. Neumann 1928) and using Cauchy Swartz inequality which is attained in the last equality. Finally, we obtain:

$$\begin{aligned}
\inf_{\mathcal{P}\in\mathcal{U}^{\mathbf{s},\widetilde{\sigma}}(P)} \mathcal{P}^\pi V &= \sum_a \pi(a|s)\Big(\max_{\mu\geq 0} P_{sa}(V-\mu) - \tilde{\sigma}\,\|\pi_s\|_*\,\mathrm{sp}(V-\mu)_*\Big) \\
&\overset{(a)}{=} \sum_a \pi(a|s)\Big(\max_{\alpha^{\lambda,\omega}_{P_{sa}} \in \mathrm{A}^{\lambda,\omega}_{P_{sa}}} P_{sa}[V]_{\alpha^{\lambda,\omega}_{P_{sa}}} - \tilde{\sigma}\,\|\pi_s\|_*\,\mathrm{sp}([V]_{\alpha^{\lambda,\omega}_{P_{sa}}})_*\Big)
\end{aligned}$$

where in (a) we use the previous lemma for $sa-$ rectangular case. Note that as we are using sa-rectangular case, for TV or L_1, this lemma holds, but the vector α_P^λ reduces to a positive scalar denoted α which is equal to $\|v - \mu^*\|_\infty$. (See also Iyengar (2005)).

□

7 Proof of the upper bound : Theorem 3.4.1 and 3.4.3

7.1 Technical lemmas

We begin with a key lemma concerning the dynamic range of the robust value function $V^{\pi,\sigma}$ (cf. (3.7)), which produces tighter control when σ is large; the proof is deferred to Appendix 7.3.1. This lemma allows tighter control compared to Clavier et al. (2023).

Lemma 7.1. *In $sa-$rectangular case (see (3.3), for any nominal transition kernel $P \in \mathbb{R}^{SA\times S}$, any fixed uncertainty level σ, and any policy π, its corresponding robust value function $V^{\pi,\sigma}$ (cf. (3.7)) satisfies*

$$\mathrm{sp}(V^{\pi,\sigma})_\infty \leq \frac{1}{\gamma \max\{1-\gamma, C_g\sigma\}} \tag{A.102}$$

where $C_g = 1/(\min_s \|e_s\|)$ is a geometric constant depending on the geometry of the norm. For example, for L_p, norms $p \geq 1$, $C_g \geq 1$ which reduce the sample complexity. In s-rectangular case, we obtain a slightly different lemma because of the dependency on π.

Lemma 7.2. *The infinite span semi norm can be controlled as follows for every s in s-rectanuglar case (See (3.5)):*

$$\mathrm{sp}(V^{\pi,\sigma})_\infty \leq \frac{1}{\gamma \max\{1-\gamma, \|\pi_s\|_* C_g\tilde{\sigma}\}} \leq \frac{1}{\gamma \max\{1-\gamma, \min_s \|\pi_s\|_* C_g\tilde{\sigma}\}} \tag{A.103}$$

where $C_g = \frac{1}{\min_s \|e_s\|}$ is a geometric constant depending on the geometry of the norm. These lemmas are required to get tight bounds for the sample complexity. The main difference between sa- and s- rectangular case is that we have an extra dependency on $\|\pi_s\|_*$, which represents how stochastic the policy can be in s rectangular MDPs.

Lemma 7.3. *Consider an MDP with transition kernel matrix P and reward function $0 \leq r \leq 1$. For any policy π and its associated state transition matrix $P_\pi := \Pi^\pi P$ and value function $0 \leq V^{\pi,P} \leq \frac{1}{1-\gamma}$ (cf. (3.1)), one has for sa- and s- rectangular assumptions.*

$$(I - \gamma P_\pi)^{-1} \sqrt{\mathrm{Var}_{P_\pi}(V^{\pi,P})} \leq \sqrt{\frac{8}{\gamma^2(1-\gamma)^2}\mathrm{sp}(V^{\pi,P})_\infty}1.$$

See 7.3.7 for the proof

7.2 Proof of Theorem 3.4.1 and Theorem 3.4.3

The first decomposition of the proof of Theorem 3.4.1 and Theorem 3.4.3 Agarwal et al. (2020) while the argument needs essential adjustments in order to adapt to the robustness setting. One has by assumptions using any planner in empirical RMDPs :

$$\|\widehat{V}^{\star,\sigma} - \widehat{V}^{\widehat{\pi},\sigma}\|_\infty \leq \varepsilon_{\mathsf{opt}}, \tag{A.104}$$

using previous inequality, performance gap $\left\|V^{\star,\sigma} - V^{\widehat{\pi},\sigma}\right\|_\infty$, can be upper bounded using 3 steps.

First step: subdivide the performance gap in 3 terms. We recall the definition of the optimal robust policy $\pi^\star$ with regard to $\mathcal{M}_{\mathsf{rob}}$ and the optimal robust policy $\widehat{\pi}^\star$, the optimal robust value function $\widehat{V}^{\star,\sigma}$ (resp. robust value function $\widehat{Q}^{\pi,\sigma}$) w.r.t. $\widehat{\mathcal{M}}_{\mathsf{rob}}$. Then, the performance gap $V^{\star,\sigma} - V^{\widehat{\pi},\sigma}$ can be decomposed in one optimization term and two statistical error terms

$$\begin{aligned} V^{\star,\sigma} - V^{\widehat{\pi},\sigma} &= \left(V^{\pi^\star,\sigma} - \widehat{V}^{\pi^\star,\sigma}\right) + \left(\widehat{V}^{\pi^\star,\sigma} - \widehat{V}^{\widehat{\pi}^\star,\sigma}\right) + \left(\widehat{V}^{\widehat{\pi}^\star,\sigma} - \widehat{V}^{\widehat{\pi},\sigma}\right) + \left(\widehat{V}^{\widehat{\pi},\sigma} - V^{\widehat{\pi},\sigma}\right) \\ &\overset{\text{(i)}}{\leq} \left(V^{\pi^\star,\sigma} - \widehat{V}^{\pi^\star,\sigma}\right) + \left(\widehat{V}^{\widehat{\pi}^\star,\sigma} - \widehat{V}^{\widehat{\pi},\sigma}\right) + \left(\widehat{V}^{\widehat{\pi},\sigma} - V^{\widehat{\pi},\sigma}\right) \\ &\overset{\text{(ii)}}{\leq} \left(V^{\pi^\star,\sigma} - \widehat{V}^{\pi^\star,\sigma}\right) + \varepsilon_{\mathsf{opt}} + \left(\widehat{V}^{\widehat{\pi},\sigma} - V^{\widehat{\pi},\sigma}\right) \end{aligned} \tag{A.105}$$

where (i) holds by $\widehat{V}^{\pi^\star,\sigma} - \widehat{V}^{\widehat{\pi}^\star,\sigma} \leq 0$ since $\widehat{\pi}^\star$ is the robust optimal policy for $\widehat{\mathcal{M}}_{\mathsf{rob}}$, and (ii) comes from (A.104) and definition of optimization error. The proof aims to control the last remaining terms in (A.105) using concentration theory and sufficiently big number of step N. To do so, we will consider a more general term $\widehat{V}^{\pi,\sigma} - V^{\pi,\sigma}$ for any policy π even if control of these two terms slightly differ at the end. Using (A.81), it holds that for both *sa*- and *s*-rectangular assumptions:

$$\begin{aligned} \widehat{V}^{\pi,\sigma} - V^{\pi,\sigma} &= r_\pi + \gamma \underline{\widehat{P}}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma} - \left(r_\pi + \gamma \underline{P}^{\pi,V} V^{\pi,\sigma}\right) \\ &= \left(\gamma \underline{\widehat{P}}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma} - \gamma \underline{P}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma}\right) + \left(\gamma \underline{P}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma} - \gamma \underline{P}^{\pi,V} V^{\pi,\sigma}\right) \\ &\overset{\text{(i)}}{\leq} \gamma \left(\underline{P}^{\pi,V} \widehat{V}^{\pi,\sigma} - \underline{P}^{\pi,V} V^{\pi,\sigma}\right) + \left(\gamma \underline{\widehat{P}}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma} - \gamma \underline{P}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma}\right), \end{aligned}$$

where (i) holds because $\underline{P}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma} \leq \underline{P}^{\pi,V} \widehat{V}^{\pi,\sigma}$ because of the optimality of $\underline{P}^{\pi,\widehat{V}}$ (see. (A.74)). Factorizing terms leads to the following equation

$$\widehat{V}^{\pi,\sigma} - V^{\pi,\sigma} \leq \gamma \left(I - \gamma \underline{P}^{\pi,V}\right)^{-1} \left(\underline{\widehat{P}}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma} - \underline{P}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma}\right). \tag{A.106}$$

In the same manner, we can also obtain a lower bound of this quantity:

$$\begin{aligned} \widehat{V}^{\pi,\sigma} - V^{\pi,\sigma} &= r_\pi + \gamma \underline{\widehat{P}}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma} - \left(r_\pi + \gamma \underline{P}^{\pi,V} V^{\pi,\sigma}\right) \\ &= \left(\gamma \underline{\widehat{P}}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma} - \gamma \underline{P}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma}\right) + \left(\gamma \underline{P}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma} - \gamma \underline{P}^{\pi,V} V^{\pi,\sigma}\right) \\ &\geq \gamma \left(\underline{P}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma} - \underline{P}^{\pi,\widehat{V}} V^{\pi,\sigma}\right) + \left(\gamma \underline{\widehat{P}}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma} - \gamma \underline{P}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma}\right) \\ &\geq \gamma \left(I - \gamma \underline{P}^{\pi,\widehat{V}}\right)^{-1} \left(\underline{\widehat{P}}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma} - \underline{P}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma}\right). \end{aligned} \tag{A.107}$$

Using both (A.106) and (A.107), we obtain infinite norm control:

$$\begin{aligned} \|\widehat{V}^{\pi,\sigma} - V^{\pi,\sigma}\|_\infty \leq \gamma \max \Big\{ &\left\| \left(I - \gamma \underline{P}^{\pi,V}\right)^{-1} \left(\underline{\widehat{P}}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma} - \underline{P}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma}\right) \right\|_\infty, \\ &\left\| \left(I - \gamma \underline{P}^{\pi,\widehat{V}}\right)^{-1} \left(\underline{\widehat{P}}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma} - \underline{P}^{\pi,\widehat{V}} \widehat{V}^{\pi,\sigma}\right) \right\|_\infty \Big\}. \end{aligned} \tag{A.108}$$

By decomposing the error in a symmetric way, he have

$$\begin{aligned} \|\widehat{V}^{\pi,\sigma} - V^{\pi,\sigma}\|_\infty \leq \gamma \max \Big\{ &\left\| \left(I - \gamma \underline{\widehat{P}}^{\pi,V}\right)^{-1} \left(\underline{\widehat{P}}^{\pi,V} V^{\pi,\sigma} - \underline{P}^{\pi,V} V^{\pi,\sigma}\right) \right\|_\infty, \\ &\left\| \left(I - \gamma \underline{\widehat{P}}^{\pi,\widehat{V}}\right)^{-1} \left(\underline{\widehat{P}}^{\pi,V} V^{\pi,\sigma} - \underline{P}^{\pi,V} V^{\pi,\sigma}\right) \right\|_\infty \Big\}. \end{aligned} \tag{A.109}$$

Armed with these inequalities, we can use concentration inequalities to upper bound the two remaining terms $\|\widehat{V}^{\pi^\star,\sigma} - V^{\pi^\star,\sigma}\|_\infty$ and $\|\widehat{V}^{\widehat{\pi},\sigma} - V^{\widehat{\pi},\sigma}\|_\infty$ in (A.105). Taking $\pi = \widehat{\pi}$, applying (A.108) leads to

$$\|\widehat{V}^{\widehat{\pi},\sigma} - V^{\widehat{\pi},\sigma}\|_\infty \leq \gamma \max\Big\{\Big\|\Big(I - \gamma \underline{P}^{\widehat{\pi},\widehat{V}}\Big)^{-1}\Big(\underline{\widehat{P}}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma} - \underline{P}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma}\Big)\Big\|_\infty, \\ \Big\|\Big(I - \gamma \underline{P}^{\widehat{\pi},V}\Big)^{-1}\Big(\underline{\widehat{P}}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma} - \underline{P}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma}\Big)\Big\|_\infty\Big\}. \tag{A.110}$$

Finally, $\pi = \pi^\star$, applying (A.109) gives us

$$\|\widehat{V}^{\pi^\star,\sigma} - V^{\pi^\star,\sigma}\|_\infty \leq \gamma \max\Big\{\Big\|\Big(I - \gamma \underline{\widehat{P}}^{\pi^\star,V}\Big)^{-1}\Big(\underline{\widehat{P}}^{\pi^\star,V}V^{\pi^\star,\sigma} - \underline{P}^{\pi^\star,V}V^{\pi^\star,\sigma}\Big)\Big\|_\infty, \\ \Big\|\Big(I - \gamma \underline{\widehat{P}}^{\pi^\star,\widehat{V}}\Big)^{-1}\Big(\underline{\widehat{P}}^{\pi^\star,V}V^{\pi^\star,\sigma} - \underline{P}^{\pi^\star,V}V^{\pi^\star,\sigma}\Big)\Big\|_\infty\Big\}. \tag{A.111}$$

Note that to control $\|\widehat{V}^{\pi^\star,\sigma} - V^{\pi^\star,\sigma}\|_\infty$, we use decomposition not depending on $\widehat{\pi}$ for value function as $V^{\pi^\star,\sigma}$ is deterministic and fixed, allowing use of classical concentration analysis tools. This decomposition is the same for both *sa*-rectangular and *s*-rectangular case.

Second step: bound first term and second term in (A.111) to control $\|\widehat{V}^{\pi^\star,\sigma} - V^{\pi^\star,\sigma}\|_\infty$
To control the two terms in (A.111), we use lemma 7.4 based Bernstein's concentration argument and whose proof is in Appendix 7.3.3.

Lemma 7.4. *For both sa− and s-rectangular setting, consider any $\delta \in (0,1)$, with probability $1-\delta$, it holds:*

$$\Big|\underline{\widehat{P}}^{\pi^\star,V}V^{\pi^\star,\sigma} - \underline{P}^{\pi^\star,V}V^{\pi^\star,\sigma}\Big| \leq 2\sqrt{\frac{L}{N}}\sqrt{\mathrm{Var}_{P^{\pi^\star}}(V^{\star,\sigma})} + \frac{3LC_S\|1\|_*}{N(1-\gamma)}1 \tag{A.112}$$

with $L = 2\log(18\|1\|_* SAN/\delta)$ and where $\mathrm{Var}_{P^{\pi^\star}}(V^{\star,\sigma})$ is defined in (A.76). Moreover, for the specific case of TV, this lemma is true without the smoothness term $\frac{3LC_S\|1\|_*}{N(1-\gamma)}$.

Armed with the above lemma, now we control the **first term** on the right-hand side of (A.111) as follows:

$$\begin{aligned}
&\Big(I - \gamma \underline{\widehat{P}}^{\pi^\star,V}\Big)^{-1}\Big(\underline{\widehat{P}}^{\pi^\star,V}V^{\pi^\star,\sigma} - \underline{P}^{\pi^\star,V}V^{\pi^\star,\sigma}\Big) \\
&\overset{(a)}{\leq} \Big(I - \gamma \underline{\widehat{P}}^{\pi^\star,V}\Big)^{-1}\Big\|\underline{\widehat{P}}^{\pi^\star,V}V^{\pi^\star,\sigma} - \underline{P}^{\pi^\star,V}V^{\pi^\star,\sigma}\Big\|_\infty \\
&\overset{(b)}{\leq} \Big(I - \gamma \underline{\widehat{P}}^{\pi^\star,V}\Big)^{-1}\Big(2\sqrt{\frac{L}{N}}\sqrt{\mathrm{Var}_{P^{\pi^\star}}(V^{\star,\sigma})} + \frac{3LC_S\|1\|_*}{N(1-\gamma)}\Big) \\
&\leq \Big(I - \gamma \underline{\widehat{P}}^{\pi^\star,V}\Big)^{-1}\frac{3LC_S\|1\|_*}{N(1-\gamma)}1 + \underbrace{2\sqrt{\frac{L}{N}}\Big(I - \gamma \underline{\widehat{P}}^{\pi^\star,V}\Big)^{-1}\sqrt{\mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,V}}(V^{\star,\sigma})}}_{=:\mathcal{R}_1} \\
&\quad + \underbrace{2\sqrt{\frac{L}{N}}\Big(I - \gamma \underline{\widehat{P}}^{\pi^\star,V}\Big)^{-1}\sqrt{\Big|\mathrm{Var}_{\widehat{P}^{\pi^\star}}(V^{\star,\sigma}) - \mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,V}}(V^{\star,\sigma})\Big|}}_{=:\mathcal{R}_2} \\
&\quad + \underbrace{2\sqrt{\frac{L}{N}}\Big(I - \gamma \underline{\widehat{P}}^{\pi^\star,V}\Big)^{-1}\Big(\sqrt{\mathrm{Var}_{P^{\pi^\star}}(V^{\star,\sigma})} - \sqrt{\mathrm{Var}_{\widehat{P}^{\pi^\star}}(V^{\star,\sigma})}\Big)}_{=:\mathcal{R}_3},
\end{aligned} \tag{A.113}$$

where (a) holds as the matrix $\left(I-\gamma\underline{\widehat{P}}^{\pi^\star,V}\right)^{-1}$ is positive definite, (b) holds due to Lemma 7.4, and the last point holds from the following decomposition for variance and triangular inequality

$$\begin{aligned}
&\sqrt{\mathrm{Var}_{P^{\pi^\star}}(V^{\star,\sigma})}=\left(\sqrt{\mathrm{Var}_{P^{\pi^\star}}(V^{\star,\sigma})}-\sqrt{\mathrm{Var}_{\widehat{P}^{\pi^\star}}(V^{\star,\sigma})}\right)+\sqrt{\mathrm{Var}_{\widehat{P}^{\pi^\star}}(V^{\star,\sigma})}\\
&\leq\left(\sqrt{\mathrm{Var}_{P^{\pi^\star}}(V^{\star,\sigma})}-\sqrt{\mathrm{Var}_{\widehat{P}^{\pi^\star}}(V^{\star,\sigma})}\right)\\
&+\sqrt{\left|\mathrm{Var}_{\widehat{P}^{\pi^\star}}(V^{\star,\sigma})-\mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,V}}(V^{\star,\sigma})\right|}+\sqrt{\mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,V}}(V^{\star,\sigma})}.
\end{aligned}$$

Finally, the fact that $\underline{\widehat{P}}^{\pi^\star,V}$ is a stochastic matrix, so

$$\left(I-\gamma\underline{\widehat{P}}^{\pi^\star,V}\right)^{-1}1=\left(I+\sum_{t=1}^{\infty}\gamma^t\left(\underline{\widehat{P}}^{\pi^\star,V}\right)^t\right)1\leq\frac{1}{1-\gamma}1. \tag{A.114}$$

Armed with these inequalities, the three terms $\mathcal{R}_1,\mathcal{R}_2,\mathcal{R}_3$ in (A.113) can be controlled separately.

- Consider $\mathcal{R}_1$. We first introduce the following lemma, whose proof is postponed to Appendix 7.3.4.

  Lemma 7.5. *Consider any $\delta\in(0,1)$. With probability at least $1-\delta$, one has*

  $$\begin{aligned}
  &\left(I-\gamma\underline{\widehat{P}}^{\pi^\star,V}\right)^{-1}\sqrt{\mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,V}}(V^{\star,\sigma})}\leq 4\sqrt{\frac{\left(1+\left(\sqrt{\frac{L}{(1-\gamma)^2N}}+\frac{C_S\|1\|_*L}{N(1-\gamma)}\right)\right)}{\gamma^3(1-\gamma)^2\max\{1-\gamma,C_g\sigma\}}}1\\
  &\leq 4\sqrt{\frac{\left(1+\left(\sqrt{\frac{L}{(1-\gamma)^2N}}+\frac{C_S\|1\|_*L}{N(1-\gamma)}\right)\right)}{\gamma^3(1-\gamma)^3}}1
  \end{aligned}$$

  with $L=2\log\left(\frac{18\|1\|_*SAN}{\delta}\right)$ in the sa-rectangular case. In the s-rectangular case, it holds:

  $$\begin{aligned}
  &\left(I-\gamma\underline{\widehat{P}}^{\pi^\star,V}\right)^{-1}\sqrt{\mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,V}}(V^{\star,\sigma})}\leq \quad 4\sqrt{\frac{\left(1+\left(\sqrt{\frac{L}{(1-\gamma)^2N}}+\frac{C_S\|1\|_*L}{N(1-\gamma)}\right)\right)}{\gamma^3(1-\gamma)^2\max\{1-\gamma,C_g\tilde{\sigma}\min_s\|\pi_s\|_*\}}}1\\
  &\leq 4\sqrt{\frac{\left(1+\left(\sqrt{\frac{L}{(1-\gamma)^2N}}+\frac{C_S\|1\|_*L}{N(1-\gamma)}\right)\right)}{\gamma^3(1-\gamma)^3}}1
  \end{aligned}$$

  Using Lemma 7.5 and inserting back to (A.113) gives in *sa*-rectangular case

  $$\begin{aligned}
  \mathcal{R}_1&=2\sqrt{\frac{L}{N}}\left(I-\gamma\underline{\widehat{P}}^{\pi^\star,V}\right)^{-1}\sqrt{\mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,V}}(V^{\star,\sigma})}\\
  &\leq 8\sqrt{\frac{L}{\gamma^3(1-\gamma)^2\max\{1-\gamma,C_g\sigma\}N}\left(1+\sqrt{\frac{L}{(1-\gamma)^2N}}+\frac{C_S\|1\|_*L}{N(1-\gamma)}\right)}1.
  \end{aligned} \tag{A.115}$$

- Consider $\mathcal{R}_2$. First, denote $V'\coloneqq V^{\star,\sigma}-\eta 1\ \eta\in\mathbb{R}$, by Lemma 7.1, we have for any π,

  $$0\leq\min_\eta\|V-\eta 1\|_\infty\leq\frac{1}{\gamma\max\{1-\gamma,C_g\sigma\}} \tag{A.116}$$

for sa-rectangular case or in s-rectangular we obtain

$$0 \leq \min_{\eta} \|V - \eta 1\|_\infty \leq \frac{1}{\gamma \max\{1-\gamma, \tilde{\sigma} C_g \|\pi_s\|_*\}} \tag{A.117}$$

by the definition of the span semi norm. Moreover, we can use Holder with L_1 and L_∞ we have for both sa and s-rectangular case to as it holds that:

$$\begin{aligned}
&|\mathsf{Var}_{\widetilde{P}_{s,a}}(V^{\star,\sigma}) - \mathsf{Var}_{P_{s,a}}(V^{\star,\sigma})| = |\mathsf{Var}_{\widetilde{P}_{s,a}}(V') - \mathsf{Var}_{P_{s,a}}(V')| \\
&\leq \|\widetilde{P}_{s,a} - P_{s,a}\|_1 \|V'\|_\infty^2 \overset{a}{\leq} \frac{\sigma_1}{(\gamma^2(\max(1-\gamma), C_g\sigma)^2} \\
&\leq \frac{1}{\gamma^2 \max\{(1-\gamma), \sigma C_g\}}.
\end{aligned} \tag{A.118}$$

In the first inequality, we use $\|V'\|_\infty^2 = \|V'^2\|_\infty$ and and we use Lemma 7.1 in (a) where $C_g\sigma = \sigma_1$.

With the same arguments for s-rectangular, we obtain for $V' \coloneqq V^{\star,\sigma} - \eta 1$, $\eta \in \mathbb{R}$,

$$|\Pi^{\pi^\star}\Big(\mathsf{Var}_{\widetilde{P}_s}(V^{\star,\sigma}) - \mathsf{Var}_{P_s}(V^{\star,\sigma})\Big)| = |\Pi^{\pi^\star}\Big(\mathsf{Var}_{\widetilde{P}_s}(V') - \mathsf{Var}_{P_s}(V')\Big)|$$

$$\leq |\sum_a \pi^*(a|s) \sum_{s'} (\widetilde{P}_s(s',a) - P_s(s',a)) V'(s')^2| \tag{A.119}$$

$$\leq \|V'\|_\infty^2 \sum_a \sum_{s'} \pi^*(a|s)(\widetilde{P}_s(s',a) - P_s(s',a)) \overset{a}{\leq} \|V'\|_\infty^2 \tilde{\sigma} \|\pi_s^*\|_* C_g^s 1 \tag{A.120}$$

$$\overset{b}{\leq} \frac{\tilde{\sigma} C_g^s \|\pi_s^*\|_* \|V'\|_\infty}{\gamma \|\pi_s^*\|_* \tilde{\sigma} C_g^s} 1 \leq \frac{\|V'\|}{\gamma} 1. \tag{A.121}$$

where (a) comes Eq A.175, (b) comes lemma 7.2 or more precisely eq (A.188). Then, taking the sup over s in the previous equations, it holds

$$|\Pi^{\pi^\star}\Big(\mathsf{Var}_{\widetilde{P}_s}(V^{\star,\sigma}) - \mathsf{Var}_{P_s}(V^{\star,\sigma})\Big)| \leq \frac{\inf_{\eta \in \mathbb{R}^+} \|V - \eta 1'\|}{\gamma} 1 \tag{A.122}$$

$$\leq \frac{1}{\gamma^2 \tilde{\sigma} \min_s \|\pi_s^*\|_* C_g} 1. \tag{A.123}$$

Applying the previous inequality, it holds in sa-rectangular case:

$$\begin{aligned}
\mathcal{R}_2 &= 2\sqrt{\frac{L}{N}}\Big(I - \gamma \underline{\widehat{P}}^{\pi^\star, V}\Big)^{-1} \sqrt{\Big|\mathrm{Var}_{\widehat{P}^{\pi^\star}}(V^{\star,\sigma}) - \mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,V}}(V^{\star,\sigma})\Big|} \\
&= 2\sqrt{\frac{L}{N}}\Big(I - \gamma \underline{\widehat{P}}^{\pi^\star, V}\Big)^{-1} \sqrt{\Big|\Pi^{\pi^\star}\Big(\mathrm{Var}_{\widehat{P}^0}(V^{\star,\sigma}) - \mathrm{Var}_{\widehat{P}^{\pi^\star,V}}(V^{\star,\sigma})\Big)\Big|} \\
&\leq 2\sqrt{\frac{L}{N}}\Big(I - \gamma \underline{\widehat{P}}^{\pi^\star, V}\Big)^{-1} \sqrt{\Big\|\mathrm{Var}_{\widehat{P}^0}(V^{\star,\sigma}) - \mathrm{Var}_{\widehat{P}^{\pi^\star,V}}(V^{\star,\sigma})\Big\|_\infty 1} \\
&\leq 2\sqrt{\frac{L}{N}}\Big(I - \gamma \underline{\widehat{P}}^{\pi^\star, V}\Big)^{-1} \sqrt{\frac{1}{\gamma^2 \max\{1-\gamma, C_g\sigma\}}} 1 \qquad \text{(A.124)} \\
&\leq 4\sqrt{\frac{L}{\gamma^2(1-\gamma)^2 \max\{1-\gamma, C_g\sigma\} N}} 1, \qquad \text{(A.125)}
\end{aligned}$$

where the last inequality uses $\left(I-\gamma\underline{\widehat{P}}^{\pi^\star,V}\right)^{-1}1\le\frac{1}{1-\gamma}1$ (cf. (A.114)) for *sa*-rectangular. In the *s*-rectangular case, we obtain a different result as

$$\begin{aligned}\mathcal{R}_2&=2\sqrt{\frac{L}{N}}\left(I-\gamma\underline{\widehat{P}}^{\pi^\star,V}\right)^{-1}\sqrt{\left|\mathrm{Var}_{\widehat{P}^{\pi^\star}}(V^{\star,\sigma})-\mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,V}}(V^{\star,\sigma})\right|}\\&=2\sqrt{\frac{L}{N}}\left(I-\gamma\underline{\widehat{P}}^{\pi^\star,V}\right)^{-1}\sqrt{\left|\Pi^{\pi^\star}\left(\mathrm{Var}_{\widehat{P}^0}(V^{\star,\sigma})-\mathrm{Var}_{\widehat{P}^{\pi^\star,V}}(V^{\star,\sigma})\right)\right|}\\&\le 2\sqrt{\frac{L}{N}}\left(I-\gamma\underline{\widehat{P}}^{\pi^\star,V}\right)^{-1}\sqrt{\frac{1}{\gamma^2\max\{1-\gamma,\min_s\|\pi_s^*\|_\infty C_g\tilde{\sigma}\}}}1 \qquad (A.126)\\&\le 2\sqrt{\frac{L}{\gamma^2(1-\gamma)^2\max\{1-\gamma,\min_s\|\pi_s^*\|_\infty\tilde{\sigma}C_g\}N}}1, \qquad (A.127)\end{aligned}$$

- Consider $\mathcal{R}_3$. The following lemma plays an important role.
  Applying Lemma 6.2 and using $\pi=\pi^\star$ and $V=V^{\star,\sigma}$, it holds

$$\sqrt{\mathrm{Var}_{P^{\pi^\star}}(V^{\star,\sigma})}-\sqrt{\mathrm{Var}_{\widehat{P}^{\pi^\star}}(V^{\star,\sigma})}\le\sqrt{\frac{2\|V^{\star,\sigma}\|_\infty^2\log\left(\frac{2SA}{\delta}\right)}{N}}1,$$

  which can be inserted in (A.113) to gives

$$\begin{aligned}\mathcal{R}_3&=2\sqrt{\frac{L}{N}}\left(I-\gamma\underline{\widehat{P}}^{\pi^\star,V}\right)^{-1}\left(\sqrt{\mathrm{Var}_{P^{\pi^\star}}(V^{\star,\sigma})}-\sqrt{\mathrm{Var}_{\widehat{P}^{\pi^\star}}(V^{\star,\sigma})}\right)\\&\le\frac{4}{(1-\gamma)}\frac{\log\left(\frac{SAN}{\delta}\right)\|[V^{\star,\sigma}\|_\infty}{N}1\le\frac{4L}{(1-\gamma)^2N}1, \qquad (A.128)\end{aligned}$$

  where the last line uses $\left(I-\gamma\underline{\widehat{P}}^{\pi^\star,V}\right)^{-1}1\le\frac{1}{1-\gamma}1$ (cf. (A.114)).

Finally, inserting the results of $\mathcal{R}_1$ in (A.115), $\mathcal{R}_2$ in (A.125), $\mathcal{R}_3$ in (A.128), and (A.114) back into (A.113) gives

$$\begin{aligned}&\left(I-\gamma\underline{\widehat{P}}^{\pi^\star,V}\right)^{-1}\left(\underline{\widehat{P}}^{\pi^\star,V}V^{\pi^\star,\sigma}-\underline{P}^{\pi^\star,V}V^{\pi^\star,\sigma}\right) \qquad (A.129)\\&\le 8\sqrt{\frac{L}{\gamma^3(1-\gamma)^2\max\{1-\gamma,C_g\sigma\}N}\left(1+\sqrt{\frac{L}{(1-\gamma)^2N}}+\frac{C_S\|1\|_*L}{N(1-\gamma)}\right)}1+\frac{3LC_S\|1\|_*}{N(1-\gamma)^2}1\\&\quad+2\sqrt{\frac{2L}{\gamma^2(1-\gamma)^2\max\{1-\gamma,C_g\sigma\}N}}1+\frac{4L}{(1-\gamma)^2N}1\\&\le 10\sqrt{\frac{2L}{\gamma^3(1-\gamma)^2\max\{1-\gamma,C_g\sigma\}N}\left(1+\sqrt{\frac{L}{(1-\gamma)^2N}}+\frac{C_S\|1\|_*L}{N(1-\gamma)}\right)}1+\frac{4L}{(1-\gamma)^2N}1 \qquad (A.130)\\&\quad+\frac{3LC_S\|1\|_*}{N(1-\gamma)^2}1\\&\le 160\sqrt{\frac{L(1+\frac{C_S\|1\|_*}{N(1-\gamma)})}{(1-\gamma)^2\max\{1-\gamma,C_g\sigma\}N}}1+\frac{7LC_S\|1\|_*}{N(1-\gamma)^2}1, \qquad (A.131)\end{aligned}$$

where the last inequality holds by the fact $\gamma \geq \frac{1}{4}$ and letting $N \geq \frac{L}{(1-\gamma)^2}$. We have the same result for s-rectangular, replacing, $\max\{1-\gamma, C_g\sigma\}$ by $\max\{1-\gamma, \min_s \|\pi_s^*\|_* \tilde{\sigma} C_g\}$.

Now we are ready to control **second term in** (A.111) to control $\|\widehat{V}^{\pi^\star,\sigma} - V^{\pi^\star,\sigma}\|_\infty$. To proceed, applying Lemma 7.4 on the second term of the right-hand side of (A.111) leads to

$$
\begin{aligned}
&\left(I - \gamma \underline{\widehat{P}}^{\pi^\star,\widehat{V}}\right)^{-1}\left(\underline{\widehat{P}}^{\pi^\star,V} V^{\pi^\star,\sigma} - \underline{P}^{\pi^\star,V} V^{\pi^\star,\sigma}\right) \\
&\leq \left(I - \gamma \underline{\widehat{P}}^{\pi^\star,\widehat{V}}\right)^{-1}\left(2\sqrt{\frac{L}{N}}\sqrt{\mathrm{Var}_{P^{\pi^\star}}(V^{\star,\sigma})} + \frac{3LC_S\,\|1\|_*}{N(1-\gamma)}\right) \\
&\leq \left(I - \gamma \underline{\widehat{P}}^{\pi^\star,\widehat{V}}\right)^{-1}\frac{L'C_S\,\|1\|_*}{N(1-\gamma)} + \underbrace{2\sqrt{\frac{L}{N}}\left(I - \gamma \underline{\widehat{P}}^{\pi^\star,\widehat{V}}\right)^{-1}\sqrt{\mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,\widehat{V}}}(\widehat{V}^{\pi^\star,\sigma})}}_{=:\mathcal{R}_4} \\
&\underbrace{2\sqrt{\frac{L}{N}}\left(I - \gamma \underline{\widehat{P}}^{\pi^\star,\widehat{V}}\right)^{-1}\left(\sqrt{\mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,\widehat{V}}}(V^{\star,\sigma} - \widehat{V}^{\pi^\star,\sigma})}\right)}_{=:\mathcal{R}_5} \\
&\underbrace{+\,2\sqrt{\frac{L}{N}}\left(I - \gamma \underline{\widehat{P}}^{\pi^\star,\widehat{V}}\right)^{-1}\left(\sqrt{\left|\mathrm{Var}_{\widehat{P}^{\pi^\star}}(V^{\star,\sigma}) - \mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,\widehat{V}}}(V^{\star,\sigma})\right|}\right)}_{=:\mathcal{R}_6} \\
&\underbrace{+\,2\sqrt{\frac{L}{N}}\left(I - \gamma \underline{\widehat{P}}^{\pi^\star,\widehat{V}}\right)^{-1}\left(\sqrt{\mathrm{Var}_{P^{\pi^\star}}(V^{\star,\sigma})} - \sqrt{\mathrm{Var}_{\widehat{P}^{\pi^\star}}(V^{\star,\sigma})}\right)}_{=:\mathcal{R}_7}.
\end{aligned}
\tag{A.132}
$$

We now bound the above four terms $\mathcal{R}_4, \mathcal{R}_5, \mathcal{R}_6, \mathcal{R}_7$ separately.

- Using Lemma 7.3 with $P = \widehat{P}^{\pi^\star,\widehat{V}}$, $\pi = \pi^\star$ and $V = \widehat{V}^{\pi^\star,\sigma}$ which follow $\widehat{V}^{\pi^\star,\sigma} = r_{\pi^\star} + \gamma \underline{\widehat{P}}^{\pi^\star,\widehat{V}} \widehat{V}^{\pi^\star,\sigma}$, and in view of (A.114), the term $\mathcal{R}_4$ in (A.132) can be controlled as follows:

$$
\begin{aligned}
\mathcal{R}_4 &= 2\sqrt{\frac{L}{N}}\left(I - \gamma \underline{\widehat{P}}^{\pi^\star,\widehat{V}}\right)^{-1}\sqrt{\mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,\widehat{V}}}(\widehat{V}^{\pi^\star,\sigma})} \\
&\leq 2\sqrt{\frac{L}{N}}\sqrt{\frac{8\min\{\mathrm{sp}(\widehat{V}^{\pi^\star,\sigma})_*, 1/(1-\gamma))}{\gamma^2(1-\gamma)^2}1} \\
&\leq 8\sqrt{\frac{L}{\gamma^3(1-\gamma)^2\max\{1-\gamma, C_g\sigma\}N}1},
\end{aligned}
\tag{A.133}
$$

  where the last inequality is due to Lemma 7.1 for *sa*-rectangular case and with the same quantity replacing $\max\{1-\gamma, \sigma\}$ by $\max\{1-\gamma, \min_s \|\pi_s^*\|_* \tilde{\sigma}\}$ in the $s-$ rectangular case.

- For bounding $\mathcal{R}_5$, we can simply use (A.114)) to get

$$
\begin{aligned}
\mathcal{R}_5 &= 2\sqrt{\frac{L}{N}}\left(I - \gamma \underline{\widehat{P}}^{\pi^\star,\widehat{V}}\right)^{-1}\sqrt{\mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,\widehat{V}}}(V^{\pi^\star,\sigma} - \widehat{V}^{\pi^\star,\sigma})} \\
&\leq 2\sqrt{\frac{L}{(1-\gamma)^2 N}}\left\|V^{\star,\sigma} - \widehat{V}^{\pi^\star,\sigma}\right\|_\infty 1.
\end{aligned}
\tag{A.134}
$$

$$\mathcal{R}_5 \leq 2\sqrt{\frac{L}{(1-\gamma)^2 N}} \left\| V^{\star,\sigma} - \widehat{V}^{\pi^\star,\sigma} \right\|_\infty 1. \tag{A.135}$$

- The term $\mathcal{R}_6$ can upper bounded as (A.125) as follows:

$$\mathcal{R}_6 \leq 2\sqrt{\frac{2L}{\gamma^2(1-\gamma)^2 \max\{1-\gamma, C_g\sigma\} N}} 1. \tag{A.136}$$

  for sa-rectangular case and with the same quantity replacing $\max\{1-\gamma, C_g\sigma\}$ by $\max\{1-\gamma, \min_s \|\pi_s^*\|_* \tilde{\sigma} C_g\}$ in the $s-$ rectangular case.

- Finally, $\mathcal{R}_7$ can be controlled the same as (A.128) shown below:

$$\mathcal{R}_7 \leq \frac{4L}{(1-\gamma)^2 N} 1. \tag{A.137}$$

Combining the results in (A.133), (A.135), (A.136), and (A.137) and inserting back to (A.132) leads to for $N \geq \frac{L}{(1-\gamma)^2}$

$$\begin{aligned}
&\left(I - \gamma \underline{\widehat{P}}^{\pi^\star,\widehat{V}}\right)^{-1} \left(\underline{\widehat{P}}^{\pi^\star,V} V^{\pi^\star,\sigma} - \underline{P}^{\pi^\star,V} V^{\pi^\star,\sigma}\right) \leq 8\sqrt{\frac{L(1+\frac{C_S\|1\|_*}{N(1-\gamma)})}{\gamma^3(1-\gamma)^2 \max\{1-\gamma, C_g\sigma\} N}} 1 \\
&+ 2\sqrt{\frac{L}{(1-\gamma)^2 N}} \left\| V^{\star,\sigma} - \widehat{V}^{\pi^\star,\sigma} \right\|_\infty 1 + 2\sqrt{\frac{2L}{\gamma^2(1-\gamma)^2 \max\{1-\gamma, C_g\sigma\} N}} 1 + \frac{7LC_S \|1\|_*}{N(1-\gamma)^2} \\
&\leq 80\sqrt{\frac{L(1+\frac{C_S\|1\|_*}{N(1-\gamma)})}{(1-\gamma)^2 \max\{1-\gamma, C_g\sigma\} N}} 1 + 2\sqrt{\frac{L}{(1-\gamma)^2 N}} \left\| V^{\star,\sigma} - \widehat{V}^{\pi^\star,\sigma} \right\|_\infty 1 + \frac{7LC_S \|1\|_*}{N(1-\gamma)^2} 1,
\end{aligned} \tag{A.138}$$

where the last inequality follows from the assumption $\gamma \geq \frac{1}{4}$. Finally, inserting (A.131) and (A.138) back to (A.111) yields

$$\begin{aligned}
&\left\| \widehat{V}^{\pi^\star,\sigma} - V^{\pi^\star,\sigma} \right\|_\infty \leq \max\left\{ 160\sqrt{\frac{L(1+\frac{C_S\|1\|_*}{N(1-\gamma)})}{(1-\gamma)^2 \max\{1-\gamma, C_g\sigma\} N}} + \frac{7LC_S \|1\|_*}{N(1-\gamma)^2}, \right. \\
&\left. 80\sqrt{\frac{L(1+\frac{C_S\|1\|_*}{N(1-\gamma)})}{(1-\gamma)^2 \max\{1-\gamma, C_g\sigma\} N}} + 2\sqrt{\frac{L}{(1-\gamma)^2 N}} \left\| V^{\star,\sigma} - \widehat{V}^{\pi^\star,\sigma} \right\|_\infty + \frac{7LC_S \|1\|_*}{N(1-\gamma)^2} \right\} \\
&\leq 160\sqrt{\frac{L(1+\frac{C_S\|1\|_*}{N(1-\gamma)})}{(1-\gamma)^2 \max\{1-\gamma, C_g\sigma\} N}} + \frac{14LC_S \|1\|_*}{N(1-\gamma)^2},
\end{aligned} \tag{A.139}$$

where the last inequality holds by taking $N \geq \frac{16 \log\left(\frac{SAN}{\delta}\right)}{(1-\gamma)^2}$ rearranging terms. In s-rectangular case, we obtain the same result, replacing $\max\{1-\gamma, C_g\sigma\}$ by $\max\{1-\gamma, \min_s \|\pi_s^*\|_* C_g\tilde{\sigma}\}$.

Third step: controlling $\|\widehat{V}^{\widehat{\pi},\sigma} - V^{\widehat{\pi},\sigma}\|_\infty$ or bounding the first and second term in (A.110). Unlike the earlier term, one has to face a more complicated statistical dependency between $\widehat{\pi}$ and the empirical RMDP. To begin with, we introduce the following lemma which controls the main term on the right-hand side of (A.110), which is proved in Appendix 7.3.5.

Lemma 7.6. *Consider any $\delta \in (0,1)$. Taking $N \geq L''$ with probability at least $1-\delta$, one has for sa- or s-rectangular case :*

$$\begin{aligned}\left|\underline{\widehat{P}}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma} - \underline{P}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma}\right| &\leq 2\sqrt{\frac{L'}{N}}\sqrt{\mathrm{Var}_{P^0_{s,a}}(\widehat{V}^{\star,\sigma})}1 + 2\varepsilon_{\mathsf{opt}}1 + \frac{15L''C_S\left\|1\right\|_*}{N(1-\gamma)} \\ &\leq 2\sqrt{\frac{L''}{(1-\gamma)^2N}}1 + 2\varepsilon_{\mathsf{opt}}1 + \frac{14L''C_S\left\|1\right\|_*}{N(1-\gamma)}1. \end{aligned} \tag{A.140}$$

with $L'' = 2\log\left(\frac{54\|1\|_* SAN^2}{(1-\gamma)\delta}\right)$. Moreover, for TV this lemma holds but without the geometric term $\frac{14L''C_S\|1\|_*}{N(1-\gamma)}1$. Taking the sup over s gives the final result.

With Lemma 7.6 in hand, we have to control **first term** in (A.110)

$$\begin{aligned}&\left(I-\gamma\underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}\left(\underline{\widehat{P}}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma} - \underline{P}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma}\right) \\ &\overset{\text{(i)}}{\leq} \left(I-\gamma\underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}\left|\underline{\widehat{P}}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma} - \underline{P}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma}\right| \\ &\leq 2\sqrt{\frac{L'}{N}}\left(I-\gamma\underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}\sqrt{\mathrm{Var}_{P^{\widehat{\pi}}}(\widehat{V}^{\star,\sigma})} + \left(I-\gamma\underline{P}^{\widehat{\pi},V^{\widehat{\pi}}}\right)^{-1}\left(2\varepsilon_{\mathsf{opt}}\right)1 \\ &+ \left(I-\gamma\underline{P}^{\widehat{\pi},V^{\widehat{\pi}}}\right)^{-1}\frac{14L''C_S\left\|1\right\|_*}{N(1-\gamma)}1 \\ &\overset{\text{(ii)}}{\leq} \left(\frac{2\varepsilon_{\mathsf{opt}}}{1-\gamma}\right)1 + \underbrace{2\sqrt{\frac{L'}{N}}\left(I-\gamma\underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}\sqrt{\mathrm{Var}_{\underline{P}^{\widehat{\pi},\widehat{V}}}(\widehat{V}^{\widehat{\pi},\sigma})}}_{=:\mathcal{S}_1} \\ &\quad + \underbrace{2\sqrt{\frac{L'}{N}}\left(I-\gamma\underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}\sqrt{\left|\mathrm{Var}_{\underline{P}^{\widehat{\pi},\widehat{V}}}(\widehat{V}^{\star,\sigma}) - \mathrm{Var}_{\underline{P}^{\widehat{\pi},\widehat{V}}}(\widehat{V}^{\widehat{\pi},\sigma})\right|}}_{=:\mathcal{S}_2} \\ &\quad + \underbrace{2\sqrt{\frac{L'}{N}}\left(I-\gamma\underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}\sqrt{\left|\mathrm{Var}_{P^{\widehat{\pi}}}(\widehat{V}^{\star,\sigma}) - \mathrm{Var}_{\underline{P}^{\widehat{\pi},\widehat{V}}}(\widehat{V}^{\star,\sigma})\right|}}_{=:\mathcal{S}_3} + \frac{14L''C_S\left\|1\right\|_*}{N(1-\gamma)^2}1, \end{aligned} \tag{A.141, A.142}$$

where (i) and (ii) hold by the fact that each row of $(1-\gamma)\left(I-\gamma\underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}$ is a probability vector that falls into $\Delta(\mathcal{S})$. The remainder of the proof will focus on controlling the three terms in (A.142) separately.

- For $\mathcal{S}_1$, we introduce the following lemma, whose proof is postponed to 7.3.6.

  Lemma 7.7. *Consider any $\delta \in (0,1)$. Taking $N \geq \frac{L''}{(1-\gamma)^2}$ one has with probability at least $1-\delta$, for sa− rectangular*

$$\begin{aligned}&\left(I-\gamma\underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}\sqrt{\mathrm{Var}_{\underline{P}^{\widehat{\pi},\widehat{V}}}(\widehat{V}^{\widehat{\pi},\sigma})} \qquad \leq 6\sqrt{\frac{\left(1+\varepsilon_{\mathsf{opt}}+\frac{L''C_S\|1\|_*}{N(1-\gamma)}\right)}{\gamma^3(1-\gamma)^2\max\{1-\gamma,\sigma\}}}1 \\ &\leq 6\sqrt{\frac{\left(1+\varepsilon_{\mathsf{opt}}+\frac{L''C_S\|1\|_*}{N(1-\gamma)}\right)}{(1-\gamma)^3\gamma^3}}1. \end{aligned}$$

and for s-rectangular

$$\left(I-\gamma \underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}\sqrt{\mathrm{Var}_{\underline{P}^{\widehat{\pi},\widehat{V}}}(\widehat{V}^{\widehat{\pi},\sigma})}\leq 6\sqrt{\frac{L''\left(1+\varepsilon_{\mathsf{opt}}+\frac{C_S\|1\|_*}{N(1-\gamma)}\right)}{\gamma^3(1-\gamma)^2\max\{1-\gamma,C_g\tilde{\sigma}\min_s\|\hat{\pi}_s\|_\infty\}}}1$$
$$\leq 6\sqrt{\frac{L''\left(1+\varepsilon_{\mathsf{opt}}+\frac{C_S\|1\|_*}{N(1-\gamma)}\right)}{(1-\gamma)^3\gamma^2}}1.$$

Applying Lemma 7.7 and (A.114) to (A.142) leads to

$$\mathcal{S}_1=2\sqrt{\frac{L'}{N}}\left(I-\gamma \underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}\sqrt{\mathrm{Var}_{\underline{P}^{\widehat{\pi},\widehat{V}}}(\widehat{V}^{\widehat{\pi},\sigma})}$$
$$\leq 12\sqrt{\frac{L''}{\gamma^3(1-\gamma)^2\max\{1-\gamma,C_g\sigma\}N}}1. \tag{A.143}$$

for *sa*-rectangular and the same quantity replacing $\max\{1-\gamma,C_g\sigma\}$ by $\max\{1-\gamma,C_g\tilde{\sigma}\min_s\|\hat{\pi}_s\|_*\}$ for $s-$ rectangular case.

- Applying Lemma 6.1 with $\|\widehat{V}^{\star,\sigma}-\widehat{V}^{\widehat{\pi},\sigma}\|_\infty\leq\varepsilon_{\mathsf{opt}}$ and (A.114), $\mathcal{S}_2$ can be controlled as

$$\mathcal{S}_2=2\sqrt{\frac{L''}{N}}\left(I-\gamma \underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}\sqrt{\left|\mathrm{Var}_{\underline{P}^{\widehat{\pi},\widehat{V}}}(\widehat{V}^{\star,\sigma})-\mathrm{Var}_{\underline{P}^{\widehat{\pi},\widehat{V}}}(\widehat{V}^{\widehat{\pi},\sigma})\right|}$$
$$\leq 4\sqrt{\frac{L''}{N}}\left(I-\gamma \underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}\sqrt{\varepsilon_{\mathsf{opt}}\frac{1}{1-\gamma}^2}\leq 8\sqrt{\frac{\varepsilon_{\mathsf{opt}}L''}{(1-\gamma)^4N}}1. \tag{A.144}$$

- $\mathcal{S}_3$ can be controlled similar to $\mathcal{R}_2$ in (A.125) as follows:

$$\mathcal{S}_3=2\sqrt{\frac{L''}{N}}\left(I-\gamma \underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}\sqrt{\left|\mathrm{Var}_{P^{\widehat{\pi}}}(\widehat{V}^{\star,\sigma})-\mathrm{Var}_{\underline{P}^{\widehat{\pi},\widehat{V}}}(\widehat{V}^{\star,\sigma})\right|}$$
$$\leq 4\sqrt{\frac{L''}{N}}\left(I-\gamma \underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}\sqrt{\frac{1}{\gamma^2\max\{1-\gamma,C_g\sigma\}}}1 \tag{A.145}$$
$$\leq 8\sqrt{\frac{L''}{\gamma^2(1-\gamma)^2\max\{1-\gamma,C_g\sigma\}N}}1 \tag{A.146}$$

for *sa*-rectangular and replacing $\max\{1-\gamma,\sigma\}$ by $\max\{1-\gamma,\tilde{\sigma}\min_s\|\hat{\pi}_s\|_*\}$ for $s-$ rectangular case.

Finally, summing up the results in (A.143), (A.144), and (A.146) and inserting them back to

(A.142) yields: taking $N \geq \frac{L''}{(1-\gamma)^2}$, with probability at least $1-\delta$,

$$\left(I-\gamma \underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}\left(\underline{\widehat{P}}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma}-\underline{P}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma}\right) \leq \left(\frac{2\varepsilon_{\mathsf{opt}}}{1-\gamma}\right)1+\frac{14L''C_S\left\|1\right\|_*}{N(1-\gamma)^2}1$$
$$+12\sqrt{\frac{L''\left(1+\varepsilon_{\mathsf{opt}}+\frac{C_S\|1\|_*}{N(1-\gamma)}\right)}{\gamma^3(1-\gamma)^2\max\{1-\gamma,C_g\sigma\}N}}1+8\sqrt{\frac{\varepsilon_{\mathsf{opt}}L'}{(1-\gamma)^4N}}1+ \tag{A.147}$$
$$8\sqrt{\frac{L'}{\gamma^2(1-\gamma)^2\max\{1-\gamma,C_g\sigma\}N}}1$$
$$\leq 16\sqrt{\frac{L''\left(1+\varepsilon_{\mathsf{opt}}+\frac{C_S\|1\|_*}{N(1-\gamma)}\right)}{\gamma^3(1-\gamma)^2\max\{1-\gamma,\sigma\}N}}1+\left(\frac{2\varepsilon_{\mathsf{opt}}\gamma}{(1-\gamma)}+8\sqrt{\frac{\varepsilon_{\mathsf{opt}}\gamma L'}{(1-\gamma)^4N}}1+\frac{15L''C_S\left\|1\right\|_*}{N(1-\gamma)^2}1\right) \tag{A.148}$$
$$\tag{A.149}$$

for sa-rectangular and the same quantity replacing $\max\{1-\gamma,\sigma\}$ by $\max\{1-\gamma,\tilde{\sigma}\min_s\|\hat{\pi}_s\|_*\}$ for $s-$ rectangular case. In this step, it is harder to decouple terms as $\widehat{V}^{\widehat{\pi}}$ depends on data both in $\widehat{\pi}$ and $\widehat{V}$.

Step 5: controlling $\|\widehat{V}^{\widehat{\pi},\sigma}-V^{\widehat{\pi},\sigma}\|_\infty$: bounding the second term in (A.110). Towards this, applying Lemma 7.6 leads to in sa-rectangular case:

$$\left(I-\gamma \underline{P}^{\widehat{\pi},V}\right)^{-1}\left(\underline{\widehat{P}}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma}-\underline{P}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma}\right) \leq \left(I-\gamma \underline{P}^{\widehat{\pi},V}\right)^{-1}\left|\underline{\widehat{P}}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma}-\underline{P}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma}\right|$$
$$\leq 2\sqrt{\frac{L''}{N}}\left(I-\gamma \underline{P}^{\widehat{\pi},V}\right)^{-1}\sqrt{\mathrm{Var}_{P^{\widehat{\pi}}}(\widehat{V}^{\star,\sigma})}+\left(I-\gamma \underline{P}^{\widehat{\pi},V}\right)^{-1}\left(2\varepsilon_{\mathsf{opt}}\right)1 \tag{A.150}$$
$$+\left(I-\gamma \underline{P}^{\widehat{\pi},V}\right)^{-1}\frac{L''14C_S\left\|1\right\|_*}{N(1-\gamma)}1$$
$$\leq \left(\frac{2\varepsilon_{\mathsf{opt}}}{(1-\gamma)}\right)1+\underbrace{2\sqrt{\frac{L''}{N}}\left(I-\gamma \underline{P}^{\widehat{\pi},V}\right)^{-1}\sqrt{\mathrm{Var}_{\underline{P}^{\widehat{\pi},V}}(V^{\widehat{\pi},\sigma})}}_{=:\mathcal{S}_4}+\left(I-\gamma \underline{P}^{\widehat{\pi},V}\right)^{-1}\frac{14L''C_S\left\|1\right\|_*}{N(1-\gamma)}1$$
$$+\underbrace{2\sqrt{\frac{L'}{N}}\left(I-\gamma \underline{P}^{\widehat{\pi},V}\right)^{-1}\sqrt{\mathrm{Var}_{\underline{P}^{\widehat{\pi},V}}(\widehat{V}^{\widehat{\pi},\sigma}-V^{\widehat{\pi},\sigma})}}_{=:\mathcal{S}_5}$$
$$+\underbrace{2\sqrt{\frac{L''}{N}}\left(I-\gamma \underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}\sqrt{\left|\mathrm{Var}_{\underline{P}^{\widehat{\pi},V}}(\widehat{V}^{\star,\sigma})-\mathrm{Var}_{\underline{P}^{\widehat{\pi},V}}([\widehat{V}^{\widehat{\pi},\sigma})\right|}}_{=:\mathcal{S}_6}$$
$$+\underbrace{2\sqrt{\frac{L''}{N}}\left(I-\gamma \underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}\sqrt{\left|\mathrm{Var}_{P^{\widehat{\pi}}}(\widehat{V}^{\star,\sigma})-\mathrm{Var}_{\underline{P}^{\widehat{\pi},V}}([\widehat{V}^{\star,\sigma})\right|}}_{=:\mathcal{S}_7}. \tag{A.151}$$

We shall bound each of the terms separately.

- Applying Lemma 7.3 with $P=\underline{P}^{\widehat{\pi},V}$, $\pi=\widehat{\pi}$, and taking $V=V^{\widehat{\pi},\sigma}$ which obeys $V^{\widehat{\pi},\sigma}=$

$r_{\widehat{\pi}} + \gamma \underline{P}^{\widehat{\pi},V} V^{\widehat{\pi},\sigma}$, the term $\mathcal{S}_4$ can be controlled similar to (A.133) as follows:

$$\mathcal{S}_4 \le 8\sqrt{\frac{L''\left(1+\varepsilon_{\mathsf{opt}}+\frac{C_S\|1\|_*}{N(1-\gamma)}\right)}{\gamma^3(1-\gamma)^2\max\{1-\gamma, C_g\sigma\}N}}1. \tag{A.152}$$

for *sa*-rectangular and the same quantity replacing $\max\{1-\gamma, C_g\sigma\}$ by $\max\{1-\gamma, \min_s \|\hat{\pi}_s\|_* \tilde{\sigma} C_g\}$ for $s-$ rectangular case.

- For $\mathcal{S}_5$, it is observed that

$$\begin{aligned}\mathcal{S}_5 &= 2\sqrt{\frac{L''}{N}}\left(I-\gamma\underline{P}^{\widehat{\pi},V}\right)^{-1}\sqrt{\mathrm{Var}_{\underline{P}^{\widehat{\pi},V}}(\widehat{V}^{\widehat{\pi},\sigma}-V^{\widehat{\pi},\sigma})} \\ &\le 2\sqrt{\frac{L''}{(1-\gamma)^2N}}\left\|V^{\widehat{\pi},\sigma}-\widehat{V}^{\widehat{\pi},\sigma}\right\|_\infty 1.\end{aligned} \tag{A.153}$$

- Next, observing that $\mathcal{S}_6$ and $\mathcal{S}_7$ are almost the same as the terms $\mathcal{S}_2$ (controlled in (A.144)) and $\mathcal{S}_3$ (controlled in (A.146)) in (A.142), it is easily verified that they can be controlled as follows

$$\mathcal{S}_6 \le 4\sqrt{\frac{\varepsilon_{\mathsf{opt}}L''}{(1-\gamma)^4N}}1, \qquad \mathcal{S}_7 \le 4\sqrt{\frac{L''}{\gamma^2(1-\gamma)^2\max\{1-\gamma, C_g\sigma\}N}}1. \tag{A.154}$$

for *sa*-rectangular and the same quantity replacing $\max\{1-\gamma,\sigma\}$ by $\max\{1-\gamma, \min_s \|\hat{\pi}_s\|_* \tilde{\sigma}\}$ for $s-$ rectangular case. Then inserting the results in (A.152), (A.153), and (A.154) back to (A.151) leads to

$$\left(I-\gamma\underline{P}^{\widehat{\pi},V}\right)^{-1}\left(\underline{\widehat{P}}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma}-\underline{P}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma}\right) \tag{A.155}$$

$$\begin{aligned}&\le \left(\frac{2\varepsilon_{\mathsf{opt}}}{(1-\gamma)}\right)1 + 8\sqrt{\frac{L''\left(1+\varepsilon_{\mathsf{opt}}+\frac{C_S\|1\|_*}{N(1-\gamma)}\right)}{\gamma^3(1-\gamma)^2\max\{1-\gamma,\sigma\}N}}1 + \frac{14L''C_S\|1\|_*}{N(1-\gamma)^2}1 \\ &\quad + 2\sqrt{\frac{L''}{(1-\gamma)^2N}}\left\|V^{\widehat{\pi},\sigma}-\widehat{V}^{\widehat{\pi},\sigma}\right\|_\infty 1 + 4\sqrt{\frac{L''\varepsilon_{\mathsf{opt}}}{(1-\gamma)^4N}}1 + 4\sqrt{\frac{L''}{\gamma^2(1-\gamma)^2\max\{1-\gamma,C_g\sigma\}N}}1 \\ &\le 12\sqrt{\frac{L''\left(1+\varepsilon_{\mathsf{opt}}+\frac{C_S\|1\|_*}{N(1-\gamma)}\right)}{\gamma^3(1-\gamma)^2\max\{1-\gamma,\sigma\}N}} + 4\sqrt{\frac{L''}{(1-\gamma)^2N}}\left\|V^{\widehat{\pi},\sigma}-\widehat{V}^{\widehat{\pi},\sigma}\right\|_\infty 1 \qquad \text{(A.156)} \\ &\quad + \frac{3\varepsilon_{\mathsf{opt}}}{(1-\gamma)}1 + \frac{14L''C_S\|1\|_*}{N(1-\gamma)^2}1. \qquad \text{(A.157)}\end{aligned}$$

(A.158)

Taking $N \ge \frac{16L''}{1-\gamma}$, we obtain $\frac{2\varepsilon_{\mathsf{opt}}}{(1-\gamma)} + 4\varepsilon_{\mathsf{opt}}\sqrt{\frac{L''}{(1-\gamma)^4N}}1 \le \frac{3\varepsilon_{\mathsf{opt}}}{(1-\gamma)}$ with probability at least $1-\delta$,

inserting (A.148) and (A.156) back to (A.110)

$$\left\|\widehat{V}^{\widehat{\pi},\sigma}-V^{\widehat{\pi},\sigma}\right\|_\infty \leq \max\Big\{16\sqrt{\frac{L''\Big(1+\varepsilon_{\mathsf{opt}}+\frac{C_S\|1\|_*}{N(1-\gamma)}\Big)}{\gamma^3(1-\gamma)^2\max\{1-\gamma,\sigma\}N}}+\Big(\frac{2\varepsilon_{\mathsf{opt}}\gamma}{(1-\gamma)}+\frac{14L''C_S\left\|1\right\|_*}{N(1-\gamma)^2}\Big),$$
$$12\sqrt{\frac{L''\Big(1+\varepsilon_{\mathsf{opt}}+\frac{C_S\|1\|_*}{N(1-\gamma)}\Big)}{\gamma^3(1-\gamma)^2\max\{1-\gamma,\sigma\}N}}+4\sqrt{\frac{L''}{(1-\gamma)^2N}}\left\|V^{\widehat{\pi},\sigma}-\widehat{V}^{\widehat{\pi},\sigma}\right\|_\infty \tag{A.159}$$
$$+\frac{3\varepsilon_{\mathsf{opt}}}{(1-\gamma)}+\frac{14L''C_S\left\|1\right\|_*}{N(1-\gamma)^2}\Big\}$$
$$\leq 48\sqrt{\frac{L''\Big(1+\varepsilon_{\mathsf{opt}}+\frac{C_S\|1\|_*}{N(1-\gamma)}\Big)}{\gamma^3(1-\gamma)^2\max\{1-\gamma,C_g\sigma\}N}}+\frac{6\varepsilon_{\mathsf{opt}}}{(1-\gamma)}+\frac{28L''C_S\left\|1\right\|_*}{N(1-\gamma)^2} \tag{A.160}$$

for sa-rectangular and the same quantity, replacing $\max\{1-\gamma,C_g\sigma\}$ by $\max\{1-\gamma,\tilde{\sigma}\min_s\|\hat{\pi}_s\|_*\}$ for $s-$ rectangular case. The proof is similar for TV without the geometric term depending on C_S.

Step 6: summing all the previous inequalities results. Using all the previous results in (A.139) and (A.160) and inserting back to (A.105) complete the proof as follows: taking $N\geq\frac{16L''}{(1-\gamma)^2},\gamma>1/4,$, with probability at least $1-\delta$, for sa-rectangular

$$\begin{aligned}\left\|V^{\star,\sigma}-V^{\widehat{\pi},\sigma}\right\|_\infty &\leq \left\|V^{\pi^\star,\sigma}-\widehat{V}^{\pi^\star,\sigma}\right\|_\infty+\varepsilon_{\mathsf{opt}}+\left\|\widehat{V}^{\widehat{\pi},\sigma}-V^{\widehat{\pi},\sigma}\right\|_\infty\\ &\leq \varepsilon_{\mathsf{opt}}+48\sqrt{\frac{L''\Big(1+\varepsilon_{\mathsf{opt}}+\frac{C_S\|1\|_*}{N(1-\gamma)}\Big)}{\gamma^3(1-\gamma)^2\max\{1-\gamma,C_g\sigma\}N}}+\frac{6\varepsilon_{\mathsf{opt}}}{(1-\gamma)}+\frac{28L''C_S\left\|1\right\|_*}{N(1-\gamma)^2}\\ &+160\sqrt{\frac{L(1+\frac{C_S\|1\|_*}{N(1-\gamma)})}{(1-\gamma)^2\max\{1-\gamma,C_g\sigma\}N}}+\frac{14LC_S\left\|1\right\|_*}{N(1-\gamma)^2}\\ &\leq \frac{8\varepsilon_{\mathsf{opt}}}{1-\gamma}+\frac{42L''C_S\left\|1\right\|_*}{N(1-\gamma)^2}+1508\sqrt{\frac{L''(1+\frac{C_S\|1\|_*}{N(1-\gamma)})}{(1-\gamma)^2\max\{1-\gamma,C_g\sigma\}N}}\end{aligned} \tag{A.161}$$

where the last inequality holds by $\gamma\geq\frac{1}{4}$ and $N\geq\frac{16L''}{(1-\gamma)^2}$ for sa-rectangular and the same quantity replacing $\max\{1-\gamma,\sigma\}$ by $\max\{1-\gamma,\tilde{\sigma}\min_s\{\|\pi_s^*\|_*\}\}$ for $s-$ rectangular case. The proof is similar for TV without the geometric term depending on C_S.

7.3 Proof of the auxiliary lemmas

7.3.1 Proof of Lemma 7.1

Similarly to Shi et al. (2023), denoting s_0 the argmax of $V^{\pi,\sigma}$ such that $V^{\pi,\sigma}(s_0)=\min_{s\in\mathcal{S}}V^{\pi,\sigma}(s)$ using recursive Bellman's equation

$$\max_{s\in\mathcal{S}} V^{\pi,\sigma}(s) = \max_{s\in\mathcal{S}} \mathbb{E}_{a\sim\pi(\cdot|s)}\left[r(s,a) + \gamma \inf_{\mathcal{P}\in\mathcal{U}^{\sigma}(P_{s,a})} \mathcal{P}V^{\pi,\sigma}\right] \tag{A.162}$$

$$\leq \max_{(s,a)\in\mathcal{S}\times\mathcal{A}} \left(1 + \gamma \inf_{\mathcal{P}\in\mathcal{U}^{\sigma}(P_{s,a})} \mathcal{P}V^{\pi,\sigma}\right) \tag{A.163}$$

where the second line holds since the reward function $r(s,a) \in [0,1]$ for all $(s,a) \in \mathcal{S}\times\mathcal{A}$.

Then we construct for any $(s,a) \in \mathcal{S}\times\mathcal{A}$, $\widetilde{P}_{s,a} \in \mathbb{R}^S$ by reducing the values of some elements of $P_{s,a}$ such that $P_{s,a} \geq \widetilde{P}_{s,a} \geq 0$ and $\sum_{s'}\left(P_{s,a}(s') - \widetilde{P}_{s,a}(s')\right) = \sigma C_g^{s,a}$. with $C_g^{s,a} = \frac{1}{\|e_{s_0}\|}$ It lead to $\widetilde{P}_{s,a} + \sigma C_g^{s,a} e_{s_0}^{\top} \in \mathcal{U}^{\sigma}_{\|\|}(P_{s,a})$, where e_{s_0} is the standard basis vector supported on s_0, since

$$\frac{1}{2}\left\|\widetilde{P}_{s,a} + \sigma C_g^{s,a} e_{s_0}^{\top} - P_{s,a}\right\| \leq \frac{1}{2}\left\|\widetilde{P}_{s,a} - P_{s,a}\right\| + \frac{C_g^{s,a}\sigma\|e_{s_0}\|}{2} = \sigma/2 + \sigma/2 = \sigma \tag{A.164}$$

Consequently,

$$\inf_{\mathcal{P}\in\mathcal{U}^{\sigma}_{\|.\|}(P_{s,a})} \mathcal{P}V^{\pi,\sigma} \leq \left(\widetilde{P}_{s,a} + \sigma C_g^{s,a} e_{s_0}^{\top}\right) V^{\pi,\sigma} \leq \left\|\widetilde{P}_{s,a}\right\|_1 \|V^{\pi,\sigma}\|_{\infty} + \sigma V^{\pi,\sigma}(s_0) C_g^{s,a} \tag{A.165}$$

$$\leq (1 - C_g^{s,a}\sigma) \max_{s\in\mathcal{S}} V^{\pi,\sigma}(s) + \sigma C_g^{s,a} \min_{s\in\mathcal{S}} V^{\pi,\sigma}(s) \tag{A.166}$$

where the second inequality holds by

$$\left\|\widetilde{P}_{s,a}\right\|_1 = \sum_{s'} \widetilde{P}_{s,a}(s') = -\sum_{s'}\left(P_{s,a}(s') - \widetilde{P}_{s,a}(s')\right) + \sum_{s'} P_{s,a}(s') = 1 - \sigma C_g^{s,a} \tag{A.167}$$

Plugging this back to the previous relation gives

$$\max_{s\in\mathcal{S}} V^{\pi,\sigma}(s) \leq 1 + \gamma(1 - C_g^{s,a}\sigma) \max_{s\in\mathcal{S}} V^{\pi,\sigma}(s) + \gamma C_g^{s,a}\sigma \min_{s\in\mathcal{S}} V^{\pi,\sigma}(s) \tag{A.168}$$

which, by rearranging terms, yields

$$\max_{s\in\mathcal{S}} V^{\pi,\sigma}(s) \leq \frac{1 + \gamma C_g^{s,a}\sigma \min_{s\in\mathcal{S}} V^{\pi,\sigma}(s)}{1 - \gamma(1 - C_g^{s,a}\sigma)} \tag{A.169}$$

$$\leq \frac{1}{(1-\gamma) + \gamma C_g^{s,a}\sigma} + \min_{s\in\mathcal{S}} V^{\pi,\sigma}(s) \leq \frac{1}{\gamma\max\{1-\gamma, C_g^{s,a}\sigma\}} + \min_{s\in\mathcal{S}} V^{\pi,\sigma}(s) \tag{A.170}$$

So rearranging terms it holds :

$$\mathrm{sp}(V^{\pi,\sigma})_{\infty} \leq \frac{1}{\gamma\max\{1-\gamma, C_g^{s,a}\sigma\}} \tag{A.171}$$

or taking the sup over s:

$$\mathrm{sp}(V^{\pi,\sigma})_{\infty} \leq \frac{1}{\gamma\max\{1-\gamma, C_g\sigma\}} \tag{A.172}$$

As we pick the supreme over s, the quantity, $C_g^{s,a}$ is replaced by $C_g = 1/(\min_s \|e_s\|)$ to obtain a control for every s.

7.3.2 Proof of Lemma 7.2

Similarly to 7.1 denoting s_0 the argmax of $V^{\pi,\sigma}$ such that $V^{\pi,\sigma}(s_0) = \min_{s\in\mathcal{S}} V^{\pi,\sigma}(s)$ using recursive Bellman's equation

$$\max_{s\in\mathcal{S}} V^{\pi,\sigma}(s) = \max_{s\in\mathcal{S}} \mathbb{E}_{a\sim\pi(\cdot|s)}\left[r(s,a) + \gamma \inf_{\mathcal{P}\in\mathcal{U}^{\tilde{\sigma}}(P_s)} \mathcal{P}V^{\pi,\tilde{\sigma}}\right] \tag{A.173}$$

$$\leq \max_{s\in\mathcal{S}} \left(1 + \gamma \inf_{\mathcal{P}^\pi\in\mathcal{U}^{\tilde{\sigma}}(P_s^\pi)} \mathcal{P}^\pi V^{\pi,\tilde{\sigma}}\right) \tag{A.174}$$

where the second line holds since the reward function $r(s,a) \in [0,1]$ for all $(s,a) \in \mathcal{S}\times\mathcal{A}$.Then we construct for any $s \in \mathcal{S}$ $\widetilde{P}_s \in \mathbb{R}^{S\times A}$ by reducing the values of some elements of P_s such that $P_s \geq \widetilde{P}_s \geq 0$ and

$$\forall a \in A, \quad \sum_{s'} \left(P_s(s',a) - \widetilde{P}_s(s',a)\right) = \sigma_{s,a} C_g^s$$

where C_g^s is defined as $1/\|e_s\|$. Writting $\|\sigma_{s,a}\| \leq \tilde{\sigma}$ we construction $\sigma_{s,a}$ such that

$$\sum_a \pi(a|s) \sum_{s'} \left(P_s(s',a) - \widetilde{P}_s(s',a)\right) = \|\pi_s\|_* \tilde{\sigma} C_g^s. \tag{A.175}$$

Not that this construction is possible as it is simply Cauchy Swartz equality case. It leads to $\widetilde{P}_s + \sigma e_{s_0,a}^\top \in \mathcal{U}^{\tilde{\sigma}}(P_s)$, where $e_{s_0,a} \in \mathbb{R}^{S\times A}$ is the standard basis vector supported on s_0 which is equal to 1 at s_0 for every a and otherwise.

$$\frac{1}{2}\left\|\widetilde{P}_s + \sigma_{s,a} C_g^s e_{s_0,a}^\top - P_s\right\| \leq \frac{1}{2}\left\|\widetilde{P}_s - P_s\right\| + \frac{\tilde{\sigma}\|e_{s_0}\| C_g^s}{2} = \tilde{\sigma}/2 + \tilde{\sigma}/2 \tag{A.176}$$

as $C_g^s \|\sigma_{s,a} e_{s_0,a}\|$ is equal to $C_g^s \tilde{\sigma} \|e_{s_0}\|$ Consequently,

$$\inf_{\mathcal{P}^\pi\in\mathcal{U}^\sigma(P_s)} \mathcal{P}^\pi V^{\pi,\tilde{\sigma}} \leq \Pi^\pi \left(\widetilde{P}_s^\pi + \sigma C_g^s e_{s_0}^\top\right) V^{\pi,\tilde{\sigma}} \tag{A.177}$$

$$= \sum_a \sum_{s'} \widetilde{P}_s(s',a)\pi(a|s) V^{\pi,\tilde{\sigma}}(s') + \sigma e_{s_0,a} C_g^s V^{\pi,\tilde{\sigma}}(s_0)\pi(a|s) \tag{A.178}$$

$$\leq \sum_a \sup_{s'}[V^{\pi,\tilde{\sigma}}(s')](\sum_{s'} \widetilde{P}_s(s',a)))\pi(a|s) + V^{\pi,\tilde{\sigma}}(s_0)\pi(a|s)\sigma_{s,a} C_g^s \tag{A.179}$$

$$\overset{(a)}{=} \max_{s\in\mathcal{S}} V^{\pi,\sigma}(s) \sum_a (1 - \sigma C_g^s)\pi(a|s) + \sum_a V^{\pi,\tilde{\sigma}}(s_0)\pi(a|s)\sigma_{s,a} C_g^s \tag{A.180}$$

$$\overset{(b)}{=} \max_{s\in\mathcal{S}} V^{\pi,\sigma}(s)(1 - \tilde{\sigma} C_g^s)\|\pi_s\|_* + \|\pi_s\|_* \tilde{\sigma} C_g^s \min_{s\in\mathcal{S}} V^{\pi,\tilde{\sigma}}(s) \tag{A.181}$$

$$\leq (1 - C_g^s \tilde{\sigma}) \max_{s\in\mathcal{S}} V^{\pi,\sigma}(s) + \sigma C_g^s \min_{s\in\mathcal{S}} V^{\pi,\tilde{\sigma}}(s) \tag{A.182}$$

where $\|\pi\|_\infty$ is the norm of the vector $\pi(.|s)$ and where (a) holds because

$$\sum_{s'} \widetilde{P}_s(s') = -\sum_{s'} \left(P_s(s') - \widetilde{P}_s(s')\right) + \sum_{s'} P_s(s') = 1 - \sigma_{s,a} C_g^s \tag{A.183}$$

Finally (b) is due to (A.175) and using Holder's inequality in the second term. Plugging this back to the previous relation gives

$$\max_{s\in\mathcal{S}} V^{\pi,\tilde{\sigma}}(s) \leq 1 + \gamma(1-\tilde{\sigma}C_g^s \left\|\pi_s\right\|_*) \max_{s\in\mathcal{S}} V^{\pi,\sigma}(s) + \gamma \left\|\pi_s\right\|_* \tilde{\sigma}C_g^s \min_{s\in\mathcal{S}} V^{\pi,\tilde{\sigma}}(s) \tag{A.184}$$

which, by rearranging terms, yields

$$\begin{aligned}
\max_{s\in\mathcal{S}} V^{\pi,\tilde{\sigma}}(s) &\leq \frac{1+\gamma\tilde{\sigma}\left\|\pi_s\right\|_* C_g^s \min_{s\in\mathcal{S}} V^{\pi,\tilde{\sigma}}(s)}{1-\gamma(1-C_g^s\tilde{\sigma}\left\|\pi_s\right\|_*)} && \text{(A.185)}\\
&\leq \frac{1}{(1-\gamma)+\left\|\pi_s\right\|_*\gamma C_g^s\tilde{\sigma}} + \min_{s\in\mathcal{S}} V^{\pi,\tilde{\sigma}}(s) && \text{(A.186)}\\
&\leq \frac{1}{(1-\gamma)+\gamma\left\|\pi_s\right\|_* C_g^s\tilde{\sigma}} + \min_{s\in\mathcal{S}} V^{\pi,\tilde{\sigma}}(s) && \text{(A.187)}\\
&\leq \frac{1}{\gamma\max\{1-\gamma, C_g^s\left\|\pi_s\right\|_*\tilde{\sigma}\}} + \min_{s\in\mathcal{S}} V^{\pi,\tilde{\sigma}}(s). && \text{(A.188)}
\end{aligned}$$

So rearranging and taking the sumpremum over all $s$term it holds :

$$\mathrm{sp}(V^{\pi,\tilde{\sigma}})_\infty \leq \frac{1}{\gamma\max\{1-\gamma, \min_s\left\|\pi_s\right\|_* C_g\tilde{\sigma}\}}. \tag{A.189}$$

As we pick the supreme over s ovf this quantity, C_g^s is replaced by $C_g = 1/\min_s\left\|e_s\right\|$.

7.3.3 Proof of Lemma 7.4

Proof. Concentration of the robust values function. with probability $1-\delta$, it holds:

$$\left|P_{s,a}^{\pi,V}V - \widehat{P}_{s,a}^{\pi,V}V\right| \leq 2\sqrt{\frac{L}{N}}\sqrt{\mathrm{Var}_{P_{s,a}^0}(V)} + \frac{3LC_S\left\|1\right\|_*}{N(1-\gamma)}$$

with $L = 2\log(18\,\|1\|_* SAN/\delta)$ and First we can use optimization duality such as in (A.99):

$$\begin{aligned}
&\left|P^{\pi,V}_{s,a}V - \widehat{P}^{\pi,V}_{s,a}V\right| && \text{(A.190)}\\
&= \Big|\max_{\mu^{\lambda,\omega}_{P^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{P^0_{s,a}}} \left\{P^0_{s,a}(V-\mu) - \sigma\left(\mathrm{sp}((V-\mu))_*\right)\right\}\\
&- \max_{\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{\hat{P}^0_{s,a}}} \left\{\widehat{P}^0_{s,a}(V-\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}) - \sigma\left(\mathrm{sp}((V-\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}))_*\right)\right\}\Big|\\
&\leq \max\Big\{\Big|\max_{\mu^{\lambda,\omega}_{P^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{P^0_{s,a}}} \left\{P^0_{s,a}(V-\mu^{\lambda,\omega}_{P^0_{s,a}}) - \sigma\left(\mathrm{sp}((V-\mu^{\lambda,\omega}_{P^0_{s,a}}))_*\right)\right\}\\
&- \max_{\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{\hat{P}^0_{s,a}}} \left\{\widehat{P}^0_{s,a}(V-\mu^{\lambda,\omega}_{P^0_{s,a}}) - \sigma\left(\mathrm{sp}((V-\mu^{\lambda,\omega}_{P^0_{s,a}}))_*\right)\right\}\Big|; && \text{(A.191)}\\
&\Big|\max_{\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{\hat{P}^0_{s,a}}} \left\{\widehat{P}^0_{s,a}(V-\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}) - \sigma\left(\mathrm{sp}((V-\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}))_*\right)\right\} && \text{(A.192)}\\
&- \max_{\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{\hat{P}^0_{s,a}}} \left\{P^0_{s,a}(V-\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}) - \sigma\left(\mathrm{sp}((V-\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}))_*\right)\right\} \quad \Big|\Big\}\\
&\leq \max\Big\{\underbrace{\Big|\max_{\mu\in\mu^{\lambda,\omega}_{P^0_{s,a}}} \left(P^0_{s,a}-\widehat{P}^0_{s,a}\right)(V-\mu^{\lambda,\omega}_{P^0_{s,a}})\Big|}_{=:g_{s,a}(\alpha^{\lambda,\omega}_P,V)}, \underbrace{\Big|\max_{\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}} \in \mathcal{M}^{\lambda,\omega}_{\hat{P}^0_{s,a}}} \left(P^0_{s,a}-\widehat{P}^0_{s,a}\right)(V-\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}})\Big|}_{=:g_{s,a}(\alpha^{\lambda,\omega}_{\hat{P}},V)}\Big\} && \text{(A.193)}
\end{aligned}$$

where in the first equality we use Lemma 6.3. The final inequality is a consequence of the 1-Lipschitzness of the max operator. First, we control $g_{s,a}(\alpha^{\lambda,\omega}_P, V)$. To do so, we use for a fixed $\alpha^{\lambda,\omega}_P$ and any vector V that is independent with $\widehat{P}^0$, the Bernstein's inequality, one has with probability at least $1-\delta$ with *sa*-rectangular notations,

$$g_{s,a}(\alpha^{\lambda,\omega}_P, V) = \left|\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)[V]_{\alpha^{\lambda,\omega}_P}\right| \leq \sqrt{\frac{2\log\left(\frac{2}{\delta}\right)}{N}}\sqrt{\mathrm{Var}_{P^0_{s,a}}(V)} + \frac{2\log\left(\frac{2}{\delta}\right)}{3N(1-\gamma)}. \qquad \text{(A.194)}$$

Once pointwise concentration derived, we will use uniform concentration to yield this lemma. First, union bound, is obtained noticing that $g_{s,a}(\alpha^{\lambda,\omega}_P, V)$ is 1-Lipschitz w.r.t. λ and ω as it is linear in λ and ω. Moreover, $\lambda^* = \|V - \mu^* - \omega\|_*$ obeying $\lambda^* \leq \frac{\|1\|_*}{1-\gamma}$. The quantity $\omega \in [0, 1/(1-\gamma)]$ as it is always smaller that V by definition. We construct then a 2-dimensional a ε_1-net N_{ε_1} over $\lambda^* \in [0, \frac{\|1\|_*}{1-\gamma}]$ and $\omega \in [0, 1/(1-\gamma)]$ whose size satisfies $|N_{\varepsilon_1}| \leq \left(\frac{3\|1\|_*}{\varepsilon_1(1-\gamma)}\right)^2$ (Vershynin 2018). Using union bound and (A.194), it holds with probability at least $1 - \frac{\delta}{SA}$ that for all $\lambda \in N_{\varepsilon_1}$,

$$g_{s,a}(\alpha^{\lambda}_P, V) \leq \sqrt{\frac{2\log\left(\frac{2SA|N_{\varepsilon_1}|}{\delta}\right)}{N}}\sqrt{\mathrm{Var}_{P^0_{s,a}}(V)} + \frac{2\log\left(\frac{2SA|N_{\varepsilon_1}|}{\delta}\right)}{3N(1-\gamma)}. \qquad \text{(A.195)}$$

Using the previous equation and also (A.193), it results in using notation $2\log\Big(\frac{18SAN\|1\|_*}{\delta}\Big) = L$,

$$
\begin{aligned}
g_{s,a}(\alpha_P^\lambda, V) &\overset{(a)}{\leq} \sup_{\alpha_P^\lambda \in N_{\varepsilon_1}} \left|\left(P_{s,a}^0 - \widehat{P}_{s,a}^0\right)[V]_{\alpha_P^\lambda}\right| + \varepsilon_1 \\
&\overset{(b)}{\leq} \sqrt{\frac{2\log\Big(\frac{SA|N_{\varepsilon_1}|}{\delta}\Big)}{N}}\sqrt{\mathrm{Var}_{P_{s,a}^0}(V)} + \frac{2\log\Big(\frac{2SA|N_{\varepsilon_1}|}{\delta}\Big)}{3N(1-\gamma)} + \varepsilon_1 \qquad \text{(A.196)} \\
&\overset{(c)}{\leq} \sqrt{\frac{2\log\Big(\frac{2SA|N_{\varepsilon_1}|}{\delta}\Big)}{N}}\sqrt{\mathrm{Var}_{P_{s,a}^0}(V)} + \frac{\log\Big(\frac{2SA|N_{\varepsilon_1}|}{\delta}\Big)}{N(1-\gamma)} \\
&\overset{(d)}{\leq} \sqrt{\frac{L}{N}}\sqrt{\mathrm{Var}_{P_{s,a}^0}(V)} + \frac{L}{N(1-\gamma)} \qquad \text{(A.197)} \\
&\leq \sqrt{\frac{L}{N}}\|V\|_\infty + \frac{L}{N(1-\gamma)} \\
&\leq 2\sqrt{\frac{L}{(1-\gamma)^2 N}} \qquad \text{(A.198)}
\end{aligned}
$$

where (a) is because the optimal α falls into the ε_1-ball centered around some point inside N_{ε_1} and $g_{s,a}(\alpha_P^\lambda, V)$ is 1-Lipschitz with regard to λ and ω, (b) is due to Eq. (A.195), (c) arises from taking $\varepsilon_1 = \frac{\log\Big(\frac{2SA|N_{\varepsilon_1}|}{\delta}\Big)}{3N(1-\gamma)}$, (d) is verified by $|N_{\varepsilon_1}| \leq \Big(\frac{3\|1\|_*}{\varepsilon_1(1-\gamma)}\Big)^2 \leq 9N\|1\|$ and that variance of a ceiling function of a vector is smaller than the variance of non-ceiling vector , and the last inequality comes from the fact $\|V^{\star,\sigma}\|_\infty \leq \frac{1}{1-\gamma}$ and taking $N \geq 2\log\Big(\frac{18SAN\|1\|_*}{\delta}\Big) = L$.

Contrary to the previous term, the second term $g_{s,a}(\alpha_{\hat{P}}^\lambda, V)$ is more difficult as we need concentration. Still, the data has an extra dependency through the parameter $\alpha_{\hat{P}}^\lambda$. We need to decouple this problem using absorbing MDPs. Then it leads to

$$
\begin{aligned}
& g_{s,a}(\alpha_{\hat{P}}^{\lambda,\omega}, V) && \text{(A.199)} \\
&= \Big|\max_{\mu_{\hat{P}_{s,a}^0}^{\lambda,\omega} \in \mathcal{M}_{\hat{P}_{s,a}^0}^{\lambda,\omega}} \left(P_{s,a}^0 - \widehat{P}_{s,a}^0\right)(V - \mu_{\hat{P}_{s,a}^0}^{\lambda,\omega})\Big| && \text{(A.200)} \\
&= \Big|\max_{\mu \in \mathcal{M}_{\hat{P}_{s,a}^0}^{\lambda,\omega}} \left(P_{s,a}^0 - \widehat{P}_{s,a}^0\right)(V - \mu_{P_{s,a}^0}^{\lambda,\omega}) + \left(P_{s,a}^0 - \widehat{P}_{s,a}^0\right)(\mu_{P_{s,a}^0}^{\lambda,\omega} - \mu_{\hat{P}_{s,a}^0}^{\lambda,\omega})\Big| && \text{(A.201)} \\
&\leq \Big|\max_{\mu_{P_{s,a}^0}^{\lambda,\omega} \in \mathcal{M}_{P_{s,a}^0}^{\lambda,\omega}} \left(P_{s,a}^0 - \widehat{P}_{s,a}^0\right)(V - \mu_{P_{s,a}^0}^{\lambda,\omega}) + \max_{\mu_{\hat{P}_{s,a}^0}^{\lambda,\omega} \in \mathcal{M}_{\hat{P}_{s,a}^0}^{\lambda,\omega}} \left(P_{s,a}^0 - \widehat{P}_{s,a}^0\right)(\mu_{P_{s,a}^0}^{\lambda,\omega} - \mu_{\hat{P}_{s,a}^0}^{\lambda,\omega})\Big|. && \text{(A.202)}
\end{aligned}
$$

In the first equality, we add the term $\mu_{P_{s,a}^0}^{\lambda,\omega}$ to retrieve the previous concentration problem, fixing $P_{s,a}^0$ and optimizing λ, ω. In the second, we extend the max using triangular inequality. The first term in the last equality is exactly the term we have controlled previously, while the second one needs more attention. We decouple the data's dependency, then control the difference between the μ. Then using the characterization of the optimal μ from equation (A.96):

$$
\left(P_{s,a}^0 - \widehat{P}_{s,a}^0\right)(\mu_{P_{s,a}^0}^{\lambda,\omega} - \mu_{\hat{P}_{s,a}^0}^{\lambda,\omega}) = \sum_{s'} \lambda\left(P_{s,a}^0(s') - \widehat{P}_{s,a}^0(s')\right)\left(\nabla\left\|P_{s,a}^0\right\| - \nabla\left\|\hat{P}_{s,a}^0\right\|\right)
$$

Here we assume that the subgradient is a gradient as we assume that the norm is C^2. The question that arises is whether the gradient of the norm is Lipschitz.

Note that we are considering the worst case as $(\mu^{\lambda,\omega}_{P^0_{s,a}} - \mu^{\lambda,\omega}_{\hat{P}^0_{s,a}})$ can be zero in the case where μ the Lagrangian variable is equal to zero. Finally, note that we can also control this term when one of the two terms $\mu^{\lambda,\omega}_{P^0_{s,a}}$ or $\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}$ is equal to zero as $\mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}$ and $\mu^{\lambda,\omega}_{P^0_{s,a}}$ smaller that V because $V - \mu$ need to be positive in equation (A.92). In this case, classical control using Bernstein's inequality without uniform concentration can be applied, giving the same result. In the worst case where all terms in $(\mu^{\lambda,\omega}_{P^0_{s,a}} - \mu^{\lambda,\omega}_{\hat{P}^0_{s,a}})$ are non zero, assuming that the norm is C^2, using mean value theorem, we know that

$$\left\|(\nabla\left\|P^0_{s,a}\right\| - \nabla\left\|\hat{P}^0_{s,a}\right\|)\right\|_2 \leq \sup_{x\in\Delta(S)} \left\|\nabla^2\|x\|\right\|_2 \left\|(P^0_{s,a} - \hat{P}^0_{s,a})\right\|_2.$$

As the norm is C^2, is continuous and as the simplex is bounded, this quantity exists according to the Extreme value theorem. It is possible to compute this contact depending on S for explicit norms such as L_p. Indeed, for L_2:

$$\nabla^2\|x\|_2 = \frac{(I - \frac{x \bigotimes x)}{\|x\|_2^2}}{\|x\|_2} \leq \frac{1}{\|x\|_2} I \leq \frac{1}{\min_{x\in\Delta(S)} \|x\|_2} I = \sqrt{S}$$

where $\bigotimes$ is the Kronecker product. So we have an upper bound independent of x. For $L_p = \|x\|_p$ norms, $p \geq 2$, we have simple taking derivative twice:

$$\nabla^2\|x\|_p = \frac{p-1}{L_p}\left(\mathcal{A}^{p-2} - g_p g_p^T\right)$$

with

$$\mathcal{A} = \text{Diag}\left(\frac{\text{abs}(x)}{L_p}\right)$$
$$g_p = \mathcal{A}^{p-2}\left(\frac{x}{L_p}\right).$$

where Diag is the diagonal matrix. However, as $x \leq L_p$, $\mathcal{A} \leq I$, we get

$$H \leq \frac{p-1}{\|x\|_p} \leq (p-1)S^{1/q} = C_S \tag{A.203}$$

where the $1/L_p$ is minimized for the uniform distribution. Then using Cauchy Swartz inequality, it holds

$$\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)(\mu^{\lambda,\omega}_{P^0_{s,a}} - \mu^{\lambda,\omega}_{\hat{P}^0_{s,a}}) \leq \lambda\left\|\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)\right\|_2^2. \tag{A.204}$$

Then the question is how to bound the quantity $\left\|\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)\right\|_2^2$. To do so, we will use McDiarmid inequality.

Definition 7.1. *Bounded difference property*

A function $f : \mathcal{X}_1 \times \ldots \mathcal{X}_n \to \mathbb{R}$ satisfies the bounded difference property if for each $i = 1, \ldots, n$ the change of coordinate from s_i to s_i' may change the value of the function at most on c_i

$$\forall i \in [n] : \sup_{x_i'\in\mathcal{X}_i} \left|f\left(x_1, \ldots, x_i, \ldots, x_n\right) - f\left(x_1, \ldots, x_i', \ldots, x_n\right)\right| \leq c_i$$

In our case, we consider $f(X_1, \ldots, X_n) = \|\sum_{k=1}^n X_k\|_2$. Then we can notice that by triangle inequality for any $x_1, \ldots, x_n$ and x'_k with $X_{i,s'} = P^0_{i,s,a}(s') - P^0_{s,a}(s')$ (index i holds for index of sample generated from the generative model) that

$$f(x_1, \ldots, x_k, \ldots, x_n) = \|x_1 + \ldots + x_n\|_2 \leq \|x_1 + \ldots + x_n - x_k + x'_k\|_2 + \|x_k - x'_k\|_2$$
$$\leq f(x_1, \ldots, x'_k, \ldots, x_n) + 2$$

Theorem 7.8. *(McDiarmid's inequality). McDiarmid et al. (1989) Let $f : \mathcal{X}_1 \times \ldots \mathcal{X}_n \to \mathbb{R}$ be a function satisfying the bounded difference property with bounds $c_1, \ldots, c_n$. Consider independent random variables $X_1, \ldots, X_n, X_i \in \mathcal{X}_i$ for all i. Then for any $t > 0$*

$$\mathbb{P}[f(X_1, \ldots, X_n) - \mathbb{E}[f(X_1, \ldots, X_n)] \geq t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2}\right)$$

Using McDiarmid's inequality and union bound, we can bound the term here

$$\left(\left\|\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)\right\|_2 - \mathbb{E}[\left\|\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)\right\|_2]\right)^2 \leq \frac{2N\log(|S||A|/\delta))}{N^2}$$

with probability $1 - \delta/(|S||A|)$. Moreover, the additional term can be bounded as follows:

$$\mathbb{E}[\left\|\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)\right\|_2^2] = \mathbb{E}[\sum_{s'}(P^0_{s,a}(s') - P^0_{s,a}(s'))^2 = \mathbb{E}[\sum_{s'}(\frac{1}{N}\sum_i^N X_{i,s'})^2]$$

with $X_{i,s'} = P^0_{i,s,a}(s') - P^0_{s,a}(s')$ is one sample sampled from the generative model. Then

$$\mathbb{E}[\left\|\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)\right\|_2^2] = \frac{1}{N^2}\sum_{s'}\mathsf{Var}(\sum_i^N X_{i,s}) \overset{a}{=} \frac{1}{N^2}\sum_i^N\sum_{s'}\mathsf{Var}(X_{i,s}) \tag{A.205}$$

$$= \frac{1}{N^2}\sum_i^N \mathbb{E}(\sum_{s'} X_{i,s}^2) \leq \frac{4}{N} \tag{A.206}$$

where (a) the last equality comes from the independence of the random variables, and where the last inequality comes from the fact the maximum of two elements in the simplex is bounded by 2.

Moreover, we know that,

$$\mathbb{E}\left[\left\|\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)\right\|_2\right]^2 \leq \mathbb{E}[\left\|\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)\right\|_2^2] \tag{A.207}$$

due to Jensen's inequality. Finally, regrouping the two terms, we obtain with probability $1 - \delta/(|S||A|)$:

$$\left\|\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)\right\|_2^2 = \left(\left\|\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)\right\|_2 - \mathbb{E}\left[\left\|\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)\right\|_2\right]\right)^2 + \left(\mathbb{E}\left[\left\|\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)\right\|_2\right]\right)^2$$
$$+ 2\mathbb{E}\left[\left\|\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)\right\|_2\right]\left(\left\|\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)\right\|_2 - \mathbb{E}\left[\left\|\left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)\right\|_2\right]\right)$$
$$\overset{a}{\leq} \frac{2N\log(|S||A|/(\delta)))}{N^2} + \frac{4}{N} + \frac{\sqrt{\frac{4}{N}}\sqrt{2N\log(|S||A|/(\delta)))}}{N}$$
$$\leq \frac{10\log(|S||A|/(\delta))}{N} = \frac{L'}{N}$$

where in first inequality use $(a+b)^2 = a^2 + b^2 + 2ab$ and where in (a) we combine equation (A.207) and (A.206) and (A.205).

with $L' = 10\log(|S||A|/(\delta))$. Finally, plugging the previous equation in (A.204):

$$\max_{\mu \in \mu^{\lambda}_{\widehat{P}^0_{s,a}}} \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)(\mu^{\lambda}_{P^0_{s,a}} - \mu)| \leq \max_{\lambda} \left\| \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right) \right\|_2^2 C_S \lambda.$$

This term can be easily controlled by taking the supremum over λ, which is a 1 dimensional parameter. Then we can bound $\lambda \in [0, H\,\|1\|_*]$. Indeed,

$$\lambda^* = \|V - \mu^* - \eta\|_* \leq \|V\|_* \leq H\,\|1\|_*\,.$$

Finally, we obtain:

$$\max_{\lambda} \left\| \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right) \right\|_2^2 C_S \lambda \leq \frac{L' C_S\,\|1\|_*}{N(1-\gamma)}.$$

Regrouping all terms:

$$\begin{aligned} g_{s,a}(\alpha^{\lambda}_{\hat{P}}, V) &\leq |\max_{\mu^{\lambda}_{P^0_{s,a}} \in \mathcal{M}^{\lambda}_{P^0_{s,a}}} \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)(V - \mu^{\lambda}_{P^0_{s,a}}) + \max_{\mu^{\lambda}_{\hat{P}^0_{s,a}} \in \mathcal{M}^{\lambda}_{\hat{P}^0_{s,a}}} \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right)(\mu^{\lambda}_{P^0_{s,a}} - \mu^{\lambda}_{\hat{P}^0_{s,a}})| \\ &\leq 2\sqrt{\frac{L}{N}}\sqrt{\mathrm{Var}_{P^0_{s,a}}(V)} + \frac{L' C_S\,\|1\|_*}{N(1-\gamma)} + \frac{L}{N(1-\gamma)} \\ &\leq 2\sqrt{\frac{L}{N}}\sqrt{\mathrm{Var}_{P^0_{s,a}}(V)} + \frac{3LC_S\,\|1\|_*}{N(1-\gamma)} \qquad \text{(A.208)} \\ & \qquad \text{(A.209)} \end{aligned}$$

We can recognize that the second term is a second-order term as long as $N \geq (C_S\,\|1\|_*)^2$, we can regroup the two terms. Finally, as $g_{s,a}(\alpha^{\lambda}_{\hat{P}}, V) \geq g_{s,a}(\alpha^{\lambda}_{P}, V)$, we obtain

$$\left|P^{\pi,V}_{s,a} V - \widehat{P}^{\pi,V}_{s,a} V\right| \leq 2\sqrt{\frac{L}{N}}\sqrt{\mathrm{Var}_{P^0_{s,a}}(V)} + \frac{3LC_S\,\|1\|_*}{N(1-\gamma)} \qquad \text{(A.210)}$$

It is important to note that the geometry of the norm is present in the second order term $\frac{3LC_S\|1\|}{N(1-\gamma)}$ but this term is negligible as it is proportional to $1/N$ with regard to the variance term in $1/\sqrt{N}$. Moreover, note that the quantity $C_S\,\|1\|_* = S$ for L_2 norms.

For the specific case of TV which is not C^2 smooth, this lemma still holds as in (A.193), we only need to control one term without the dependency on data in the supremum as α^{λ}_{P} reduces to a scalar α which does not depend on P. Then extra decomposition using smoothness of the norm is not needed, as the only remaining term in the max in (A.193) is the left-hand side term.

For the s-rectangular case, the first equation can be rewritten simply by factorizing by $\pi(a|s)$ using lemma 6.4.

$$\left|P^{\pi,V}_{s,a}V - \widehat{P}^{\pi,V}_{s,a}V\right| = \Big|\sum_a \pi(a|s) \max_{\mu^\lambda_{P^0_{s,a}} \in \mathcal{M}^\lambda_{P^0_{s,a}}} \left\{P^0_{s,a}(V-\mu) - \sigma\left(\mathrm{sp}((V-\mu))_*\right)\right\}$$

$$- \max_{\mu^\lambda_{\widehat{P}^0_{s,a}} \in \mathcal{M}^\lambda_{\widehat{P}^0_{s,a}}} \left\{\widehat{P}^0_{s,a}(V-\mu^\lambda_{\widehat{P}^0_{s,a}}) - \sigma\left(\mathrm{sp}((V-\mu^\lambda_{\widehat{P}^0_{s,a}})_*\right)\right\}\Big| \tag{A.211}$$

$$\leq \sum_a \pi(a|s)\Big(2\sqrt{\frac{L}{N}}\sqrt{\mathrm{Var}_{P^0_{s,a}}(V)} + \frac{LC_S\left\|1\right\|_*}{N(1-\gamma)}\Big) \tag{A.212}$$

$$= 2\sqrt{\frac{L}{N}}\sqrt{\mathrm{Var}_{P^0_{s,a}}(V)} + \frac{3LC_S\left\|1\right\|_*}{N(1-\gamma)} \tag{A.213}$$

using sa-rectangular results, which gives the result for s-rectangular case.

Combining this lemma with a matrix notation using union bound, one has with probability $1-\delta$:

$$\left|\underline{\hat{P}}^{\pi^*,V}V^{\pi^*,\sigma} - \underline{P}^{\pi^*,V}V^{\pi^*,\sigma}\right| \leq 2\sqrt{\frac{L}{N}}\sqrt{\mathrm{Var}_{P^*}\left(V^{\star,\sigma}\right)} + \frac{3LC_S\left\|1\right\|_*}{N(1-\gamma)}1 \tag{A.214}$$

$$\tag{A.215}$$

□

7.3.4 Proof of Lemma 7.5

Using the same argument as in (A.265), it holds that for any α^* solution of (A.102)

$$\left(I - \gamma\underline{\widehat{P}}^{\pi^\star,V}\right)^{-1}\sqrt{\mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,V}}(V^{\star,\sigma})} = \sqrt{\frac{1}{1-\gamma}}\sqrt{\sum_{t=0}^{\infty}\gamma^t\left(\underline{\widehat{P}}^{\pi^\star,V}\right)^t\mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,V}}(V^{\star,\sigma})}. \tag{A.216}$$

Then we can control $\mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,V}}(V^{\star,\sigma})$. Defining $V' \coloneqq V^{\star,\sigma} - \eta 1$, $\eta \in \mathbb{R}$, we use Bellman's equation in (A.81)) which lead to

$$V' = V^{\star,\sigma} - \eta 1 \leq V^{\star,\sigma} - \eta 1 = r_{\pi^\star} + \gamma\underline{P}^{\pi^\star,V}V^{\star,\sigma} - \eta 1 \tag{A.217}$$

$$= r_{\pi^\star} + \gamma P^{\pi^\star,V}V^{\star,\sigma} - \gamma\sigma\mathrm{sp}(V^{\star,\sigma})_* - \eta 1 \tag{A.218}$$

$$= r'_{\pi^\star} + \gamma\underline{\widehat{P}}^{\pi^\star,V}V' + \gamma\Big(P^{\pi^\star,V} - \underline{\widehat{P}}^{\pi^\star,V}\Big)V^{\star,\sigma} - \gamma\sigma\mathrm{sp}(V^{\star,\sigma})_* \tag{A.219}$$

$$= r'_{\pi^\star} + \gamma\underline{\widehat{P}}^{\pi^\star,V}V' + \gamma\Big(\underline{P}^{\pi^\star,V} - \underline{\widehat{P}}^{\pi^\star,V}\Big)V^{\star,\sigma} \tag{A.220}$$

$$\leq r'_{\pi^\star} + \gamma\underline{\widehat{P}}^{\pi^\star,V}V' + \gamma\Big(\underline{P}^{\pi^\star,V} - \underline{\widehat{P}}^{\pi^\star,V}\Big)V^{\star,\sigma} \tag{A.221}$$

where in the second line we use Lemma 6.3. and we define $r'_{\pi^\star} = r_{\pi^\star} - (1-\gamma)\eta < r_{\pi^\star} < 1$. We obtain the same result in s-rectangular case using lemma 6.4 instead. Then

$$
\begin{aligned}
\mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,V}}(V^{\star,\sigma}) &\overset{(a)}{=} \mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,V}}(V') = \underline{\widehat{P}}^{\pi^\star,V}(V'\circ V') - (\underline{\widehat{P}}^{\pi^\star,V}V')\circ(\underline{\widehat{P}}^{\pi^\star,V}V') \\
&= \underline{\widehat{P}}^{\pi^\star,V}(V'\circ V') - (\underline{\widehat{P}}^{\pi^\star,V}V')\circ(\underline{\widehat{P}}^{\pi^\star,V}V') \\
&\overset{(b)}{\leq} \underline{\widehat{P}}^{\pi^\star,V}(V'\circ V') - \frac{1}{\gamma^2}\Big(V' - r'_{\pi^\star} - \gamma\Big(\underline{P}^{\pi^\star,V} - \underline{\widehat{P}}^{\pi^\star,V}\Big)V^{\star,\sigma}\Big)^{\circ 2} \\
&= \underline{\widehat{P}}^{\pi^\star,V}(V'\circ V') - \frac{1}{\gamma^2}V'\circ V' + \frac{2}{\gamma^2}V'\circ\Big(r'_{\pi^\star} + \gamma\Big(\underline{P}^{\pi^\star,V} - \underline{\widehat{P}}^{\pi^\star,V}\Big)V^{\star,\sigma}\Big) \\
&\quad - \frac{1}{\gamma^2}\Big(r'_{\pi^\star} + \gamma\Big(\underline{P}^{\pi^\star,V} - \underline{\widehat{P}}^{\pi^\star,V}\Big)V^{\star,\sigma}\Big)^{\circ 2} \\
&\overset{(c)}{\leq} \underline{\widehat{P}}^{\pi^\star,V}(V'\circ V') - \frac{1}{\gamma}V'\circ V' + \frac{2}{\gamma^2}\|V'\|_\infty 1 && \text{(A.222)} \\
&\quad + \frac{2}{\gamma}\|V'\|_\infty\Big|\Big(\underline{P}^{\pi^\star,V} - \underline{\widehat{P}}^{\pi^\star,V}\Big)V^{\star,\sigma}\Big| && \text{(A.223)} \\
&\leq \underline{\widehat{P}}^{\pi^\star,V}(V'\circ V') - \frac{1}{\gamma}V'\circ V' + \frac{2}{\gamma^2}\|V'\|_\infty 1 && \text{(A.224)} \\
&\quad + \frac{2}{\gamma}\|V'\|_\infty\Big(2\sqrt{\frac{L}{(1-\gamma)^2N}} + \frac{3C_S\,\|1\|_*\,L}{N(1-\gamma)}\Big)1, && \text{(A.225)}
\end{aligned}
$$

where (a) holds by the fact that $\mathrm{Var}_{P_\pi}(V-\eta 1) = \mathrm{Var}_{P_\pi}(V)$ for any scalar η, (b) follows from (A.221), moreover (c) comes from $\frac{1}{\gamma^2}V'\circ V' \geq \frac{1}{\gamma}V'\circ V'$ and $-1 \leq r_{\pi^\star} - (1-\gamma)V_{\min}1 = r'_{\pi^\star} \leq r_{\pi^\star} \leq 1$. Finally, the inequality is due to Lemma 7.4. Plugging (A.225) into (A.216) gives,

$$
\begin{aligned}
&\Big(I - \gamma\underline{\widehat{P}}^{\pi^\star,V}\Big)^{-1}\sqrt{\mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,V}}(V^{\star,\sigma})} && \text{(A.226)} \\
&\leq \sqrt{\frac{1}{1-\gamma}}\Big(\sum_{t=0}^{\infty}\gamma^t\Big(\underline{\widehat{P}}^{\pi^\star,V}\Big)^t\Big(\underline{\widehat{P}}^{\pi^\star,V}(V'\circ V') - \frac{1}{\gamma}V'\circ V' + \frac{2}{\gamma^2}\|V'\|_\infty 1 && \text{(A.227)} \\
&\quad + \frac{2}{\gamma}\|V'\|_\infty\Big(2\sqrt{\frac{L}{(1-\gamma)^2N}} + \frac{3C_S\,\|1\|_*\,L}{N(1-\gamma)}\Big)1\Big)\Big)^{1/2} \\
&\overset{(i)}{\leq} \sqrt{\frac{1}{1-\gamma}}\sqrt{\Big|\sum_{t=0}^{\infty}\gamma^t\Big(\underline{\widehat{P}}^{\pi^\star,V}\Big)^t\Big(\underline{\widehat{P}}^{\pi^\star,V}(V'\circ V') - \frac{1}{\gamma}V'\circ V'\Big)\Big|} \\
&\quad + \sqrt{\frac{1}{1-\gamma}}\sqrt{\sum_{t=0}^{\infty}\gamma^t\Big(\underline{\widehat{P}}^{\pi^\star,V}\Big)^t\Big(\frac{2}{\gamma^2}\|V'\|_\infty 1 + \frac{2}{\gamma}\|V'\|_\infty\Big(2\sqrt{\frac{L}{(1-\gamma)^2N}} + \frac{3C_S\,\|1\|_*\,L}{N(1-\gamma)}\Big)1\Big)} \\
&\leq \sqrt{\frac{1}{1-\gamma}}\sqrt{\Big|\sum_{t=0}^{\infty}\gamma^t\Big(\underline{\widehat{P}}^{\pi^\star,V}\Big)^t\Big[\underline{\widehat{P}}^{\pi^\star,V}(V'\circ V') - \frac{1}{\gamma}V'\circ V'\Big]\Big|} && \text{(A.228)} \\
&\quad + \sqrt{\frac{\Big(2 + 2\Big(2\sqrt{\frac{L}{(1-\gamma)^2N}} + \frac{3C_S\|1\|_*L}{N(1-\gamma)}\Big)\Big)\|V'\|_\infty}{(1-\gamma)^2\gamma^2}}1, && \text{(A.229)}
\end{aligned}
$$

using in (i) the triangle inequality. The final part of the proof focuses on the first term, which follows

$$
\begin{aligned}
&\Big|\sum_{t=0}^{\infty}\gamma^t\Big(\underline{\widehat{P}}^{\pi^\star,V}\Big)^t\Big(\underline{\widehat{P}}^{\pi^\star,V}(V'\circ V') - \frac{1}{\gamma}V'\circ V'\Big)\Big| \\
&= \Big|\Big(\sum_{t=0}^{\infty}\gamma^t\Big(\underline{\widehat{P}}^{\pi^\star,V}\Big)^{t+1} - \sum_{t=0}^{\infty}\gamma^{t-1}\Big(\underline{\widehat{P}}^{\pi^\star,V}\Big)^t\Big)(V'\circ V')\Big| \leq \frac{1}{\gamma}\|V'\|_\infty^2 1 && \text{(A.230)}
\end{aligned}
$$

using recursion between the two sums. Then, using (A.230) back to (A.229) leads to

$$\begin{aligned}
&\left(I-\gamma\underline{\widehat{P}}^{\pi^\star,V}\right)^{-1}\sqrt{\mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,V}}(V^{\star,\sigma})}\\
&\leq\sqrt{\frac{\|V\|_\infty^2}{\gamma(1-\gamma)}}1+3\sqrt{\frac{\left(1+\left(\sqrt{\frac{L}{(1-\gamma)^2N}}+\frac{C_S\|1\|_*L}{N(1-\gamma)}\right)\right)\|V'\|_\infty}{(1-\gamma)^2\gamma^2}}1\\
&\leq 4\sqrt{\frac{\left(1+\left(\sqrt{\frac{L}{(1-\gamma)^2N}}+\frac{C_S\|1\|_*L}{N(1-\gamma)}\right)\right)\|V'\|_\infty}{(1-\gamma)^2\gamma^2}}1 \qquad \text{(A.231)}\\
&\leq 4\sqrt{\frac{\left(1+\left(1\sqrt{\frac{L}{(1-\gamma)^2N}}+\frac{C_S\|1\|_*L}{N(1-\gamma)}\right)\right)\|V'\|_*}{(1-\gamma)^2\gamma^2}}1 \qquad \text{(A.232)}
\end{aligned}$$

Taking the infimum over η in the right-hand side, recall $V' \coloneqq V^{\star,\sigma}-\eta 1$, we obtain the definition of the span semi norm.

$$\begin{aligned}
\left(I-\gamma\underline{\widehat{P}}^{\pi^\star,V}\right)^{-1}\sqrt{\mathrm{Var}_{\underline{\widehat{P}}^{\pi^\star,V}}(V^{\star,\sigma})} &\leq 4\sqrt{\frac{\left(1+\left(\sqrt{\frac{L}{(1-\gamma)^2N}}+\frac{C_S\|1\|_*L}{N(1-\gamma)}\right)\right)\mathrm{sp}(V^{\star,\sigma})_*}{(1-\gamma)^2\gamma^2}}1\\
&\leq 4\sqrt{\frac{\left(1+\left(\sqrt{\frac{L}{(1-\gamma)^2N}}+\frac{C_S\|1\|_*L}{N(1-\gamma)}\right)\right)}{\gamma^3(1-\gamma)^2\max\{1-\gamma,C_g\sigma\}}}1 \qquad \text{(A.233)}\\
&\leq 4\sqrt{\frac{\left(1+\left(\sqrt{\frac{L}{(1-\gamma)^2N}}+\frac{C_S\|1\|_*L}{N(1-\gamma)}\right)\right)}{\gamma^3(1-\gamma)^3}}1, \qquad \text{(A.234)}
\end{aligned}$$

where the penultimate inequality follows from applying Lemma 7.1 with $P=P^0$ and $\pi=\pi^\star$:

$$\mathrm{sp}(V^{\star,\sigma})_* \leq \frac{1}{\gamma\max\{1-\gamma,C_g\sigma\}}$$

or with an extra factor for s rectangular assumptions.

$$\mathrm{sp}(V^{\star,\sigma})_* \leq \frac{1}{\gamma\max\{1-\gamma,\min_s\|\pi_s\|_*\tilde{\sigma}Cg\}}.$$

7.3.5 Proof of Lemma 7.6

In this proof, we will *sa*-rectangular notations, for any $(s,a)\in\mathcal{S}\times\mathcal{A}$, using the results in (A.193). In the *sa*-rectangular case:

$$\left|\widehat{P}_{s,a}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma}-P_{s,a}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma}\right| \leq \max\left\{\left|\left(P_{s,a}^0-\widehat{P}_{s,a}^0\right)[\widehat{V}^{\widehat{\pi},\sigma}]_{\alpha_{P_{s,a}}^{\lambda,\omega*}}\right|,\left|\left(P_{s,a}^0-\widehat{P}_{s,a}^0\right)[\widehat{V}^{\widehat{\pi},\sigma}]_{\alpha_{\hat{P}_{s,a}}^{\lambda,\omega*}}\right|\right\} \qquad \text{(A.235)}$$

The first term in this max can be bounded using:

$$\left|\left(P^0_{s,a}-\widehat{P}^0_{s,a}\right)\left[\widehat{V}^{\widehat{\pi},\sigma}\right]_{\alpha^{\lambda,\omega*}_{P_{sa}}}\right| \tag{A.236}$$

$$\overset{(a)}{\leq}\left(\left|\left(P^0_{s,a}-\widehat{P}^0_{s,a}\right)\left[\widehat{V}^{\star,\sigma}\right]_{\alpha^{\lambda,\omega*}_{P_{sa}}}\right|+\left|\left(P^0_{s,a}-\widehat{P}^0_{s,a}\right)\left(\left[\widehat{V}^{\widehat{\pi},\sigma}\right]_{\alpha^{\lambda,\omega*}_{P_{sa}}}-\left[\widehat{V}^{\star,\sigma}\right]_{\alpha^{\lambda,\omega*}_{P_{sa}}}\right)\right|\right)$$

$$\leq\left(\left|\left(P^0_{s,a}-\widehat{P}^0_{s,a}\right)\left[\widehat{V}^{\star,\sigma}\right]_{\alpha^{\lambda,\omega*}_{P_{sa}}}\right|+\left\|P^0_{s,a}-\widehat{P}^0_{s,a}\right\|_1\left\|\left[\widehat{V}^{\widehat{\pi},\sigma}\right]_{\alpha^{\lambda,\omega*}_{P_{sa}}}-\left[\widehat{V}^{\star,\sigma}\right]_{\alpha^{\lambda,\omega*}_{P_{sa}}}\right\|_\infty\right)$$

$$\overset{(b)}{\leq}\left|\left(P^0_{s,a}-\widehat{P}^0_{s,a}\right)\left[\widehat{V}^{\star,\sigma}\right]_{\alpha^{\lambda,\omega*}_{P_{sa}}}\right|+2\left\|\widehat{V}^{\widehat{\pi},\sigma}-\widehat{V}^{\star,\sigma}\right\|_\infty$$

$$\overset{(c)}{\leq}\left|\left(P^0_{s,a}-\widehat{P}^0_{s,a}\right)\left[\widehat{V}^{\star,\sigma}\right]_{\alpha^{\lambda,\omega*}_{P_{sa}}}\right|+2\varepsilon_{\mathsf{opt}} \tag{A.237}$$

where (a) comes from the triangle inequality, and (b) comes from $\|P^0_{s,a}-\widehat{P}^0_{s,a}\|_1\leq 2$ and $\left\|\left[\widehat{V}^{\widehat{\pi},\sigma}\right]_{\alpha^{\lambda,\omega*}_{P_{sa}}}-\left[\widehat{V}^{\star,\sigma}\right]_{\alpha^{\lambda,\omega*}_{P_{sa}}}\right\|_\infty\leq\left\|\widehat{V}^{\widehat{\pi},\sigma}-\widehat{V}^{\star,\sigma}\right\|_\infty$, and (c) follows from the definition of the optimization error in (A.104). The second term of the max can be controlled in the same manner, i.e.:

$$\left|\left(P^0_{s,a}-\widehat{P}^0_{s,a}\right)\left[\widehat{V}^{\widehat{\pi},\sigma}\right]_{\alpha^{\lambda,\omega*}_{\hat{P}_{sa}}}\right|\leq\left|\left(P^0_{s,a}-\widehat{P}^0_{s,a}\right)\left[\widehat{V}^{\star,\sigma}\right]_{\alpha^{\lambda,\omega*}_{\hat{P}_{sa}}}\right|+2\varepsilon_{\mathsf{opt}} \tag{A.238}$$

$$\leq|\max_{\mu^\lambda_{P^0_{s,a}}\in\mathcal{M}^\lambda_{P^0_{s,a}}}\left(P^0_{s,a}-\widehat{P}^0_{s,a}\right)(\widehat{V}^{\star,\sigma}-\mu^\lambda_{P^0_{s,a}})+\max_{\mu^\lambda_{\hat{P}^0_{s,a}}\in\mathcal{M}^\lambda_{\hat{P}^0_{s,a}}}\left(P^0_{s,a}-\widehat{P}^0_{s,a}\right)(\mu^\lambda_{P^0_{s,a}}-\mu^\lambda_{\hat{P}^0_{s,a}})| \tag{A.239}$$

$$+2\varepsilon_{\mathsf{opt}} \tag{A.240}$$

where the last inequality follow the decomposition of (A.199). Finally, to control the remaining term

$$\max_{\mu^\lambda_{P^0_{s,a}}\in\mathcal{M}^\lambda_{P^0_{s,a}}}\left(P^0_{s,a}-\widehat{P}^0_{s,a}\right)(\widehat{V}^{\star,\sigma}-\mu^\lambda_{P^0_{s,a}})=\max_{\alpha^\lambda_P\in\mathrm{A}^\lambda_P}\left\{(P^0_{s,a}-\widehat{P}^0_{s,a})\left[V\right]_{\alpha^\lambda_P}\right\} \tag{A.241}$$

(A.240) for any given $\alpha\in[0,\alpha^{\lambda,\omega*}_{P_{sa}}[\subset[0,\frac{1}{1-\gamma}]^S$ in the variational family with one parameter λ, with the dependency between $\widehat{V}^{\star,\sigma}$ and $\widehat{P}^0$, we resort to the following leave-one-out argument or absorbing MDPs used in (Agarwal et al. 2020, Li et al. 2022, Shi and Chi 2022, Clavier et al. 2023). To begin, we create a collection of auxiliary RMDPs that exhibit the intended statistical independence between robust value functions and the estimated nominal transition kernel. These auxiliary RMDPs are designed to be minimally distinct from the initial RMDPs, subsequently, we manage to control the relevant term within these auxiliary RMDPs and demonstrate that its value closely approximates the target quantity for the desired RMDP. Recall that the empirical infinite-horizon robust MDP $\widehat{\mathcal{M}}_{\mathsf{rob}}$ is defined using the nominal transition kernel $\widehat{P}^0$. Inspired by Agarwal et al. (2020), we can construct an auxiliary absorbing robust MDP $\widehat{\mathcal{M}}^{s,u}_{\mathsf{rob}}$ for each state s and any non-negative scalar $u\geq 0$, so that it is the same as $\widehat{\mathcal{M}}_{\mathsf{rob}}$ except for the transition properties in state s. These auxiliary MDPS are called absorbing MDPs are have been used for the first time in the context of RMDPS in Clavier et al. (2023). Defining the reward function and nominal transition kernel of $\widehat{\mathcal{M}}^{s,u}_{\mathsf{rob}}$ as $P^{s,u}$ and $r^{s,u}$, which are expressed as follows using the same notation as Shi et al. (2023):

$$\begin{cases} r^{s,u}(s,a)=u & \forall a\in\mathcal{A},\\ r^{s,u}(\tilde{s},a)=r(\tilde{s},a) & \forall(\tilde{s},a)\in\mathcal{S}\times\mathcal{A}\text{ and }\tilde{s}\neq s.\end{cases} \tag{A.242}$$

$$\begin{cases} P^{s,u}(s' \,|\, s,a) = \mathbb{1}(()s' = s) & \forall (s',a) \in \mathcal{S} \times \mathcal{A}, \\ P^{s,u}(\cdot \,|\, \widetilde{s},a) = \widehat{P}^0(\cdot \,|\, \widetilde{s},a) & \forall (\widetilde{s},a) \in \mathcal{S} \times \mathcal{A} \text{ and } \widetilde{s} \neq s, \end{cases} \tag{A.243}$$

Nominal transition probability at state s of the auxiliary $\widehat{\mathcal{M}}^{s,u}_{\mathsf{rob}}$ never leaves state s once entered, which gives the name absorbing to these auxiliary RMPDs. Finally, we define the robust Bellman operator $\widehat{\mathcal{T}}^{\sigma}_{s,u}(\cdot)$ associated $\widehat{\mathcal{M}}^{s,u}_{\mathsf{rob}}$ as

$$\widehat{\mathcal{T}}^{\sigma}_{s,u}(Q)(\tilde{s},a) = r^{s,u}(\tilde{s},a) + \gamma \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}(P^{s,u}_{\tilde{s},a})} \mathcal{P}V, \qquad \text{with } V(\tilde{s}) = \max_a Q(\tilde{s},a). \tag{A.244}$$

in sa-rectangular case and with stochastic policy in s-rectangular case. Using these auxiliary RMDPs we can remark equivalence between $\widehat{\mathcal{M}}_{\mathsf{rob}}$ and the auxiliary RMDP $\widehat{\mathcal{M}}^{s,u}_{\mathsf{rob}}$ fixed-point. First, $\widehat{Q}^{\star,\sigma}$ is the unique-fixed point of $\widehat{\mathcal{T}}^{\sigma}(\cdot)$ with associated value $\widehat{V}^{\star,\sigma}$. We will show that the robust value function $\widehat{V}^{\star,\sigma}_{s,u^\star}$ obtained from the fixed point of $\widehat{\mathcal{T}}^{\sigma}_{s,u}(\cdot)$is the same as the the robust value function $\widehat{V}^{\star,\sigma}$ derived from $\widehat{\mathcal{T}}^{\sigma}(\cdot)$, as long as we choose u as

$$u^\star := u^\star(s) = \widehat{V}^{\star,\sigma}(s) - \gamma \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}(e_s)} \mathcal{P}\widehat{V}^{\star,\sigma}. \tag{A.245}$$

with e_s is the s-th standard basis vector in $\mathbb{R}^S$. This assertion is verified as:

- **First for state $s' \neq s$, for all $a \in \mathcal{A}$**: it holds

$$\begin{aligned} r^{s,u^\star}(s',a) + \gamma \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}(P^{s,u^\star}_{s',a})} \mathcal{P}\widehat{V}^{\star,\sigma} &= r(s',a) + \gamma \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}(\widehat{P}^0_{s',a})} \mathcal{P}\widehat{V}^{\star,\sigma} \\ &= \widehat{\mathcal{T}}^{\sigma}(\widehat{Q}^{\star,\sigma})(s',a) = \widehat{Q}^{\star,\sigma}(s',a), \end{aligned} \tag{A.246}$$

  where the first equality holds because of (A.242) and (A.243), and the last inequality comes from that $\widehat{Q}^{\star,\sigma}$ is the fixed point of $\widehat{\mathcal{T}}^{\sigma}(\cdot)$ (see Lemma 6.3) and the definition of the robust Bellman operator in (3.13).

- **Then for state s, for any $a \in \mathcal{A}$** :

$$\begin{aligned} &r^{s,u^\star}(s,a) + \gamma \inf_{\mathcal{P} \in \mathcal{U}^{\sigma}(P^{s,u^\star}_{s,a})} \mathcal{P}\widehat{V}^{\star,\sigma} = u^\star + \gamma \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}(e_s)} \mathcal{P}\widehat{V}^{\star,\sigma} \\ &= \widehat{V}^{\star,\sigma}(s) - \gamma \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}(e_s)} \mathcal{P}\widehat{V}^{\star,\sigma} + \gamma \inf_{\mathcal{P} \in \mathcal{U}^{\mathsf{sa},\sigma}(e_s)} \mathcal{P}\widehat{V}^{\star,\sigma} = \widehat{V}^{\star,\sigma}(s), \end{aligned} \tag{A.247}$$

  using in the first equality is the definition of $P^{s,u^\star}_{s,a}$ in (A.243) and where we use the definition of $u^\star$ in (A.245) in the second one.

Finally, we have proved that there exists a fixed point $\widehat{Q}^{\star,\sigma}_{s,u^\star}$ of the operator $\widehat{\mathcal{T}}^{\sigma}_{s,u^\star}(\cdot)$ by taking

$$\begin{cases} \widehat{Q}^{\star,\sigma}_{s,u^\star}(s,a) = \widehat{V}^{\star,\sigma}(s) & \forall a \in \mathcal{A}, \\ \widehat{Q}^{\star,\sigma}_{s,u^\star}(s',a) = \widehat{Q}^{\star,\sigma}(s',a) & \forall s' \neq s \text{ and } a \in \mathcal{A}. \end{cases} \tag{A.248}$$

we have confirmed the existence of a fixed point of the operator $\widehat{\mathcal{T}}^{\sigma}_{s,u^\star}(\cdot)$ with corresponding value function $\widehat{V}^{\star,\sigma}_{s,u^\star}$ that coincide with $\widehat{V}^{\star,\sigma}$. Note that the corresponding properties between $\widehat{\mathcal{M}}_{\mathsf{rob}}$ and $\widehat{\mathcal{M}}^{s,u}_{\mathsf{rob}}$ in Step 1 and Step 2 hold in fact for any uncertainty set and s- or sa-rectangular assumptions. Equipped with these fixed point equalities, we can use concentration inequalities to show this lemma.

Concentration inequality using an ε-net for all reward values u. First we can verify that

$$0 \leq u^\star \leq [\widehat{V}^{\star,\sigma}(s)]_{\alpha^{\lambda,\omega*}_{P_{s,a}}} \leq \widehat{V}^{\star,\sigma}(s) \leq \frac{1}{1-\gamma}. \tag{A.249}$$

Then, we define a N_{ε_2}-net over the interval $[0, 1/(1-\gamma)]$, where $|N_{\varepsilon_2}|$ the size of the net can be controlled by $|N_{\varepsilon_2}| \leq \frac{3}{\varepsilon_2(1-\gamma)}$ (Vershynin 2018). The only parameter that varies is λ in the variation family, $\alpha^{\lambda}_{P_{sa}}$ so we have 1-dimensional control and not a vector in $\mathbb{R}^S$. Then similarly to Lemma 6.3, it holds that for each $u \in N_{\varepsilon_2}$, there exists a unique fixed point $\widehat{Q}^{\star,\sigma}_{s,u}$ of the operator $\widehat{\mathcal{T}}^{\sigma}_{s,u}(\cdot)$, which satisfies $0 \leq \widehat{Q}^{\star,\sigma}_{s,u} \leq \frac{1}{1-\gamma} \cdot 1$. Consequently, the corresponding robust value function can be upper bounded by $\left\|\widehat{V}^{\star,\sigma}_{s,u}\right\|_\infty \leq \frac{1}{1-\gamma}$. Using (A.243) and (A.242) by construction for all $u \in N_{\varepsilon_2}$, $\widehat{\mathcal{M}}^{s,u}_{\mathsf{rob}}$ is statistically independent of $\widehat{P}^0_{s,a}$. This independence indicates that $[\widehat{V}^{\star,\sigma}_{s,u}]_\alpha$ and $\widehat{P}^0_{s,a}$ are independent for a fixed α. Using (A.197) and (A.198) and taking the union bound over all $(s,a,\alpha) \in \mathcal{S} \times \mathcal{A} \times N_{\varepsilon_1}$, $u \in N_{\varepsilon_2}$ gives that, with probability at least $1-\delta$, it holds for all $(s,a,u) \in \mathcal{S} \times \mathcal{A} \times N_{\varepsilon_2}$ that

$$\begin{aligned}
\max_{\alpha^{\lambda,\omega}_{P_{sa}} \in \mathrm{A}^{\lambda,\omega}_{P_{sa}}} \left| \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right) [\widehat{V}^{\star,\sigma}_{s,u}]_{\alpha^{\lambda,\omega*}_{P_{sa}}} \right| &\leq 2\sqrt{\frac{2\log\left(\frac{18\|1\|_* SAN|N_{\varepsilon_2}|}{\delta}\right)}{N}} \sqrt{\mathrm{Var}_{P^0_{s,a}}(\widehat{V}^{\star,\sigma}_{s,u})} \qquad \text{(A.250)} \\
&\quad + \varepsilon_2 \\
&\leq 2\sqrt{\frac{2\log\left(\frac{18\|1\|_* SAN|N_{\varepsilon_2}|}{\delta}\right)}{(1-\gamma)^2 N}} + \varepsilon_2, \qquad \text{(A.251)}
\end{aligned}$$

Finally, we use **uniform concentration** to obtain the lemma. Recalling that $u^\star \in [0, \frac{1}{1-\gamma}]$ (see (A.249)), we can always find some $\overline{u} \in N_{\varepsilon_2}$ such that $|\overline{u} - u^\star| \leq \varepsilon_2$. Consequently, plugging in the operator $\widehat{\mathcal{T}}^{\sigma}_{s,u}(\cdot)$ in (A.244) yields

$$\forall Q \in \mathbb{R}^{SA} : \quad \left\|\widehat{\mathcal{T}}^{\sigma}_{s,\overline{u}}(Q) - \widehat{\mathcal{T}}^{\sigma}_{s,u^\star}(Q)\right\|_\infty = |\overline{u} - u^\star| \leq \varepsilon_2$$

We can then remark that the fixed points of $\widehat{\mathcal{T}}^{\sigma}_{s,\overline{u}}(\cdot)$ and $\widehat{\mathcal{T}}^{\sigma}_{s,u^\star}(\cdot)$ obey

$$\begin{aligned}
\left\|\widehat{Q}^{\star,\sigma}_{s,\overline{u}} - \widehat{Q}^{\star,\sigma}_{s,u^\star}\right\|_\infty &= \left\|\widehat{\mathcal{T}}^{\sigma}_{s,\overline{u}}(\widehat{Q}^{\star,\sigma}_{s,\overline{u}}) - \widehat{\mathcal{T}}^{\sigma}_{s,u^\star}(\widehat{Q}^{\star,\sigma}_{s,u^\star})\right\|_\infty \\
&\leq \left\|\widehat{\mathcal{T}}^{\sigma}_{s,\overline{u}}(\widehat{Q}^{\star,\sigma}_{s,\overline{u}}) - \widehat{\mathcal{T}}^{\sigma}_{s,\overline{u}}(\widehat{Q}^{\star,\sigma}_{s,u^\star})\right\|_\infty + \left\|\widehat{\mathcal{T}}^{\sigma}_{s,\overline{u}}(\widehat{Q}^{\star,\sigma}_{s,u^\star}) - \widehat{\mathcal{T}}^{\sigma}_{s,u^\star}(\widehat{Q}^{\star,\sigma}_{s,u^\star})\right\|_\infty \\
&\leq \gamma \left\|\widehat{Q}^{\star,\sigma}_{s,\overline{u}} - \widehat{Q}^{\star,\sigma}_{s,u^\star}\right\|_\infty + \varepsilon_2,
\end{aligned}$$

where we use that the operator $\widehat{\mathcal{T}}^{\sigma}_{s,u}(\cdot)$ is a γ-contraction. It gives that:

$$\left\|\widehat{Q}^{\star,\sigma}_{s,\overline{u}} - \widehat{Q}^{\star,\sigma}_{s,u^\star}\right\|_\infty \leq \frac{\varepsilon_2}{(1-\gamma)} \quad \text{and} \quad \left\|\widehat{V}^{\star,\sigma}_{s,\overline{u}} - \widehat{V}^{\star,\sigma}_{s,u^\star}\right\|_\infty \leq \left\|\widehat{Q}^{\star,\sigma}_{s,\overline{u}} - \widehat{Q}^{\star,\sigma}_{s,u^\star}\right\|_\infty \leq \frac{\varepsilon_2}{(1-\gamma)}. \tag{A.252}$$

Finally to control the first term in (A.240), using the identity $\widehat{V}^{\star,\sigma} = \widehat{V}^{\star,\sigma}_{s,u^\star}$ or fixed point relation between the two RMPDS, established in previous step of the proof gives that: for all

$(s,a) \in \mathcal{S} \times \mathcal{A}$,

$$
\begin{aligned}
&\max_{\alpha^{\lambda,\omega}_{P_{s,a}} \in \mathrm{A}^{\lambda,\omega}_{P_{s,a}}} \left| \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right) [\widehat{V}^{\star,\sigma}]_{\alpha^{\lambda,\omega}_{P_{s,a}}} \right| \\
&\leq \max_{\alpha^{\lambda,\omega}_{P_{s,a}} \in \mathrm{A}^{\lambda,\omega}_{P_{s,a}}} \left| \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right) [\widehat{V}^{\star,\sigma}]_{\alpha^{\lambda,\omega}_{P_{s,a}}} \right| \\
&\overset{(a)}{\leq} \max_{\alpha^{\lambda,\omega}_{P_{s,a}} \in \mathrm{A}^{\lambda,\omega}_{P_{s,a}}} \left\{ \left| \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right) [\widehat{V}^{\star,\sigma}_{s,\overline{u}}]_{\alpha^{\lambda,\omega}_{P_{s,a}}} \right| + \left| \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right) \left([\widehat{V}^{\star,\sigma}_{s,\overline{u}}]_{\alpha^{\lambda,\omega}_{P_{s,a}}} - [\widehat{V}^{\star,\sigma}_{s,u^\star}]_{\alpha^{\lambda}_{P_{s,a}}} \right) \right| \right\} \\
&\overset{(b)}{\leq} \max_{\alpha^{\lambda,\omega}_{P_{s,a}} \in \mathrm{A}^{\lambda,\omega}_{P_{s,a}}} \left| \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right) [\widehat{V}^{\star,\sigma}_{s,\overline{u}}]_{\alpha^{\lambda}_{P_{s,a}}} \right| + \frac{2\varepsilon_2}{(1-\gamma)} \\
&\overset{(c)}{\leq} \frac{2\varepsilon_2}{(1-\gamma)} + \varepsilon_2 + 2\sqrt{\frac{2\log\left(\frac{18\|1\|_* SAN|N_{\varepsilon_2}|}{\delta}\right)}{N}} \sqrt{\mathrm{Var}_{P^0_{s,a}}(\widehat{V}^{\star,\sigma}_{s,\overline{u}})} + \frac{4\log\left(\frac{18\|1\|_* SAN|N_{\varepsilon_2}|}{\delta}\right)}{3N(1-\gamma)} \\
&\leq \frac{3\varepsilon_2}{(1-\gamma)} + 2\sqrt{\frac{2\log\left(\frac{18\|1\|_* SAN|N_{\varepsilon_2}|}{\delta}\right)}{N}} \sqrt{\mathrm{Var}_{P^0_{s,a}}(\widehat{V}^{\star,\sigma})} + \frac{4\log\left(\frac{18\|1\|_* SAN|N_{\varepsilon_2}|}{\delta}\right)}{3N(1-\gamma)} \\
&\quad + 2\sqrt{\frac{2\log\left(\frac{18\|1\|_* SAN|N_{\varepsilon_2}|}{\delta}\right)}{N}} \sqrt{\left|\mathrm{Var}_{P^0_{s,a}}(\widehat{V}^{\star,\sigma}) - \mathrm{Var}_{P^0_{s,a}}(\widehat{V}^{\star,\sigma}_{s,\overline{u}})\right|} \\
&\overset{(d)}{\leq} \frac{3\varepsilon_2}{(1-\gamma)} + 2\sqrt{2\frac{\log\left(\frac{18\|1\|_* SAN|N_{\varepsilon_2}|}{\delta}\right)}{N}} \sqrt{\mathrm{Var}_{P^0_{s,a}}(\widehat{V}^{\star,\sigma})} + 2\sqrt{\frac{4\varepsilon_2 \log\left(\frac{18\|1\|_* SAN|N_{\varepsilon_2}|}{\delta}\right)}{N(1-\gamma)^2}} \qquad \text{(A.253)} \\
&\leq 2\sqrt{\frac{L''}{N}} \sqrt{\mathrm{Var}_{P^0_{s,a}}(\widehat{V}^{\star,\sigma})} + \frac{14\log\left(\frac{54\|1\|_* SAN|N_{\varepsilon_2}|}{\delta}\right)}{N(1-\gamma)} \qquad \text{(A.254)} \\
&\leq 16\sqrt{\frac{L''}{(1-\gamma)^2 N}}, \qquad \text{(A.255)}
\end{aligned}
$$

with $L'' = \log\left(\frac{54\|1\|_* SAN^2}{(1-\gamma)\delta}\right)$ where (a) comes from triangular inequality, (b) is due (A.252), for any $\alpha \in \mathbb{R}^S$

$$
\begin{aligned}
\left| \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right) \left([\widehat{V}^{\star,\sigma}_{s,\overline{u}}]_\alpha - [\widehat{V}^{\star,\sigma}_{s,u^\star}]_\alpha \right) \right| &\leq \left\| P^0_{s,a} - \widehat{P}^0_{s,a} \right\|_1 \left\| [\widehat{V}^{\star,\sigma}_{s,\overline{u}}]_\alpha - [\widehat{V}^{\star,\sigma}_{s,u^\star}]_\alpha \right\|_\infty \\
&\leq 2 \left\| \widehat{V}^{\star,\sigma}_{s,\overline{u}} - \widehat{V}^{\star,\sigma}_{s,u^\star} \right\|_\infty \leq \frac{2\varepsilon_2}{(1-\gamma)}, \qquad \text{(A.256)}
\end{aligned}
$$

(c) follows from (A.250), (d) holds using Lemma 6.1 with (A.252). Here, the two last inequalities hold by letting $\varepsilon_2 = \frac{2\log\left(\frac{18\|1\|_* SAN|N_{\varepsilon_2}|}{\delta}\right)}{N}$, which gives $|N_{\varepsilon_2}| \leq \frac{3}{\varepsilon_2(1-\gamma)} \leq \frac{3N}{1-\gamma}$, and the last inequality holds by the fact $\mathrm{Var}_{P^0_{s,a}}(\widehat{V}^{\star,\sigma}) \leq \|\widehat{V}^{\star,\sigma}\|_\infty \leq \frac{1}{1-\gamma}$ and letting $N \geq 2\log\left(\frac{54\|1\|_* SAN^2}{(1-\gamma)\delta}\right) = L''$.

Rewriting (A.235), the first term of the max is controlled.

$$
\max\left\{ \left| \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right) [\widehat{V}^{\widehat{\pi},\sigma}]_{\alpha^{\lambda*}_{P_{s,a}}} \right|, \left| \left(P^0_{s,a} - \widehat{P}^0_{s,a}\right) [\widehat{V}^{\widehat{\pi},\sigma}]_{\alpha^{\lambda*}_{\widehat{P}_{s,a}}} \right| \right\}
$$

The second term can be controlled by the same term as the first one plus an additional term with

$$\left|\left(P^0_{s,a}-\widehat{P}^0_{s,a}\right)\left[\widehat{V}^{\widehat{\pi},\sigma}\right]_{\alpha^{\lambda*}_{\widehat{P}_{s,a}}}\right| \leq$$
$$\left|\max_{\mu^{\lambda}_{P^0_{s,a}}\in\mathcal{M}^{\lambda}_{P^0_{s,a}}}\left(P^0_{s,a}-\widehat{P}^0_{s,a}\right)(\widehat{V}^{\star,\sigma}-\mu^{\lambda}_{P^0_{s,a}})+\max_{\mu^{\lambda}_{\widehat{P}^0_{s,a}}\in\mathcal{M}^{\lambda}_{\widehat{P}^0_{s,a}}}\left(P^0_{s,a}-\widehat{P}^0_{s,a}\right)(\mu^{\lambda}_{P^0_{s,a}}-\mu^{\lambda}_{\widehat{P}^0_{s,a}})\right|$$

and similarly to previous lemma in (A.208), the residual or term in the right in the previous equation can be controlled with $\frac{L'C_S\|1\|_*}{N(1-\gamma)}$ Finally, putting (A.254) and (A.255) back into Equation (A.240) and using Eq. (A.255) with probability at least $1-\delta$ we obtain

$$\begin{aligned}\left|\widehat{P}^{\widehat{\pi},\widehat{V}}_{s,a}\widehat{V}^{\widehat{\pi},\sigma}-P^{\widehat{\pi},\widehat{V}}_{s,a}\widehat{V}^{\widehat{\pi},\sigma}\right| &\leq \max_{\alpha^{\lambda,\omega}_{P_{s,a}}\in\mathrm{A}^{\lambda,\omega}_{P_{s,a}}}\left|\left(P^0_{s,a}-\widehat{P}^0_{s,a}\right)\left[\widehat{V}^{\star,\sigma}\right]_{\alpha^{\lambda,\omega}_{P_{s,a}}}\right|+2\varepsilon_{\mathsf{opt}}\\ &\leq 2\sqrt{\frac{L'}{N}}\sqrt{\mathrm{Var}_{P^0_{s,a}}(\widehat{V}^{\star,\sigma})}+2\varepsilon_{\mathsf{opt}}+\frac{14L''C_S\,\|1\|_*}{N(1-\gamma)}\\ &\leq 2\sqrt{\frac{L''}{(1-\gamma)^2N}}+2\varepsilon_{\mathsf{opt}}+\frac{14L''C_S\,\|1\|_*}{N(1-\gamma)},\end{aligned}\tag{A.257}$$

$\forall(s,a)\in\mathcal{S}\times\mathcal{A}$. Using matrix form we obtain finally:

$$\begin{aligned}\left|\underline{\widehat{P}}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma}-\underline{P}^{\widehat{\pi},\widehat{V}}\widehat{V}^{\widehat{\pi},\sigma}\right| &\leq 2\sqrt{\frac{L''}{N}}\sqrt{\mathrm{Var}_{P^0_{s,a}}(\widehat{V}^{\star,\sigma})}1+2\varepsilon_{\mathsf{opt}}1\\ &\leq 2\sqrt{\frac{L''}{(1-\gamma)^2N}}1+2\varepsilon_{\mathsf{opt}}1.+\frac{14L''C_S\,\|1\|_*}{N(1-\gamma)}1\end{aligned}\tag{A.258}$$

The proof is similar in the s-rectangular case, factorising by $\pi(a|s)$, like in in 7.4. Moreover, the proof is similar for TV without the geometric term depending on C_S.

7.3.6 Proof of Lemma 7.7

We always use the same manner as in Appendix 7.3.4. Similarly to (A.216), it holds:

$$\left(I-\gamma\underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}\sqrt{\mathrm{Var}_{\underline{P}^{\widehat{\pi},\widehat{V}}}(\widehat{V}^{\widehat{\pi},\sigma})}\leq\sqrt{\frac{1}{1-\gamma}}\sqrt{\sum_{t=0}^{\infty}\gamma^t\left(\underline{P}^{\widehat{\pi},\widehat{V}}\right)^t\mathrm{Var}_{\underline{P}^{\widehat{\pi},\widehat{V}}}(\widehat{V}^{\widehat{\pi},\sigma})}.\tag{A.259}$$

In order to upper bound $\mathrm{Var}_{\underline{P}^{\widehat{\pi},\widehat{V}}}(\widehat{V}^{\widehat{\pi},\sigma})$, we define $V'\coloneqq\widehat{V}^{\widehat{\pi},\sigma}-\eta 1$ with $\eta\in\mathbb{R}$. Using as (A.223), it holds

$$\begin{aligned}&\mathrm{Var}_{\underline{P}^{\widehat{\pi},\widehat{V}}}(\widehat{V}^{\widehat{\pi},\sigma})\leq\underline{P}^{\widehat{\pi},\widehat{V}}\left(V'\circ V'\right)-\frac{1}{\gamma}V'\circ V'+\frac{2}{\gamma^2}\|V'\|_\infty 1+\frac{2}{\gamma}\|V'\|_\infty\left|\left(\underline{\widehat{P}}^{\widehat{\pi},\widehat{V}}-\underline{P}^{\widehat{\pi},\widehat{V}}\right)\widehat{V}^{\widehat{\pi},\sigma}\right|\\ &\leq\underline{P}^{\widehat{\pi},\widehat{V}}\left(V'\circ V'\right)-\frac{1}{\gamma}V'\circ V'+ \qquad\text{(A.260)}\\ &\frac{2}{\gamma^2}\|V'\|_\infty 1+\frac{2}{\gamma}\|V'\|_\infty\left(2\sqrt{\frac{L''}{(1-\gamma)^2N}}+2\varepsilon_{\mathsf{opt}}+\frac{14L''C_S\,\|1\|_*}{N(1-\gamma)}\right)1, \qquad\text{(A.261)}\end{aligned}$$

where the last inequality makes use of Lemma 7.6. Plugging (A.261) back into (A.259) leads to

$$\begin{aligned}
&\left(I-\gamma \underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}\sqrt{\mathrm{Var}_{\underline{P}^{\widehat{\pi},\widehat{V}}}(\widehat{V}^{\widehat{\pi},\sigma})} \overset{(a)}{\leq} \sqrt{\frac{1}{1-\gamma}}\sqrt{\left|\sum_{t=0}^{\infty}\gamma^t\left(\underline{P}^{\widehat{\pi},\widehat{V}}\right)^t\left(\underline{P}^{\widehat{\pi},\widehat{V}}\left(V'\circ V'\right)-\frac{1}{\gamma}V'\circ V'\right)\right|}\\
&+\sqrt{\frac{1}{(1-\gamma)^2\gamma^2}\left(2\sqrt{\frac{L''}{(1-\gamma)^2N}}+2\varepsilon_{\mathsf{opt}}+\frac{14L''C_S\left\|1\right\|_*}{N(1-\gamma)}\right)\|V'\|_\infty 1}\\
&\overset{(b)}{\leq}\sqrt{\frac{\|V'\|_\infty^2}{\gamma(1-\gamma)}}1+\sqrt{\frac{\left(2\sqrt{\frac{L''}{(1-\gamma)^2N}}+2\varepsilon_{\mathsf{opt}}+\frac{14L''C_S\|1\|_*}{N(1-\gamma)}\right)\|V'\|_\infty}{(1-\gamma)^2\gamma^2}1}\\
&\overset{(c)}{\leq}\sqrt{\frac{\|V'\|_\infty^2}{\gamma(1-\gamma)}}1+5\sqrt{\left(1+\varepsilon_{\mathsf{opt}}+\frac{L''C_S\left\|1\right\|_*}{N(1-\gamma)}\right)\frac{\|V'\|_\infty}{(1-\gamma)^2\gamma^2}1} \qquad \text{(A.262)}\\
&\leq 6\sqrt{\left(1+\varepsilon_{\mathsf{opt}}+\frac{L''C_S\left\|1\right\|_*}{N(1-\gamma)}\right)\frac{\|V'\|_\infty}{(1-\gamma)^2\gamma^2}1}, \qquad \text{(A.263)}
\end{aligned}$$

where (a) is the same as (A.229), (b) holds by repeating the argument of (A.230), (c) follows by taking $N \geq \frac{L''}{(1-\gamma)^2}$ and then the last inequality holds by $\|V'\|_\infty \leq \|V^{\star,\sigma}\|_\infty \leq \frac{1}{1-\gamma}$. Then taking the infimum over η in the right-hand side of the equation in the definition of V' and using $\mathrm{sp}(.)_\infty \leq \|.\|_*$ gives

$$\left(I-\gamma \underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}\sqrt{\mathrm{Var}_{\underline{P}^{\widehat{\pi},\widehat{V}}}(\widehat{V}^{\widehat{\pi},\sigma})} \leq 6\sqrt{\left(1+\varepsilon_{\mathsf{opt}}+\frac{L''C_S\left\|1\right\|_*}{N(1-\gamma)}\right)\frac{\mathrm{sp}(V)_\infty}{(1-\gamma)^2\gamma^2}1}$$

Finally, applying Lemma 7.1 with $P=\widehat{P}^0$ and $\pi=\widehat{\pi}$ yields

$$\mathrm{sp}(\widehat{V}^{\widehat{\pi},\sigma})_* \leq \frac{1}{\gamma\max\{1-\gamma,\gamma C_g\sigma\}}, \qquad \text{(A.264)}$$

for *sa*-rectangular or

$$\mathrm{sp}(\widehat{V}^{\widehat{\pi},\sigma})_* \leq \frac{1}{\gamma\max\{1-\gamma,\min_s\|\hat{\pi}\|_*\,\tilde{\sigma}\}}$$

in the *s*-rectangular case, which can be inserted into (A.263) and gives in *sa*-rectangular case:

$$\begin{aligned}
\left(I-\gamma \underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}\sqrt{\mathrm{Var}_{\underline{P}^{\widehat{\pi},\widehat{V}}}(\widehat{V}^{\widehat{\pi},\sigma})} &\leq 6\sqrt{\frac{\left(1+\varepsilon_{\mathsf{opt}}+\frac{L''C_S\|1\|_*}{N(1-\gamma)}\right)}{\gamma^3(1-\gamma)^2\max\{1-\gamma,\sigma\}}1}\\
&\leq 6\sqrt{\frac{\left(1+\varepsilon_{\mathsf{opt}}+\frac{L''C_S\|1\|_*}{N(1-\gamma)}\right)}{(1-\gamma)^3\gamma^3}1}
\end{aligned}$$

where first inequalities comes from that we can bound it Eq. left-hand side of equation (A.263) by $\|V'\|_\infty \leq \|V^{\star,\sigma}\|_\infty \leq \frac{1}{1-\gamma}$. Proof for *s*-rectangular is similar, but requires adding an extra factor depending on the norm of the current policy and we have:

$$\left(I-\gamma\underline{P}^{\widehat{\pi},\widehat{V}}\right)^{-1}\sqrt{\mathrm{Var}_{\underline{P}^{\widehat{\pi},\widehat{V}}}(\widehat{V}^{\widehat{\pi},\sigma})}\leq 6\sqrt{\frac{\left(1+\varepsilon_{\mathsf{opt}}+\frac{L''C_S\|1\|_*}{N(1-\gamma)}\right)}{\gamma^3(1-\gamma)^2\max\{1-\gamma,C_g\tilde{\sigma}\min_s\|\hat{\pi}_s\|_\infty\}}}1$$
$$\leq 6\sqrt{\frac{\left(1+\varepsilon_{\mathsf{opt}}+\frac{L''C_S\|1\|_*}{N(1-\gamma)}\right)}{(1-\gamma)^3\gamma^2}}1.$$

7.3.7 Proof of Lemma 7.3

First, if each row of P_π belongs to the simplex $\Delta(S)$, it lead that the row of $(1-\gamma)(I-\gamma P_\pi)^{-1}$ falls into $\Delta(S)$. Then,

$$\begin{aligned}(I-\gamma P_\pi)^{-1}\sqrt{\mathrm{Var}_{P_\pi}(V^{\pi,P})}&=\frac{1}{1-\gamma}(1-\gamma)(I-\gamma P_\pi)^{-1}\sqrt{\mathrm{Var}_{P_\pi}(V^{\pi,P})}\\&\overset{(a)}{\leq}\frac{1}{1-\gamma}\sqrt{(1-\gamma)(I-\gamma P_\pi)^{-1}\mathrm{Var}_{P_\pi}(V^{\pi,P})}\\&=\sqrt{\frac{1}{1-\gamma}}\sqrt{\sum_{t=0}^{\infty}\gamma^t(P_\pi)^t\mathrm{Var}_{P_\pi}(V^{\pi,P})},\end{aligned}\tag{A.265}$$

where (a) is due to Jensen's inequality. Then for any $\eta\in\mathbb{R}^+$, $V'\coloneqq V^{\pi,P}-\eta 1$, we can upper bound $\mathrm{Var}_{P_\pi}(V^{\pi,P})$:

$$\begin{aligned}&\mathrm{Var}_{P_\pi}(V^{\pi,P})\overset{(i)}{=}\mathrm{Var}_{P_\pi}(V')=P_\pi(V'\circ V')-(P_\pi V')\circ(P_\pi V')\\&\overset{(ii)}{\leq}P_\pi(V'\circ V')-\frac{1}{\gamma^2}(V'-r_\pi+(1-\gamma)\eta 1)\circ(V'-r_\pi+(1-\gamma)\eta 1)\\&=P_\pi(V'\circ V')-\frac{1}{\gamma^2}V'\circ V'+\frac{2}{\gamma^2}V'\circ(r_\pi-(1-\gamma)\eta 1)\\&-\frac{1}{\gamma^2}(r_\pi-(1-\gamma)\eta 1)\circ(r_\pi-(1-\gamma)\eta 1)\\&\leq P_\pi(V'\circ V')-\frac{1}{\gamma}V'\circ V'+\frac{2}{\gamma^2}\|V'\|_\infty 1\leq P_\pi(V'\circ V')-\frac{1}{\gamma}V'\circ V'+\frac{2}{\gamma^2}\|V'\|_\infty 1,\end{aligned}\tag{A.266}\tag{A.267}$$

where (i) holds by the fact that $\mathrm{Var}_{P_\pi}(V^{\pi,P}-b1)=\mathrm{Var}_{P_\pi}(V^{\pi,P})$ for any scalar b and $V^{\pi,P}\in\mathbb{R}^S$, (ii) follows from $V'\leq r_\pi+\gamma P_\pi V^{\pi,P}-\eta 1=r_\pi-(1-\gamma)\eta 1+\gamma P_\pi V'$, and the last line arises from $\frac{1}{\gamma^2}V'\circ V'\geq\frac{1}{\gamma}V'\circ V'$ and $\|r_\pi-(1-\gamma)\eta 1\|_\infty\leq 1$. for $\eta\in[0,1/(1-\gamma)[$. Plugging (A.267) back

to (A.265) leads to

$$
\begin{aligned}
&(I-\gamma P_\pi)^{-1}\sqrt{\mathrm{Var}_{P_\pi}(V^{\pi,P})} \le \sqrt{\frac{1}{1-\gamma}}\sqrt{\sum_{t=0}^{\infty}\gamma^t (P_\pi)^t\left(P_\pi(V'\circ V') - \frac{1}{\gamma}V'\circ V' + \frac{2}{\gamma^2}\|V'\|_\infty 1\right)} \\
&\overset{\text{(i)}}{\le} \sqrt{\frac{1}{1-\gamma}}\sqrt{\left|\sum_{t=0}^{\infty}\gamma^t (P_\pi)^t\left(P_\pi(V'\circ V') - \frac{1}{\gamma}V'\circ V'\right)\right|} + \sqrt{\frac{1}{1-\gamma}}\sqrt{\sum_{t=0}^{\infty}\gamma^t (P_\pi)^t \frac{2}{\gamma^2}\|V'\|_\infty 1} \\
&\le \sqrt{\frac{1}{1-\gamma}}\sqrt{\left|\left(\sum_{t=0}^{\infty}\gamma^t (P_\pi)^{t+1} - \sum_{t=0}^{\infty}\gamma^{t-1}(P_\pi)^t\right)(V'\circ V')\right|} + \sqrt{\frac{2\|V'\|_\infty 1}{\gamma^2(1-\gamma)^2}} \\
&\overset{\text{(ii)}}{\le} \sqrt{\frac{\|V'\|_\infty^2 1}{\gamma(1-\gamma)}} + \sqrt{\frac{2\|V'\|_\infty 1}{\gamma^2(1-\gamma)^2}} \\
&\le \sqrt{\frac{8\|V'\|_\infty 1}{\gamma^2(1-\gamma)^2}}, \qquad \text{(A.268)}
\end{aligned}
$$

(A.269)

where (i) holds due to, (ii) holds by following recursion between the two sums, and the last inequality holds because$\|V'\|_\infty \le \frac{1}{1-\gamma}$. Then taking the minimum over η in the right-hand side of the equation gives the result.

$$
(I-\gamma P_\pi)^{-1}\sqrt{\mathrm{Var}_{P_\pi}(V^{\pi,P})} \le \sqrt{\frac{8\mathrm{sp}(V^{\pi,P})_\infty}{\gamma^2(1-\gamma)^2}}
$$

However, we also $\|V'\|_\infty \le \|V^{\pi,P}\|_\infty \le \frac{1}{1-\gamma}$ in (A.268). So finally, the result is

$$
(I-\gamma P_\pi)^{-1}\sqrt{\mathrm{Var}_{P_\pi}(V^{\pi,P})} \le \sqrt{\frac{8}{\gamma^2(1-\gamma)^2}\mathrm{sp}(V^{\pi,P})_\infty 1}.
$$

8 Proof of Theorem 3.4.2

In this section, we focus on the scenarios in the uncertainty sets are constructed with (s,a)-rectangularity condition with some general norms. Towards this, we firstly observe that for the two limiting cases ℓ_1 norm and ℓ_∞ norm, one has $\|p_1-p_2\|_1 \le 2$ and $\|p_1-p_2\|_\infty \le 1$ for any two probability distribution $p_1, p_2 \in \mathbb{R}^S$. Namely, the accessible ranges of the uncertainty level σ for ℓ_1 norm and ℓ_∞ norm are $(0,2]$ and $(0,1]$, respectively. In addition, we have

$$
\forall p_1, p_2 \in \mathbb{R}^S: \quad \|p_1-p_2\|_\infty \le \|p_1-p_2\| \le \|p_1-p_2\|_1 \qquad \text{(A.270)}
$$

for any norm $\|\cdot\|$. It indicates that the accessible range of the uncertainty level $\sigma_{\|\cdot\|}$ for any given norm $\|\cdot\|$ is between $(0, \sigma_{\|\cdot\|}^{\max}]$, where $1 \le \sigma_{\|\cdot\|}^{\max} \le 2$.

To continue, we specify the definition of the uncertainty set with *sa*-rectangularity condition with some given general norm $\|\cdot\|$ as below: for any nominal transition kernel $P \in \mathbb{R}^{SA\times S}$,

$$
\mathcal{U}^\sigma_{\|\cdot\|}(P) \coloneqq \mathcal{U}^\sigma_{\|\cdot\|}(P) = \otimes\, \mathcal{U}^\sigma_p(P_{s,a}), \qquad \mathcal{U}^\sigma_{\|\cdot\|}(P_{s,a}) \coloneqq \left\{P'_{s,a} \in \Delta(\mathcal{S}) : \left\|P'_{s,a} - P_{s,a}\right\| \le \sigma_{\|\cdot\|}\right\}. \qquad \text{(A.271)}
$$

Then, we recall the assumption of the uncertainty radius $\sigma_{\|\cdot\|} \in \left(0, \sigma_{\|\cdot\|}^{\max}(1-c_0)\right]$ with $0 < c_0 < 1$.

Then, resorting to the same class of hard MDPs in (Shi et al. 2023, Section C.1), we can complete the proof by directly following the same proof pipeline of Shi et al. (2023, Section C) by replacing σ with $\sigma_{\|\cdot\|}^{\max}\sigma_{\|\cdot\|}$.

9 Proof of Theorem 3.4.4

Developing the lower bound for the cases with s-rectangular uncertainty set involves several new challenges compared to that of (s,a)-rectangular cases. Specifically, the first challenge is that the optimal policy can be stochastic and hard to be characterized with a closed form for the RMDPs with a s-rectangular uncertainty set, rather than deterministic polices in (s,a)-rectangular cases. Such richer and smoother class of optimal policies makes slightly changing the transition kernel generally could only leads to a smoothly changed stochastic optimal policy instead of a completely different one. Such reduced changing of optimal policy further gives smaller performance gap, thus challenges of a tighter lower bound. Second, most of the hard instances in the literature are constructed as SA states with a constant number of action spaces without loss of generality. While when it comes to s-rectangular uncertainty set, the action space size becomes important and can't be assumed as a constant anymore. So a new class of instances are required.

To address these challenges, in this section, we construct a new set of hard RMDP instances for two limiting cases: ℓ_1 norm and ℓ_∞ norm.

9.1 Construction of the hard problem instances

Before proceeding, we introduce two useful sets related to the state space and action space as below:

$$\mathcal{S} = \{0, 1, \ldots, S\}, \qquad \text{and} \qquad \mathcal{A} = \{0, 1, \cdots, A-1\}.$$

In this section, we construct a set of RMDPs termed as $\mathcal{M}_{\ell_\infty}$, which consists of $S(A-1)$ components including $S(A-1)$ components, each associates with some different state-action pair. Specifically, it is defined as

$$\mathcal{M}_{\ell_\infty} \coloneqq \left\{\mathcal{M}_\theta = (\mathcal{S}, \mathcal{A}, \mathcal{U}^\sigma(P^\theta), r, \gamma) \mid \theta \in \Theta = \left\{(i,j) : (i,j) \in \mathcal{S} \times \mathcal{A} \setminus \{0\}\right\}\right\}. \tag{A.272}$$

We introduce the detailed definition of $\mathcal{M}_{\ell_\infty}$ by introducing several key components of it sequentially. In particular, for any RMDP $\mathcal{M}_\theta \in \mathcal{M}_{\ell_\infty}$, the state space is of size $2S$, which includes two classes of states $\mathcal{X} = \{x_0, x_1, \cdots, x_{S-1}\}$ and $\mathcal{Y} = \{y_0, y_1, \cdots, y_{S-1}\}$. The action space for each state is $\mathcal{A}$ of A possible actions. So we have totally $2S$ states and there is in total $2SA$ state-action pairs.

Armed with the above definitions, we can first introduce the following nominal transition kernel: for all $(s,a) \in \mathcal{X} \cup \mathcal{Y} \times \mathcal{A}$

$$P^{(0,0)}(s' \mid s, a) = \begin{cases} p\mathbb{1}(s' = y_i) + (1-p)\mathbb{1}(s' = x_i) & \text{if} \quad s = x_i, a = 0, \quad \forall i \in \mathcal{S} \\ q\mathbb{1}(s' = y_i) + (1-q)\mathbb{1}(s' = x_i) & \text{if} \quad s = x_i, a \neq 0, \quad \forall i \in \mathcal{S} \\ \mathbb{1}(s' = s) & \text{if} \quad s \in \mathcal{Y} \end{cases} \tag{A.273}$$

Here, p and q are set according to

$$0 \leq p \leq 1 \quad \text{and} \quad 0 \leq q = p - \Delta \tag{A.274}$$

for some p and $\Delta > 0$ that will be introduced momentarily.

Then we introduce the $S(A-1)$ components inside $\mathcal{M}_\infty$. Namely, for any $(i,j) \in \mathcal{S} \times \mathcal{A} \setminus \{0\}$, the nominal transition kernel of $\mathcal{M}_{(i,j)}$ is specified as

$$P^{(i,j)}(s' \,|\, s,a) = \begin{cases} p\mathbb{1}(s'=y_i) + (1-p)\mathbb{1}(s'=x_i) & \text{if } s = x_i, a = j \\ q\mathbb{1}(s'=y_i) + (1-q)\mathbb{1}(s'=x_i) & \text{if } s = x_i \in \mathcal{X}, a = 0 \\ P^{(0,0)}(s' \,|\, s,a) & \text{otherwise} \end{cases} \tag{A.275}$$

In words, the nominal transition kernel of each variant $\mathcal{M}_{(i,j)}$ only differs slightly from that of the basic nominal transition kernel $P^{(0,0)}$ when $s = x_i$ and $a = \{0, j\}$, which makes all the components inside $\mathcal{M}_{\ell_\infty}$ closed to each other.

In addition, the reward function is defined as

$$\forall a \in \mathcal{A}: \quad r(s,a) = \begin{cases} 1 & \text{if } s \in \mathcal{Y} \\ 0 & \text{otherwise.} \end{cases} \tag{A.276}$$

Uncertainty set of the transition kernels. Recall the following useful notation for any transition probability P, i.e., the transition vector associated with some state s is denoted as:

$$P_s \coloneqq P(\cdot,\cdot \,|\, s) \in \mathbb{R}^{1 \times SA}, \quad P_s^0 \coloneqq P^0(\cdot,\cdot \,|\, s) \in \mathbb{R}^{1 \times SA}. \tag{A.277}$$

With this in hand, the uncertainty set (definition in (3.5)) with ℓ_∞ norm for any P^θ with $\theta \in \Theta$ can be represented as:

$$\mathcal{U}_\infty^{\mathsf{s},\widetilde{\sigma}}(P_s^\theta) \coloneqq \mathcal{U}_{\|\cdot\|}^{\mathsf{s},\widetilde{\sigma}}(P_s^\theta) = \left\{ P_s' \in \Delta(\mathcal{S})^{\mathcal{A}} : \left\| P_s' - P_s^\theta \right\| \le \tilde{\sigma} = \sigma \|1\|_\infty = \sigma \right\}. \tag{A.278}$$

So without loss of generality, we set the radius $\sigma \in (0, (1-c_0)]$ with $0 < c_0 < 1$. Before proceeding, we observe that as the uncertainty set above is defined with respect to ℓ_∞, it directly implies that for each $(s,a) \in \mathcal{S} \times \mathcal{A}$, the uncertainty set is independent and can be decomposed as

$$\mathcal{U}_\infty^{\mathsf{s},\widetilde{\sigma}}(P_s^\theta) = \otimes \mathcal{U}_{\|\cdot\|}^{\mathsf{s},\widetilde{\sigma}}(P_{s,a}^\theta) = \left\{ P_{s,a}' \in \Delta(\mathcal{S}) : \left\| P_{s,a}' - P_{s,a}^\theta \right\| \le \sigma \right\}. \tag{A.279}$$

Notably, this indicates that using s-rectangular uncertainty set with ℓ_∞ norm as the divergence function is analogous to the case of using (s,a)-rectangular uncertainty set with ℓ_∞ norm. As a result, we follow the pipeline of the prior art Shi et al. (2023, Section C) which established the minimax-optimal lower bound for (s,a)-rectangular RMDPs with TV distance, which is analogous to the ℓ_∞ case. Towards this, we set p, q, Δ as the same as the ones in Shi et al. (2023, Section C.1), where we recall the expressions of p, q, Δ for self-contained as below: taking $c_1 \coloneqq \frac{c_0}{2}$,

$$p = (1+c_1)\max\{1-\gamma, \sigma\} \qquad \text{and} \qquad \Delta \le c_1 \max\{1-\gamma, \sigma\}, \tag{A.280}$$

which ensure several facts:

$$0 \le p \le 1 \quad \text{and} \quad p \ge q \ge \max\{1-\gamma, \sigma\}. \tag{A.281}$$

Value functions and optimal policies. For each RMDP instance $\mathcal{M}_\theta \in \mathcal{M}_{\ell_\infty}$, with some abuse of notation, we denote $\pi_\theta^\star$ as the optimal policy. In addition, let $V_\theta^{\pi,\sigma}$ (resp. $V_\theta^{\star,\sigma}$) represent the corresponding robust value function of any policy π (resp. $\pi_\theta^\star$) with uncertainty level σ. Armed with these notations, the following lemma shows some essential properties concerning the value functions and optimal policies; the proof is postponed to Appendix 9.3.

Lemma 9.1. *Consider any $\mathcal{M}_\theta \in \mathcal{M}_{\ell_\infty}$ and any policy π, one has*

$$\forall (i,j) \in \Theta: \quad V_{(i,j)}^{\pi,\sigma}(x_i) \le \frac{\gamma\big(z_{(i,j)}^{\pi} - \sigma\big)}{(1-\gamma)\Big(1 + \frac{\gamma\left(z_{(i,j)}^{\pi} - \sigma\right)}{1-\gamma(1-\sigma)}\Big)\big(1-\gamma\,(1-\sigma)\big)}, \tag{A.282}$$

where $z_{(i,j)}^{\pi}$ is defined as

$$\forall (i,j) \in \Theta: \quad z_{(i,j)}^{\pi} \coloneqq p\pi(j \,|\, x_i) + q\,[1 - \pi(j \,|\, x_i)]\,. \tag{A.283}$$

In addition, the robust optimal value functions and the robust optimal policies satisfy

$$\forall (i,j) \in \Theta, s \in \mathcal{X}: \quad V_{(i,j)}^{\star,\sigma}(s) = \frac{\gamma\,(p-\sigma)}{(1-\gamma)\left(1 + \frac{\gamma(p-\sigma)}{1-\gamma(1-\sigma)}\right)\big(1-\gamma\,(1-\sigma)\big)} \tag{A.284}$$

and

$$\pi_{(i,j)}^\star(j \,|\, x_i) = 1 \qquad \text{and} \qquad \pi_{(i,j)}^\star(0 \,|\, s) = 1 \quad \forall s \in \mathcal{X} \setminus \{x_i\}. \tag{A.285}$$

In words, this lemma shows that for any RMDP $\mathcal{M}_{(i,j)}$, the optimal policy on state x_i satisfies $\pi_{(i,j)}^\star(j \,|\, x_i) = 1$ and will focus on $a = 0$ for all other states $s \in \mathcal{X} \setminus \{x_i\}$.

9.2 Establishing the minimax lower bound

Step 1: converting the goal to estimate (i,j). Now we are in position to derive the lower bound. Recall the goal is to control the following quantity associated with any policy estimator $\widehat{\pi}$ based on the dataset with in total N_{all} samples:

$$\max_{(i,j)\in\Theta} \mathbb{P}_{(i,j)}\left\{\max_{s\in\mathcal{X}\cup\mathcal{Y}}\left(V_{(i,j)}^{\star,\sigma}(s) - V_{(i,j)}^{\widehat{\pi},\sigma}(s)\right)\right\} \ge \max_{(i,j)\in\Theta} \mathbb{P}_{(i,j)}\left\{\max_{s\in\mathcal{X}}\left(V_{(i,j)}^{\star,\sigma}(s) - V_{(i,j)}^{\widehat{\pi},\sigma}(s)\right)\right\}. \tag{A.286}$$

To do so, we can invoke a key claim in Shi et al. (2023) here since our problem setting can be reduced to the same one in Shi et al. (2023): With $\varepsilon \le \frac{c_1}{32(1-\gamma)}$, letting

$$\Delta = 32(1-\gamma)\max\{1-\gamma,\sigma\}\varepsilon \le c_1 \max\{1-\gamma,\sigma\} \tag{A.287}$$

which satisfies (A.280), it leads to that for any policy $\widehat{\pi}$ and all $(i,j) \in \Theta$,

$$\begin{aligned} V_{(i,j)}^{\star,\sigma}(x_i) - V_{(i,j)}^{\widehat{\pi},\sigma}(x_i) &\ge 2\varepsilon(1 - \widehat{\pi}(j \,|\, x_i)), \\ \forall s \in \mathcal{X}\setminus\{x_i\}: \quad V_{(i,j)}^{\star,\sigma}(s) - V_{(i,j)}^{\widehat{\pi},\sigma}(s) &\ge 2\varepsilon(1 - \widehat{\pi}(0 \,|\, s)). \end{aligned} \tag{A.288}$$

Before continuing, we introduce a useful notation for the subset of Θ excluding the cases with state i is selected:

$$\forall i \in \mathcal{S}: \quad \Theta_{-i} = \Theta \setminus \{(i',j) : i' = i, j \in \mathcal{A}\setminus\{0\}\}. \tag{A.289}$$

Armed with the above facts and notations, we first suppose there exists a policy $\widehat{\pi}$ such that for some $(i,j) \in \Theta$,

$$\mathbb{P}_{(i,j)}\left\{V_{(i,j)}^{\star,\sigma}(x_i) - V_{(i,j)}^{\widehat{\pi},\sigma}(x_i) \le \varepsilon\right\} \ge \frac{3}{4}. \tag{A.290}$$

which in view of (A.288) indicates that we necessarily have $\widehat{\pi}(j \,|\, x_i) \ge \frac{1}{A}$ with probability at least $\frac{3}{4}$.

As a result, taking

$$j' = \arg\max_{a \in \mathcal{A}} \widehat{\pi}(a \,|\, x_i), \tag{A.291}$$

we are motivated to construct the following estimate of θ:

$$\widehat{\theta}\begin{cases} = (i, j') & \text{if} \quad j' > 0 \\ \in \mathcal{G}_{-w} & \text{if} \quad j' = 0, \end{cases} \tag{A.292}$$

which satisfies

$$\mathbb{P}_{(i,j)}\{\widehat{\theta} = (i,j)\} \ge \mathbb{P}_{(i,j)}\{j' = j\} \ge \mathbb{P}_{(i,j)}\{\widehat{\pi}(j \,|\, x_i) > \frac{1}{A}\} \ge \frac{3}{4}. \tag{A.293}$$

Step 2: developing the probability of error in testing multiple hypotheses. Before proceeding, we discuss the dataset consisting of in total N_{all} independent samples. Observing that each RMDP inside the set $\mathcal{M}_{\ell_\infty}$ are constructed symmetrically associated with one pair of states (x_i, y_i) for all $i \in \mathcal{S}$ and another action $j \in \mathcal{A} \times \{0\}$, respectively. Therefore, it is obvious that the dataset is supposed to be generated uniformly on each (x_i, y_i, j) to maximize the information gain, leading to $\frac{N_{\mathsf{all}}}{S(A-1)}$ samples for any states-action (x_i, y_i, j) with $i \in \mathcal{S}, j \in \mathcal{A} \setminus \{0\}$.

Then we are ready to turn to the hypothesis testing problem over $(i,j) \in \Theta$. Towards this, we consider the minimax probability of error defined as follows:

$$p_{\mathrm{e}} \coloneqq \inf_{\phi} \max_{(i,j)\in\Theta} \{\mathbb{P}_{(i,j)}(\phi \neq (i,j))\}, \tag{A.294}$$

where the infimum is taken over all possible tests ϕ constructed from the dataset introduced above.

To continue, armed with the above dataset with N_{all} independent samples, we denote $\mu^{i,j}$ (resp. $\mu^{i,j}(s,a)$) as the distribution vector (resp. distribution) of each sample tuple (s,a,s') under the nominal transition kernel $P^{(i,j)}$ associated with $\mathcal{M}_{(i,j)}$. With this in mind, combined with Fano's inequality from Tsybakov (2009, Theorem 2.2) and the additivity of the KL divergence (cf. Tsybakov (2009, Page 85)), we obtain

$$\begin{aligned} p_{\mathrm{e}} &\ge 1 - N_{\mathsf{all}} \frac{\max_{(i,j),(i',j')\in\Theta,(i,j)\neq(i',j')} \mathsf{KL}(\mu^{i,j} \,|\, \mu^{i',j'}) + \log 2}{\log|\Theta|} \\ &\overset{\text{(i)}}{\ge} 1 - N_{\mathsf{all}} \max_{(i,j),(i',j')\in\Theta,(i,j)\neq(i',j')} \mathsf{KL}(\mu^{i,j} \,|\, \mu^{i',j'}) - \frac{1}{2} \\ &= \frac{1}{2} - N_{\mathsf{all}} \max_{(i,j),(i',j')\in\Theta,(i,j)\neq(i',j')} \mathsf{KL}(\mu^{i,j} \,|\, \mu^{i',j'}) \end{aligned} \tag{A.295}$$

where (i) holds by $\log|\Theta| \ge 2\log 2$ as long as $S(A-1)$ are large enough. Then following the same proof pipeline of Shi et al. (2023, Section C.2), we can arrive at

$$p_{\mathrm{e}} \ge \frac{1}{2} - \frac{N_{\mathsf{all}}}{S(A-1)} \frac{4096}{c_1} (1-\gamma)^2 \max\{1-\gamma, \sigma\} \varepsilon^2 \ge \frac{1}{4}, \tag{A.296}$$

if the sample size is selected as

$$N_{\mathsf{all}} \leq \frac{c_1 S(A-1)}{16396(1-\gamma)^2 \max\{1-\gamma,\sigma\}\varepsilon^2}. \tag{A.297}$$

Step 3: summing up the results together. Finally, we suppose that there exists an estimator $\widehat{\pi}$ such that

$$\max_{(i,j)\in\Theta} \mathbb{P}_{(i,j)}\left[\max_{s\in\mathcal{X}\cup\mathcal{Y}}\left(V_{(i,j)}^{\star,\sigma}(s) - V_{(i,j)}^{\widehat{\pi},\sigma}(s)\right) \geq \varepsilon\right] < \frac{1}{4}, \tag{A.298}$$

then according to (A.286), we necessarily have

$$\forall s \in \mathcal{X}: \quad \max_{(i,j)\in\Theta} \mathbb{P}_{(i,j)}\left[V_{(i,j)}^{\star,\sigma}(s) - V_{(i,j)}^{\widehat{\pi},\sigma}(s) \geq \varepsilon\right] < \frac{1}{4}, \tag{A.299}$$

which indicates

$$\forall s \in \mathcal{X}: \quad \max_{(i,j)\in\Theta} \mathbb{P}_{(i,j)}\left[V_{(i,j)}^{\star,\sigma}(s) - V_{(i,j)}^{\widehat{\pi},\sigma}(s) < \varepsilon\right] \geq \frac{3}{4}. \tag{A.300}$$

As a consequence, (A.293) shows we must have

$$\forall (i,j) \in \Theta: \quad \mathbb{P}_{(i,j)}\left[\widehat{\theta} = (i,j)\right] \geq \frac{3}{4} \tag{A.301}$$

to achieve (A.298). However, this would contract with (A.296) if the sample size condition in (A.297) is satisfied. Thus, we complete the proof.

9.3 Proof of Lemma 9.1

Without loss of generality, we first consider any $\mathcal{M}_{(i,j)}$ with $(i,j) \in \mathcal{S}\times\mathcal{A}\setminus\{0\}$. Following the same routine of Shi et al. (2023, Section C.3.1), we can verify that the order of the robust value function $V_{(i,j)}^{\pi,\sigma}$ over different states satisfies

$$\forall k \in \mathcal{S}: \quad V_{(i,j)}^{\pi,\sigma}(x_k) \leq V_{(i,j)}^{\pi,\sigma}(y_k), \tag{A.302}$$

which means the robust value function of the states inside $\mathcal{X}$ are always not larger than the corresponding states inside $\mathcal{Y}$.

Then we denote the minimum of the robust value function over states as below:

$$V_{(i,j),\min}^{\pi,\sigma} \coloneqq \min_{s\in\mathcal{S}} V_{(i,j)}^{\pi,\sigma}(s). \tag{A.303}$$

In the following arguments, we first take a moment to assume $V_{(i,j),\min}^{\pi,\sigma} = V_{(i,j)}^{\pi,\sigma}(x_i)$. With this in mind, we arrive at

$$V_{(i,j)}^{\pi,\sigma}(y_i) = 1 + \gamma(1-\sigma)V_{(i,j)}^{\pi,\sigma}(y_i) + \gamma\sigma V_{(i,j),\min}^{\pi,\sigma} = \frac{1+\gamma\sigma V_{(i,j)}^{\pi,\sigma}(x_i)}{1-\gamma(1-\sigma)}. \tag{A.304}$$

Then, when we move on to the characterization of the robust value function at state x_i. To do so, we notice two important facts:

1) The nominal transition probability $P_{x_i,a}^{(i,j)}$ at state-action pair (x_i,a) for any $a\in\mathcal{A}$ is a Bernoulli distribution (see (A.275) and (A.273)). The TV distance and the ℓ_∞ norm between two Bernoulli distribution are the same.

2) Invoking the definitions of the nominal transition probability in (A.275) and (A.273), we have

$$\begin{aligned} P_{x_i,j}^{(i,j)} &= p\mathbb{1}(s' = y_i) + (1-p)\mathbb{1}(s' = x_i) \\ P_{x_i,a}^{(i,j)} &= q\mathbb{1}(s' = y_i) + (1-q)\mathbb{1}(s' = x_i) \quad \forall a \in \mathcal{A} \setminus \{j\}. \end{aligned} \tag{A.305}$$

With the above two facts in hand, our problem setting is reduced to the same one in Shi et al. (2023) and can reuse the results in Shi et al. (2023, Section C.3.1) to achieve

$$V_{(i,j)}^{\pi,\sigma}(x_i) \leq \frac{\frac{\gamma\left(z_{(i,j)}^{\pi} - \sigma\right)}{1-\gamma(1-\sigma)}}{(1-\gamma)\left(1 + \frac{\gamma\left(z_{(i,j)}^{\pi} - \sigma\right)}{1-\gamma(1-\sigma)}\right)}. \tag{A.306}$$

and

$$\begin{aligned} &\pi_{(i,j)}^{\star}(j \,|\, x_i) = 1 \\ &V_{(i,j)}^{\star,\sigma}(x_i) = \frac{\frac{\gamma\left(z_{(i,j)}^{\pi^\star} - \sigma\right)}{1-\gamma(1-\sigma)}}{(1-\gamma)\left(1 + \frac{\gamma\left(z_{(i,j)}^{\pi^\star} - \sigma\right)}{1-\gamma(1-\sigma)}\right)} = \frac{\frac{\gamma(p-\sigma)}{1-\gamma(1-\sigma)}}{(1-\gamma)\left(1 + \frac{\gamma(p-\sigma)}{1-\gamma(1-\sigma)}\right)}. \end{aligned} \tag{A.307}$$

Analogously, we can verify that for other $x_k \in \mathcal{X} \setminus \{x_i\}$,

$$\begin{aligned} &\pi_{(i,j)}^{\star}(0 \,|\, x_k) = 1 \\ &V_{(i,j)}^{\star,\sigma}(x_k) = \frac{\frac{\gamma(p-\sigma)}{1-\gamma(1-\sigma)}}{(1-\gamma)\left(1 + \frac{\gamma(p-\sigma)}{1-\gamma(1-\sigma)}\right)}. \end{aligned} \tag{A.308}$$

10 DRVI for $sa-$ rectangular algorithm for arbitrary norm

In order to compute the fixed point of $\widehat{\mathcal{T}}^{\sigma}$, distributionally robust value iteration (DRVI), is defined in Algorithm 11. For sa-rectangularity, starting from an initialization $\widehat{Q}_0 = 0$, the update rule at the t-th ($t \geq 1$) iteration is the following $\forall (s,a) \in \mathcal{S} \times \mathcal{A}$:

$$\widehat{Q}_t^{\pi}(s,a) = \widehat{\mathcal{T}}^{\sigma}\widehat{Q}_{t-1}^{\pi}(s,a) = r(s,a) + \gamma \inf_{\mathcal{P} \in \mathcal{U}_{\|.\|}^{\mathsf{sa},\sigma}(\widehat{P}_{s,a}^0)} \mathcal{P}\widehat{V}_{t-1}, \tag{A.309}$$

where $\widehat{V}_{t-1}(s) = \max_{\pi} \widehat{Q}_{t-1}^{\pi}(s,a)$ for all $s \in \mathcal{S}$.

Directly solving (A.309) is computationally expensive since it involves optimization over a S-dimensional probability simplex at each iteration, especially when the dimension of the state space $\mathcal{S}$ is large. Fortunately, given strong duality (A.309) can be equivalently solved using its dual problem, which concerns optimizing a two variable (λ and ω) and thus can be solved efficiently. The specific form of the dual problem depends on the choice of the norm $\|.\|$, which we shall discuss separately in Appendix 6.3. To complete the description, we output the greedy policy of the final Q-estimate $\widehat{Q}_T$ as the final policy $\widehat{\pi}$, namely,

$$\forall s \in \mathcal{S}: \quad \widehat{\pi}(s) = \arg\max_{a} \widehat{Q}_T(s,a). \tag{A.310}$$

Algorithm 11: Distributionally robust value iteration ($DRVI$) for infinite-horizon RMDPs for sa-rectangular for arbitrary norm

input: empirical nominal transition kernel $\widehat{P}^0$; reward function r; uncertainty level σ; number of iterations T.
initialization: $\widehat{Q}_0(s,a) = 0$, $\widehat{V}_0(s) = 0$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$.
for $t = 1, 2, ..., T$ **do**
  for $s \in \mathcal{S}, a \in \mathcal{A}$ **do**
    Set $\widehat{Q}_t(s,a)$ according to (A.309);
  for $s \in \mathcal{S}$ **do**
    Set $\widehat{V}_t(s) = \max_a \widehat{Q}_t(s,a)$;
output: $\widehat{Q}_T$, $\widehat{V}_T$ and $\widehat{\pi}$ obeying $\widehat{\pi}(s) := \arg\max_a \widehat{Q}_T(s,a)$.

Encouragingly, the iterates $\{\widehat{Q}_t\}_{t\geq 0}$ of $DRVI$ converge linearly to the fixed point $\widehat{Q}^{\star,\sigma}$, owing to the appealing γ-contraction property of $\widehat{\mathcal{T}}^{\sigma}$.

Using Algorithm 11, it allows getting an ϵ_{opt} error in the empirical MDP in the sa-rectangular case. In the s-rectangular case, finding an algorithm to get ϵ_{opt} is more difficult to use, as the policy is not deterministic anymore and 11 cannot anymore be applied. For L_p norms, Clavier et al. (2023) derived an algorithm but for arbitrary norm we need to consider a more general problem for arbitrary norm in Appendix 8

Appendix of Chapter 4

11 Proof of mean-standard deviation formulation as a robust problem

We consider the following equality for $\mathbb{P}$ the distribution of trajectories following π and nominal kernel P^0:

$$\min_{\mathbb{P}\in D_{\chi^2}(\mathbb{P}\|\mathbb{P}^0)\leq\alpha} Q^{(P,\pi)}(s,a) = Q^{(P^0,\pi)}(s,a) - \alpha^{1/2}\mathbb{V}_{P^0}[Z(s,a)]^{\frac{1}{2}} \tag{A.311}$$

Consider here that τ is drawn from $\mathbb{P}$. Writing $\tilde{R}(\tau) = R(\tau) - E_{\tau\sim P^0}[R(\tau)]$

$$\begin{aligned}
\|E_{\tau\sim\mathbb{P}}[R(\tau)] - E_{\tau\sim\mathbb{P}^0}[R(\tau)]\| &= \left\|\int_\tau \tilde{R}(\tau)(\mathbb{P}(\tau) - \mathbb{P}^0(\tau))d\tau\right\| \\
&= \left\|\int_\tau \tilde{R}(\tau)\sqrt{\mathbb{P}^0(\tau)}\frac{(\mathbb{P}(\tau) - \mathbb{P}^0(\tau))}{\sqrt{\mathbb{P}^0(\tau)}}d\tau\right\| \\
&\leq \left\|\int_\tau \tilde{R}(\tau)^2\mathbb{P}^0(\tau)d\tau\right\|^{\frac{1}{2}}\left\|\int_\tau \frac{(\mathbb{P}(\tau) - \mathbb{P}^0(\tau))^2}{\mathbb{P}^0(\tau)}d\tau\right\|^{\frac{1}{2}} \\
&= \mathbb{V}_{\mathbb{P}^0}[R(\tau)]^{\frac{1}{2}} D_{\chi^2}(\mathbb{P}\|\mathbb{P}^0)^{\frac{1}{2}}
\end{aligned}$$

because of positivity of divergence and of the variance, norms are removed. We get equality if for $\lambda \in \mathbb{R}$:

$$\tilde{R}(\tau)\mathbb{P}^0(\tau) = \lambda(\mathbb{P}(\tau) - \mathbb{P}^0(\tau)) \iff \mathbb{P}(\tau) = \mathbb{P}^0(\tau)(1 + \frac{1}{\lambda}\tilde{R}(\tau)) \tag{A.312}$$

However, $\mathbb{P}(\tau)$ needs to be positive and sum to one as it is a measure. The last condition is respected but if $\lambda \leq 0$ we need $\left\|\tilde{R}(\tau)/\lambda\right\| \leq 1$ to ensure positivity of $\mathbb{P}(\tau)$. In this case we obtain from A.312 that $D_{\chi^2}(\mathbb{P}\|\mathbb{P}^0) = \frac{\mathbb{V}_{\mathbb{P}^0}R}{\lambda^2}$. Replacing the divergene in the inequality, the following result holds :

$$\|E_{\tau\sim\mathbb{P}}[R(\tau)] - E_{\tau\sim\mathbb{P}^0}[R(\tau)]\| \leq \frac{\mathbb{V}_{\mathbb{P}^0}(R(\tau))}{\lambda}$$

So for a constrained problem such that $D_{\chi^2}(\mathbb{P}\|\mathbb{P}^0) \leq \alpha$, we obtain:

$$\min_{\mathbb{P}\in D_{\chi^2}(\mathbb{P}\|\mathbb{P}^0)\leq\alpha} Q^{(P,\pi)} = Q^{(P^0,\pi)} - \alpha^{1/2}\mathbb{V}[Z(s,a)]^{\frac{1}{2}}$$

with the maximum value of α equals to $D_{\chi^2}(\mathbb{P}\|\mathbb{P}^0) = \frac{\mathbb{V}_{\mathbb{P}^0}[\tilde{R}(\tau)]}{\lambda^2} \leq \frac{\mathbb{V}_{\mathbb{P}^0}[R(\tau)]}{\|\tilde{R}\|_\infty^2} = \frac{\|\tilde{R}\|_2^2}{\|\tilde{R}\|_\infty^2} \leq 1$ If our problem is constrained, we obtain the following results with the maximum attained for $D_{\chi^2}(\mathbb{P}\|\mathbb{P}^0) = \alpha$:

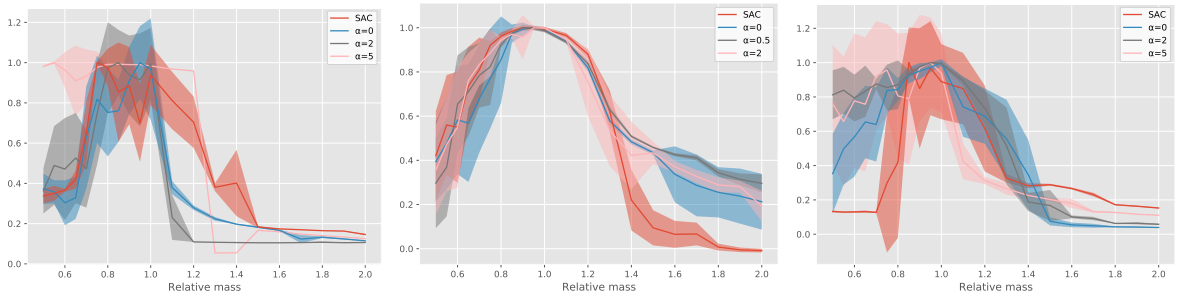
$$\min_{\mathbb{P}\in D_{\chi^2}(\mathbb{P}\|\mathbb{P}^0)\leq\alpha} Q^{(P,\pi)} = Q^{(P_0,\pi)} - \alpha^{1/2}\mathbb{V}[Z(s,a)]^{\frac{1}{2}} \tag{A.313}$$

and the formulation of our algorithm becomes :

$$\begin{cases} \pi_{k+1} \in \mathcal{G}_\alpha\left(Z_k\right) = \mathcal{G}(\xi_\alpha(Z_k) = \arg\max_{\pi\in\Pi.}\langle\mathbb{E}[Z_k] - \alpha^{1/2}\sqrt{\mathbb{V}[Z_k]}, \pi\rangle \\ Z_{k+1} = \left(T^{\pi_{k+1}}\right)^m Z_k \end{cases},$$

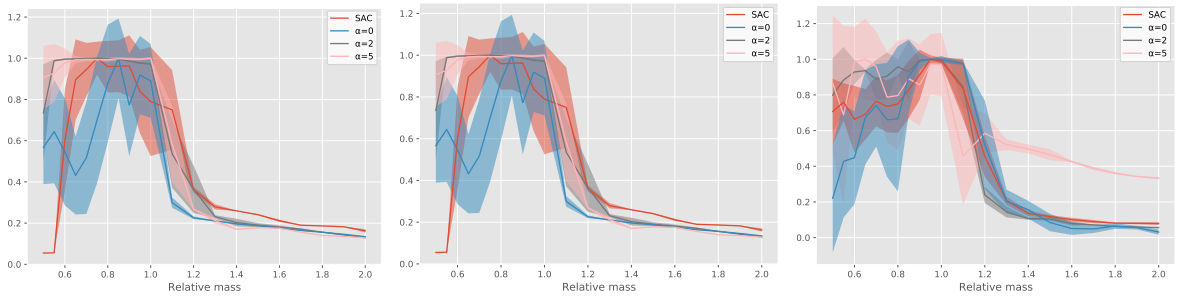
12 Further results on continuous action space

12.1 Normalised results



(a) Hopper-v3 (**wN**) **(b)** HalfCheetah-v3 (**wN**) **(c)** Walker-v3 (**wN**)

Figure A12.1: y-axis : normalised mean ± standard deviation over 20 trajectories. x-axis : relative mass.



(a) Hopper-v3 (**N**) **(b)** HalfCheetah-v3 (**N**) **(c)** Walker-v3 (**N**)

Figure A12.2: y-axis : normalised mean ± standard deviation over 20 trajectories. x-axis : relative mass.

The results were normalised to better reflect the improvement without being biased by the average performance which is higher with a distributional critic.

13 Further Experimental Details

All experiments were run on a cluster containing an Intel Xeon CPU Gold 6230, 20 cores, and all experiments were performed on a single CPU between 3 and 6 hours for continuous control and less than 1 hour for the discrete control environment.

Pre-trained models will be available for all algorithms and environments on a GitHub link.

The Mujoco OpenAI Gym task licensing information is given at https://github.com/openai/gym/blob/master/LICENSE.md. The baseline implementation of PPO, SAC, TQC, and QRDQN can be found in Raffin et al. (2019). Moreover, hyperparameters across all experiments used are displayed in Table 9.3, 9.2 and 9.4 .

14 Ablation study for discrete action space on Cartpole-v1

The purpose of this ablation study is to look at the influence of penalization in the discrete action space with QRDQN. In the figures below, we look at the influence of penalizing only during training, which will have the effect of choosing less risky actions during training in order to increase robustness. This curve is denoted *Train penalized.*

Then we look at the influence of penalizing only once the policy has been learned using classic QRDQN without penalization. Only mean-var actions are selected here during testing and not during training. This experience is denoted *Train Penalization.*

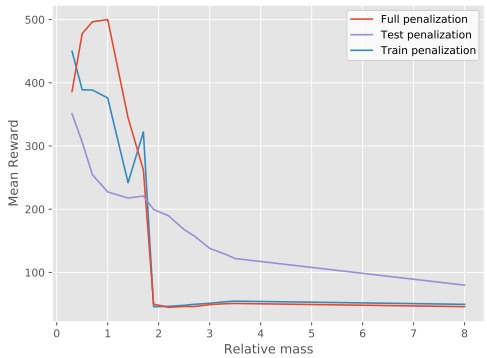
Finally, we compare its variants with our algorithm called *Full penalization.* The results of the ablation are: to achieve optimal performance, both phases are necessary.

When penalties are applied only during training. Good performance is generally obtained close to the length 1 where we train our algorithm. However, the performance is difficult to generalize when the pole length is increased,increased, as we do not penalize during testing.

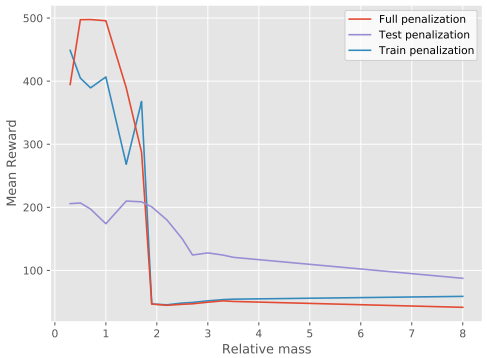
When we penalize only during testing: even if the performances deteriorate, we see that it tends to add robustness because the curves have less tendency to decrease when we increase the length of the pole. The performances are not very high as we play different acts than those taken during the learning.

So both phases are necessary for our algorithm. Penalizing during training allows for safer exploration and penalizing during testing allows for better generalization.

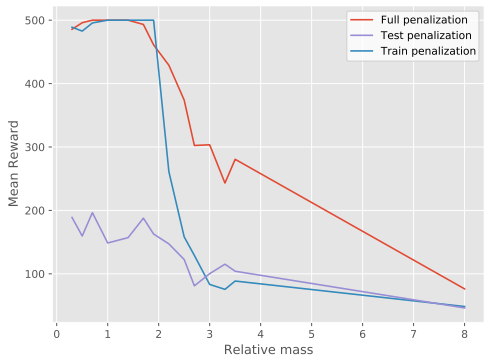
The ablation study for the continuous case is more difficult to do. Indeed, the fact that the penalty occurs only in the gradient descent phase makes it difficult to penalize only in the test phase.



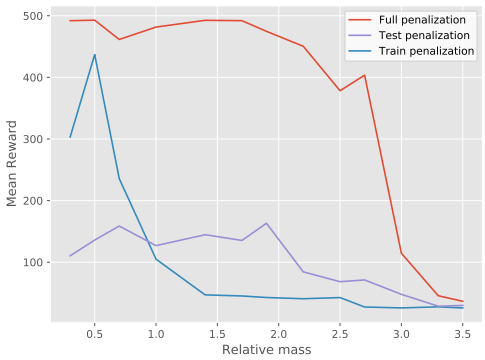
(a) $\alpha = 1$



(b) $\alpha = 3$



(a) $\alpha = 5$



(b) $\alpha = 7$

15 Further Experimental Details

For HalfCheetah-v3 , penalisation is chosen in $[0, 2]$ and not $[0, 5]$ like in Walker-v3 and Hopper-v3. This choice depend on

Table 9.2: Table of best hyperparameter for Cartpole-v1

Hyperparameter	**QRDQN with standard deviation penalisation**	**PPO**
Learning Rate	2.3e-3	3e-4
Optimizer	Adam	Adam
Replay Buffer Size	10e5	N/A
Number of Quantiles	10	N/A
Huber parameter κ	1	N/A
Penalisation α	{0,1,3,5,7 }	N/A
Network Hidden Layers for Policy	N/A	256:256
Network Hidden Layers for Critic	256:256	256:256
Number of samples per Minibatch	64	256
Discount factor γ	0.99	0.99
Target smoothing coefficient β	.0.005	N/A
Non-linearity	ReLu	ReLu
Target update interval	10	N/A
Gradient steps per iteration	1	1
Entropy coefficient	N/A	0
GAE λ	0.95	0.8

Table 9.3: Table of best hyperparameter for Acrobot-v1

Hyperparameter	**QRDQN with standard deviation penalisation**	**PPO**
Learning Rate	6.3e-4	3e-4
Optimizer	Adam	Adam
Replay Buffer Size	50 000	N/A
Number of Quantiles	25	N/A
Huber parameter κ	1	N/A
Penalisation α	$\{0, 0.5, 1, 2, 3\}$	N/A
Network Hidden Layers for Critic	256:256	256:256
Network Hidden Layers for Policy	N/A	256:256
Number of samples per Minibatch	128	64
Discount factor γ	0.99	0.99
Target smoothing coefficient β	.0.005	N/A
Non-linearity	ReLu	ReLu
Target update interval	250	N/A
Gradient steps per iteration	4	1
Entropy coefficient	N/A	0
GAE λ	0.95	0.95

Table 9.4: Table of best hyperparameter for all continuous environments

Hyperparameter	**TQC with standard deviation penalisation**	**SAC**
Learning Rate	linear decay (lr) from 7.3e-4	lr from 7.3e-4
Optimizer	Adam	Adam
Replay Buffer Size	10^6	10^6
Expected Entropy Target	$-\dim\mathcal{A}$	$-\dim\mathcal{A}$
Number of Quantiles	25	N/A
Huber parameter κ	1	N/A
Penalisation α	$\{0, 1, ...5\}$	N/A
Network Hidden Layers for Policy	256:256	256:256
Network Hidden Layers for Critic	512:512:512	256:256
Number of dropped atoms	2	N/A
Number of samples per Minibatch	256	256
Discount factor γ	0.99	0.99
Target smoothing coefficient β	.0.005	0.005
Non-linearity	ReLu	ReLu
Target update interval	1	1
Gradient steps per iteration	1	1

Appendix of Chapter 5

16 Proof

Theorem 16.1.

$$(\mathcal{T}_\alpha^\pi V)(s) = \sum_a \pi(a|s)(r(s,a) + \gamma m_\alpha(P_{sa}, V)). \tag{A.314}$$

this is a contraction:

The expectile satisfies the following properties (Bellini et al. 2014, Bellini and Di Bernardino 2017):

1. Translation invariance: $m_\tau(X + h) = m_\alpha(X) + h$
2. Monotonicity: $X \leq Y$ a.s. $\Rightarrow m_\tau(X) \leq m_\alpha(Y)$
3. Positive homogeneity:
   $\lambda \geq 0 \Rightarrow m_\alpha(\lambda X) = \lambda m_\tau(X)$
4. Superadditivity, for $\alpha \leq \frac{1}{2}, m_\tau(X + Y) \geq m_\alpha(X) + m_\alpha(Y)$.

So,

$$(\mathcal{T}_\alpha^\pi V_1)(s) - (\mathcal{T}_\alpha^\pi V_2)(s) = \sum_a \pi(a|s)(r(s,a) + \gamma m_\alpha(P_{sa}, V_1)) \tag{A.315}$$

$$- \sum_a \pi(a|s)(r(s,a) + \gamma m_\alpha(P_{sa}, V_2)) \tag{A.316}$$

$$= \gamma \sum_a \pi(a|s)(m_\alpha(P_{sa}, V_1) - m_\alpha(P_{sa}, V_2)) \tag{A.317}$$

$$\leq \gamma \sum_a \pi(a|s)(m_\alpha(P_{sa}, V_2 + \|V_2 - V_1\|_\infty) - m_\alpha(P_{sa}, v_2)) \tag{A.318}$$

(by monotonicity) and $V_1 \leq V_2 + \|V_2 - V_1\|_\infty$)

$$= \gamma \sum_a \pi(a|s)(m_\alpha(P_{sa}, V_2) + \|V_2 - V_1\|_\infty - m_\alpha(P_{sa}, V_2)) \text{ (by translation invariance)} \tag{A.319}$$

$$= \gamma \|V_2 - V_1\|_\infty. \tag{A.320}$$

In the same manner, $\mathcal{T}_\alpha^*$ is also a contraction, as the only line of this proof that differs is replacing the expectation by a $\max_a$. As maximum operator 1-Lipschitz, (ie) $\max_a f(a) - \max g(a) \leq \max f(a) - g(a)$, we obtain γ- contraction results also for the optimal Bellman operator T_α^*.

Similar ideas exist in Zhang et al. (2023), which show similar properties for risk-sensitive MDPs defined through a convex risk measure, even though they do not consider explicitly the expectile which is a convex risk measure for $\alpha < 1/2$.

Theorem 16.2. *The (optimal) Expectile value function is equal to the (optimal) robust value function*

$$V_\alpha^*(s) = V_\mathcal{E}^\pi := \max_\pi \min_{Q\in\mathcal{E}} V^{\pi,Q} \tag{A.321}$$

$$V_\alpha^\pi(s) = V_\mathcal{E}^\pi := \min_{Q\in\mathcal{E}} V^{\pi,Q} \tag{A.322}$$

where $\mathcal{E}$ is defined in section 5.2.3 or below.

Proof. This theorem is just an adaptation of Theorem 2 in Zhang et al. (2023) where we use expectile risk measure $m_\alpha(X)$ which implicitly defined the uncertainty set for robust $\mathcal{E}$ such that :

$$m_\alpha(X) = \min_{Q\in\mathcal{E}} \mathbb{E}_Q[X];$$

$$\mathcal{E} = \left\{ Q \in \mathcal{P} \mid \ \exists \eta > 0, \sqrt{\frac{\alpha}{1-\alpha}}\eta \leq \frac{dQ}{dP} \leq \sqrt{\frac{(1-\alpha)}{\alpha}}\eta \right\}$$

where $\mathcal{P}$ is the set of P-absolutely continuous probability measures. In Theorem Zhang et al. (2023), they link Risk sensitive MDPs (in our case expectile formulation) with Regularised Robust MDPs. In our case, we can rewrite the classical RMDPs to Regularised-Robust MDPs such that:

$$V_\mathcal{E}^* = \max_\pi \min_{Q\in\mathcal{E}} V^{\pi,Q} = \max_\pi \min_{Q\in\mathcal{E}} \mathbb{E}\Big[\sum_t \gamma^t r(s_t,a_t)\Big]$$
$$= \max_\pi \min_{Q\in\mathcal{P}} \mathbb{E}\Big[\sum_t \gamma^t (r(s_t,a_t) + \gamma D(P_{t;s_t,a_t}, Q_{t;s_t,a_t})\Big]$$

with D a penalty function that can be chosen as KL divergence for example and $P_{t;s_t,a_t}$ the transition kernel at time t with current state action (s_t,a_t).For the expectile risk measure, the corresponding D is simply:

$$D(P,Q) = \begin{cases} 0 & \text{if } \eta\sqrt{\frac{\alpha}{1-\alpha}} \leq P(s)/Q(s) \leq \sqrt{\frac{(1-\alpha)}{\alpha}}\eta, \forall s \in S \\ +\infty & \text{otherwise.} \end{cases}$$

where η is defined in 5.2.3. Using Theorem 2 of Zhang et al. (2023), we have directly that :

$$V_\alpha^*(s) = V_\mathcal{E}^\pi := \max_\pi \min_{Q\in\mathcal{E}} V^{\pi,Q} \tag{A.323}$$

$$V_\alpha^\pi(s) = V_\mathcal{E}^\pi := \min_{Q\in\mathcal{E}} V^{\pi,Q}. \tag{A.324}$$

□

17 `AutoExpectRL` algorithm description

In the section, we gives implementation details of our algorithm `AutoExpectRL`. First, we choose a neural network that has 4 heads for the critic, one per value of α, leading to 4 estimates of the

Algorithm 12: AutoExpectRL

Initialize critic networks Q_{ϕ_d} and actor π_θ $\forall d \in [1,4]$
Initialize target networks for all networks, i.e. $\forall d \in [1,4]$ $\phi'_d \leftarrow \phi_d$, $\theta'_d \leftarrow \theta_d$
Initialize replay buffer and bandit probabilities $\mathcal{B} \leftarrow \emptyset$, $\mathbf{p}_1^\alpha \leftarrow \mathcal{U}([0,1]^D)$
for episode in $m = 1, 2, \ldots$ **do**
  Initialize episode reward $R_m \leftarrow 0$
  Sample expectile $\alpha_m \sim \mathbf{p}_m^\alpha$
  for time step $t = 1, 2, \ldots, T$ **do**
    Select noisy action $a_t = \pi_{\theta_d}(s_t) + \epsilon$, $\epsilon \sim \mathcal{N}(0, s^2)$, obtain r_{t+1}, s_{t+1} where d is the index in the bandit problem of chosen expectile α_m
    Add to total reward $R_m \leftarrow R_m + r_{t+1}$
    Store transition $\mathcal{B} \leftarrow \mathcal{B} \cup \{(s_t, a_t, r_{t+1}, s_{t+1})\}$
    Sample N transitions $B = (s, a, r, s')_{n=1}^N \sim \mathcal{B}$.
    `Update Critics`(B, θ', ϕ') according to (A.326).
    if t mod b **then**
      `UpdateActor`(T, θ, ϕ) according to (A.327).
      Update ϕ'_d: $\phi'_d \leftarrow \tau\phi_d + (1-\tau)\phi'_d$, $d \in \{1, 4\}$
      Update θ': $\theta'_d \leftarrow \tau\theta_d + (1-\tau)\theta'_d$
  end for
  Update bandit $\mathbf{p}^\alpha$ weights using : $w_{m+1}(d) = w_m(d) + \eta \frac{R_m - R_{m-1}}{\mathbf{p}_m^\alpha(d)}$
end for

pessimist Q-function, $Q_{\phi_d}(s,a)$, $\forall d \in [1,4]$. Even if some parameters are shared in the body of the network, we denote parameters of the critic as $\phi = \{\phi_1, \phi_2, \phi_3, \phi_4\}$. A similar network is used for actor neural network, with four heads, one per policy $\pi_{\theta_d}, \forall d \in [1,4]$. with $\theta = \{\theta_1, \theta_2, \theta_3, \theta_4\}$.

Given 4 target $y_d(r, s') = r + \gamma Q_{\phi_{d,\text{targ}}}(s', \pi_{\theta_d}(s'))$ with reward r, policy π_{θ_d}, we propose to minimize the AutoExpectRL critic loss

$$L_{auto}(\phi, \mathcal{D}) = \mathop{\mathbb{E}}_{(s,a,r,s') \sim \mathcal{D}} \left[\sum_{d=1}^{4} L_2^{\alpha_d} \left(Q_{\phi_d}(s,a) - y_d(r,s') \right) \right]. \tag{A.325}$$

which as associated `UpdateCritics`(B, θ, ϕ) function which is a gradient ascent using :

$$\Delta_\phi \propto \nabla_\phi \frac{1}{|B|} \sum_{(s,a,r,s') \in B} \sum_{d=1}^{4} L_2^{\alpha_d} \left(Q_{\phi_d}(s,a) - y_d(r,s') \right). \tag{A.326}$$

The actor of our algorithm AutoExpectRL is updated according to the gradient of the sum of the actor's head losses or `UpdateActor`(T, θ, ϕ):

$$\Delta\theta \propto \nabla_\theta \frac{1}{|B|} \sum_{s \in B} \sum_{k=1}^{4} Q_{\phi_k}(s, \pi_{\theta_k}(s)). \tag{A.327}$$

The dimension of our neural network is related to the dimension of the classical network of TD3. First, we choose a common body of share weights for our neural network of hidden dimension $[400, 300]$. Then our network is composed of 4 heads, each with final matrix weights of dimensions 300×1 where 1 represents the value of one pessimist Q-function Q_k. The dimension

of the actor-network hidden layers is similar to the critic network for share weights, but the non-shared weights between the last hidden layer and the 4 policies have dimension $300 \times |A|$. Finally, the sampling of new trajectories is done using the actor head with the chosen current α proposed by the bandit algorithm using π_{θ_d} with d the index of the chosen expectile.

The algorithm can be summarised as in Algorithm 12. The blue parts are parts that differ from the traditional TD3 algorithm, as they are related to the bandit mechanism or `ExpectRL` losses. Note that the parameter b, the delay between the update of the critic and the actor, is usually chosen as 2 in TD3 algorithm. Finally, in the update of the bandit, an extra parameter, the learning rate η of the gradient ascent must be chosen. This parameter influences how fast the bandit converges to an arm, and in our case is chosen as 0.2 like in Moskovitz et al. (2021) which uses bandit to fine-tune parameters in Rl algorithm. for all environments. Finally, in the testing phase of the benchmark, the best arm is chosen to maximize the reward.

18 Hyperparameters

Hyperparameter	Value
Learning rate actor	$3e-4$
Learning rate critic	$3e-3$
Batch size	100
Memory size	3e5
Gamma	0.99
Polyak update τ	0.995
Number of steps before training	7e4
Train frequency and gradient step	100
Network Hidden Layers (Critic)	[400, 300] like original implementation of TD3
Network Hidden Layers (Actor)	[400, 300] like original implementation of TD3

Table 9.5: Hyperparameters

All experiments were run on an internal cluster containing a mixture of GPU Nvidia Tesla V100 SXM2 32 Go. Each run was performed on a single GPU and lasted between 1 and 8 hours, depending on the task and GPU model. Our baseline implementations for TD3 is Raffin et al. (2021) where we use the same base hyperparameters across all experiments, displayed in Table ??.

19 AutoExpecRL vs other expectiles on Robust benchmark for mean on Table 5.1

This section illustrates the fact that ExpecRL method outperforms on robust benchmark TD3 algorithm. Without any hyperparameter tuning, AutoExpecRL achieves a similar performance to ExpecRL with the best expectile, finding the best arms in the bandit problem. In Ant and Hopper environments, the best expectile is frequently very low, typically $\alpha = 0.2$ our 0.3 where this is less the case for HalfCheetah and Humanoid where the best expectile is bigger. Finally, we can remark that smaller expectiles give better performance in terms of min performance while for average metric, higher expectiles are chosen, which is also verified in Table 9.6 for DR benchmark.

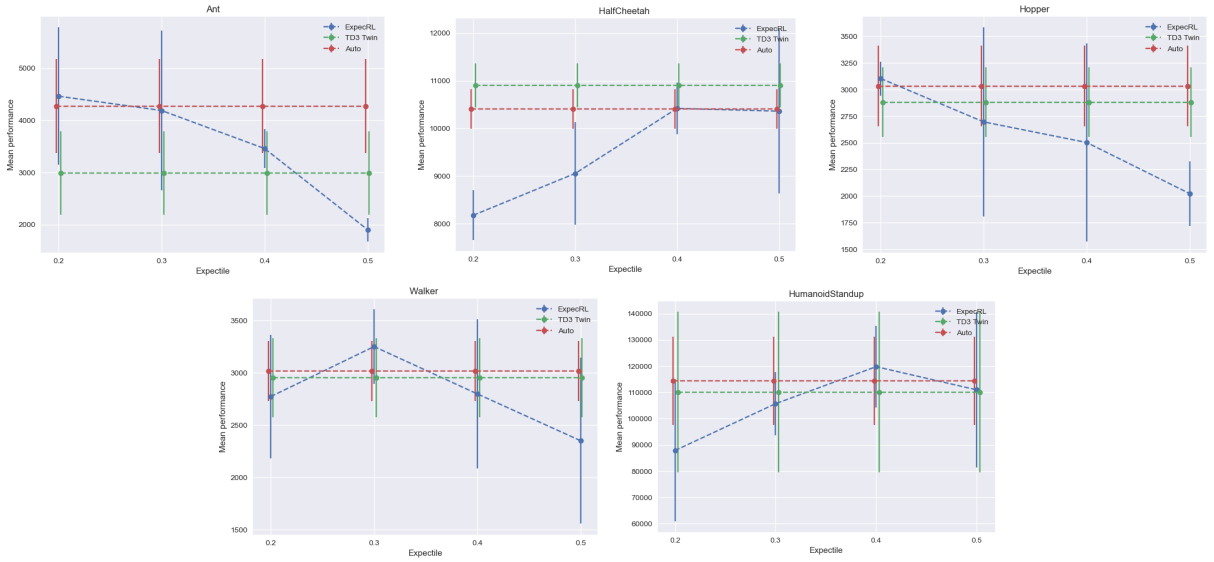


Figure A19.5: Mean performance as a function of the expectile, non-robust case (corresponding to Table 5.1).

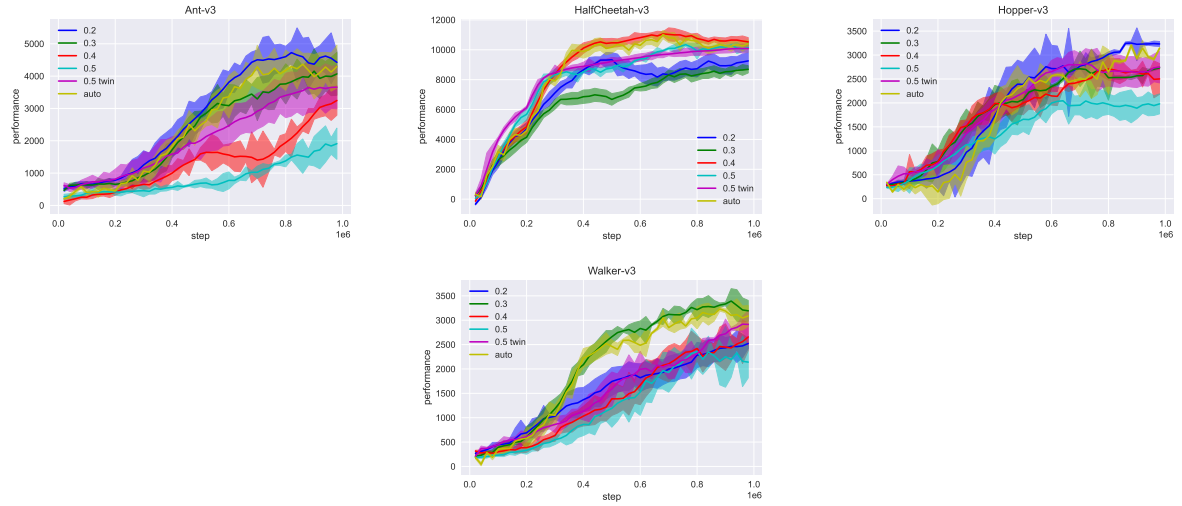


Figure A19.6: Learning curves non-robust case (corresponding to Table 5.1).

20 Worst case performance for `AutoExpecRL` and `ExpecRL` (only nominal samples) or Table 5.2.

20.1 For 1D uncertainty greed benchmark

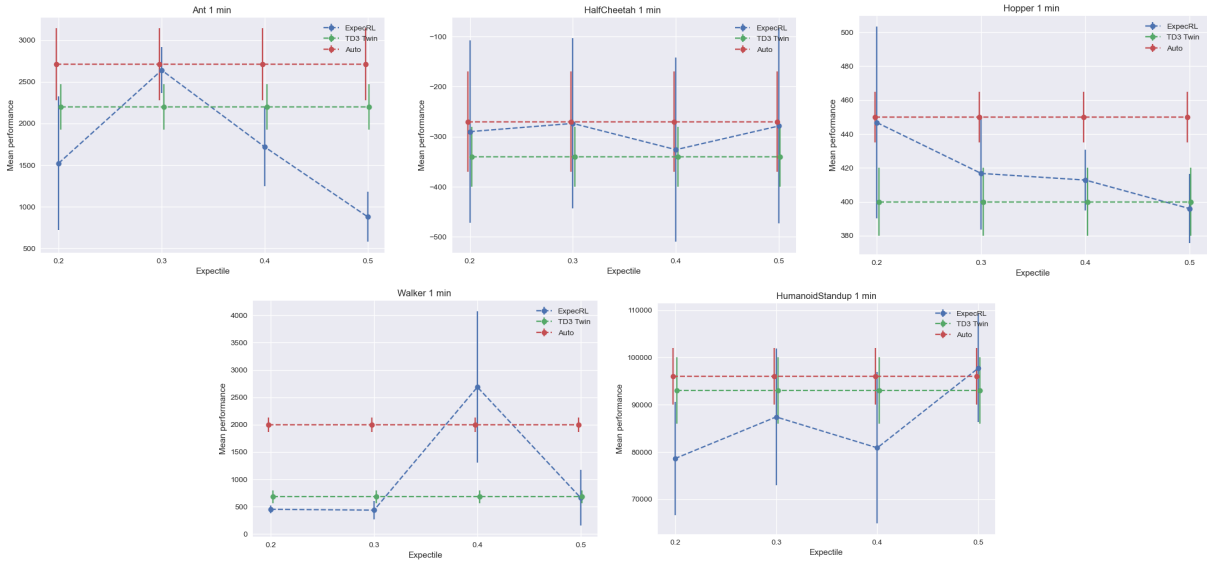


Figure A20.7: Min performance as a function of the expectile, robust case (corresponding to Table 5.2).

20.2 For 2D uncertainty greed benchmark

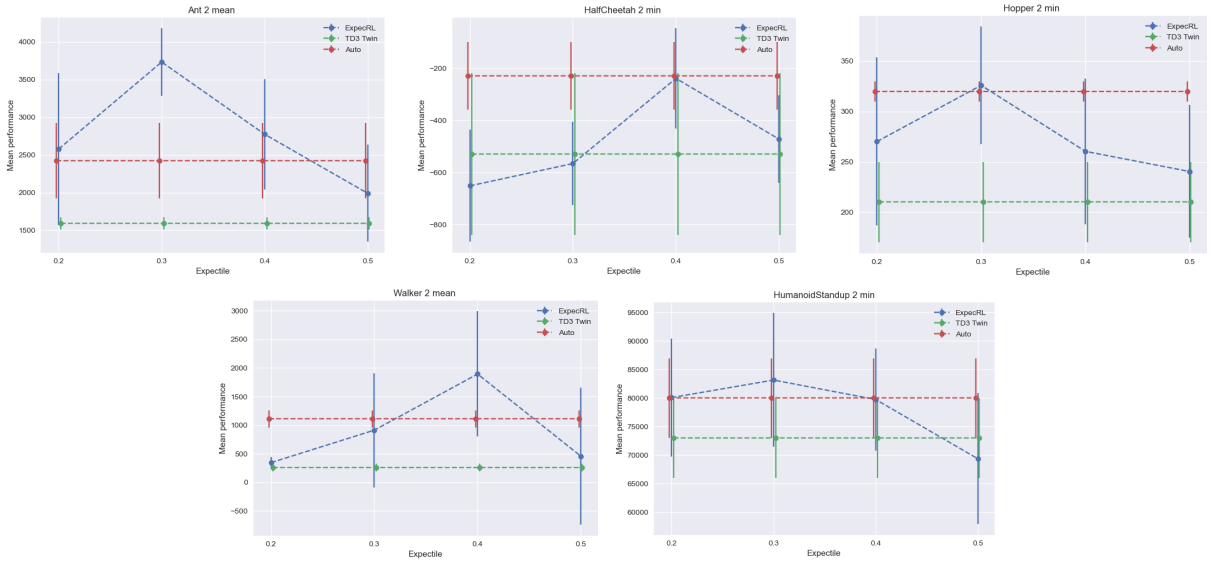


Figure A20.8: Min performance as a function of the expectile, robust case (corresponding to Table 5.2).

20.3 For 3D uncertainty greed benchmark

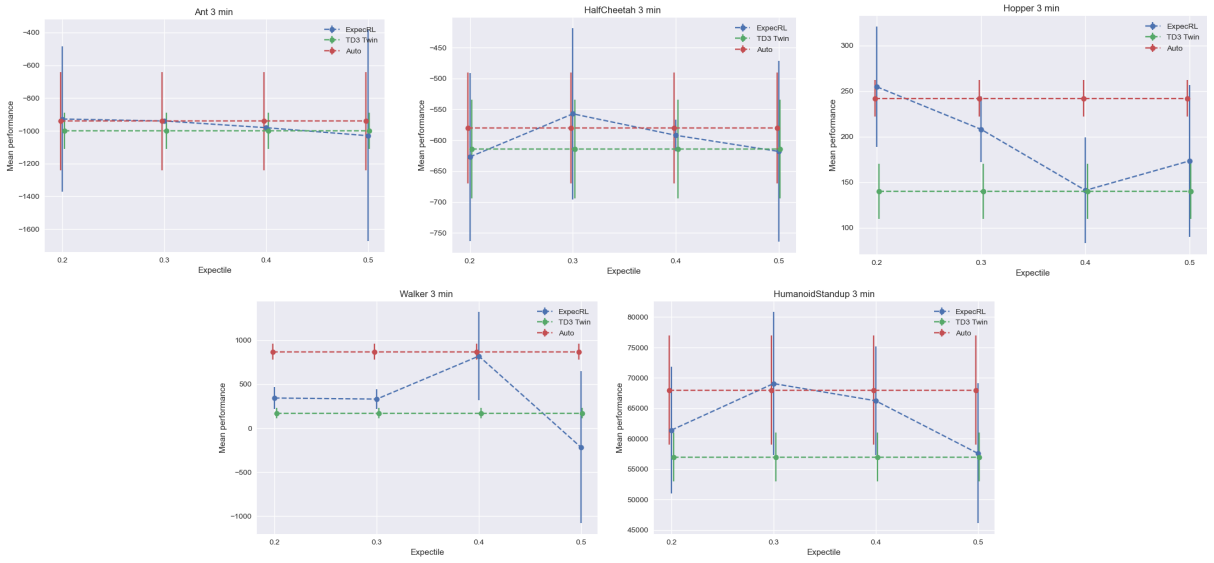


Figure A20.9: Min performance as a function of the expectile, robust case (corresponding to Table 5.2).

21 Average performance for AutoExpecRL and ExpecRL(only nominal samples) or Table 5.2.

21.1 For 1D uncertainty greed benchmark

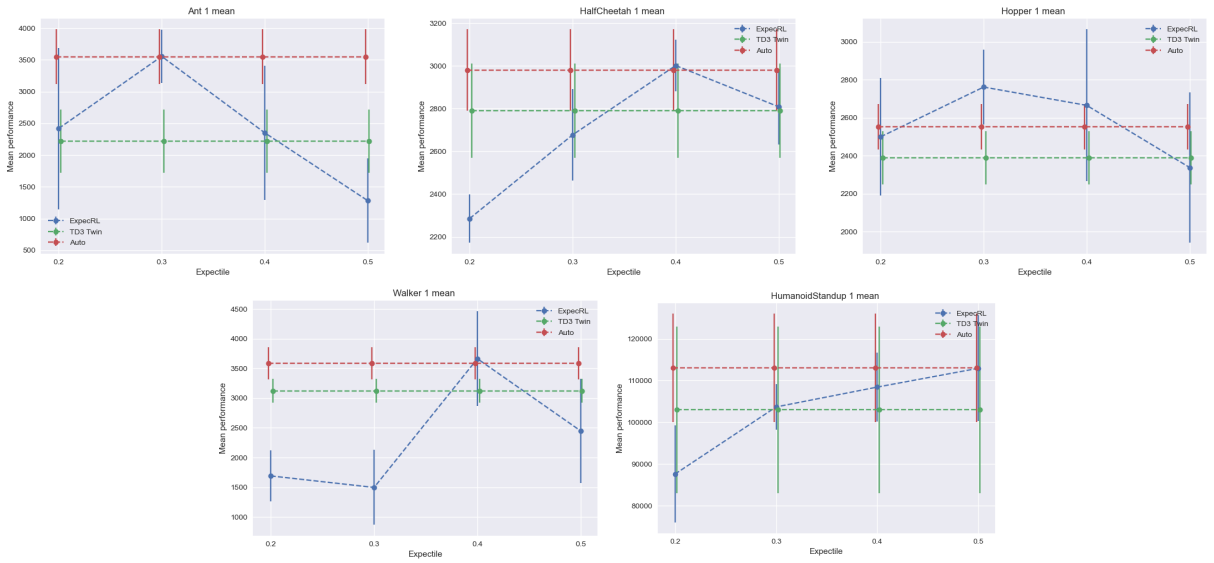


Figure A21.10: Min performance as a function of the expectile, robust case (corresponding to Table 5.2).

21.2 For 2D uncertainty greed benchmark

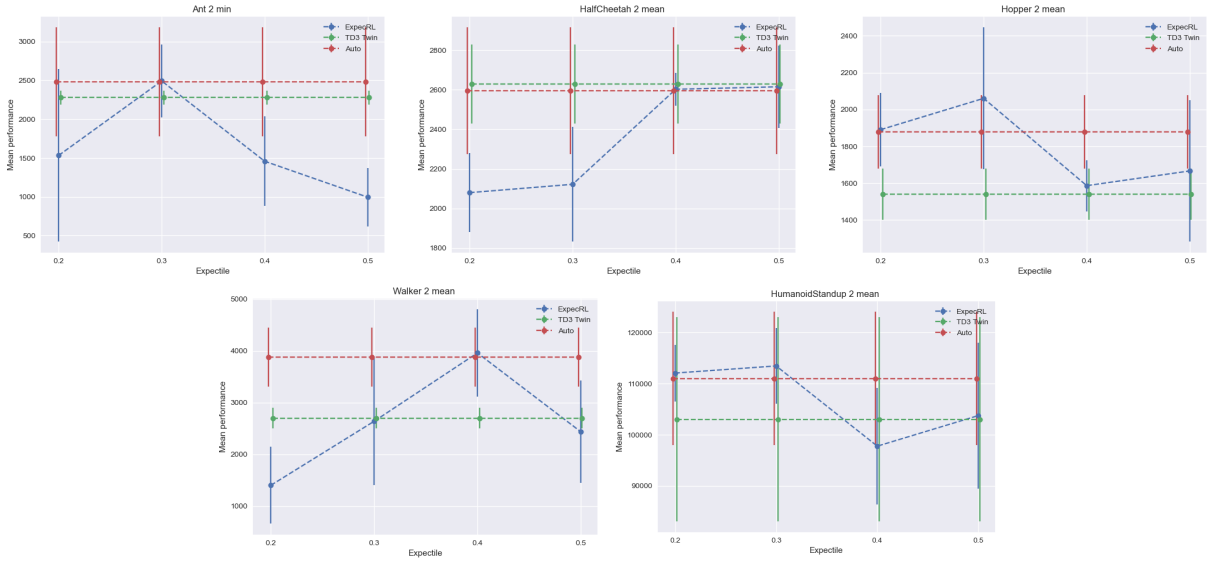


Figure A21.11: Min performance as a function of the expectile, robust case (corresponding to Table 5.2).

21.3 For 3D uncertainty greed benchmark

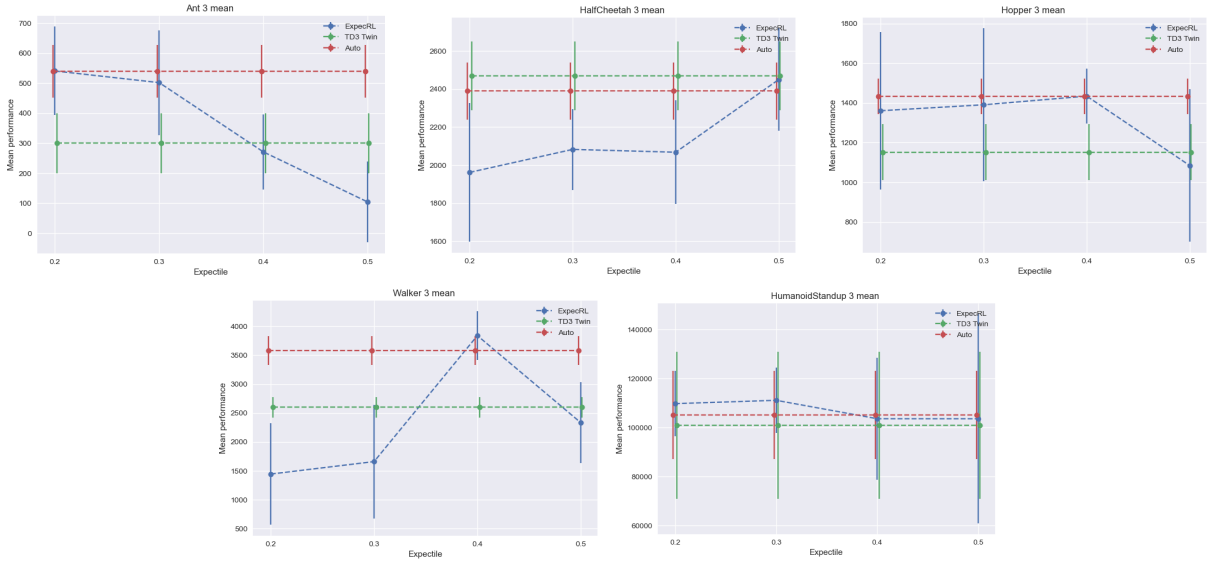


Figure A21.12: Min performance as a function of the expectile, robust case (corresponding to Table 5.2).

22 Additional details for expectiles on Robust benchmark for worst-case and mean on Table 5.3

23 Uncertainty sets used for Robust benchmark

Env	Min	Mean
$Ant1$	3	3
$Ant2$	2	3
$Ant3$	2	3
$HalfCheetah1$	3	3
$HalfCheetah2$	3	3
$HalfCheetah3$	3	3
$Hopper1$	3	4
$Hopper2$	3	4
$Hopper3$	3	3
$Walker1$	3	4
$Walker2$	4	4
$Walker3$	3	3
$HumanoidStandup1$	3	3
$HumanoidStandup2$	2	3
$HumanoidStandup3$	2	3

Table 9.6: Best Expectile in DR for `ExpectRL`

Table 9.7: Uncertainty sets used for Robust benchmark

Environment	Uncertainty Set Ω	Reference Parameter	Uncertainty Parameter Name
Baseline MuJoCo Environment: Ant			
Ant 1	[0.1, 3.0]	0.33	torso mass
Ant 2	[0.1, 3.0] × [0.01, 3.0]	(0.33, 0.04)	torso mass × front left leg mass
Ant 3	[0.1, 3.0] × [0.01, 3.0] × [0.01, 3.0]	(0.33, 0.04, 0.06)	torso mass × front left leg mass × front right leg mass
Baseline MuJoCo Environment: HalfCheetah			
HalfCheetah 1	[0.1, 4.0]	0.4	world friction
HalfCheetah 2	[0.1, 4.0] × [0.1, 7.0]	(0.4, 6.36)	world friction × torso mass
HalfCheetah 3	[0.1, 4.0] × [0.1, 7.0] × [0.1, 3.0]	(0.4, 6.36, 1.53)	world friction × torso mass × back thigh mass
Baseline MuJoCo Environment: Hopper			
Hopper 1	[0.1, 3.0]	1.00	world friction
Hopper 2	[0.1, 3.0] × [0.1, 3.0]	(1.00, 3.53)	world friction × torso mass
Hopper 3	[0.1, 3.0] × [0.1, 3.0] × [0.1, 4.0]	(1.00, 3.53, 3.93)	world friction × torso mass × thigh mass
Baseline MuJoCo Environment: HumanoidStandup			
HumanoidStandup 1	[0.1, 16.0]	8.32	torso mass
HumanoidStandup 2	[0.1, 16.0] × [0.1, 8.0]	(8.32, 1.77)	torso mass × right foot mass
HumanoidStandup 3	[0.1, 16.0] × [0.1, 5.0] × [0.1, 8.0]	(8.32, 1.77, 4.53)	torso mass × right foot mass × left thigh mass
Baseline MuJoCo Environment: Walker			
Walker 1	[0.1, 4.0]	0.7	world friction
Walker 2	[0.1, 4.0] × [0.1, 5.0]	(0.7, 3.53)	world friction × torso mass
Walker 3	[0.1, 4.0] × [0.1, 5.0] × [0.1, 6.0]	(0.7, 3.53, 3.93)	world friction × torso mass × thigh mass

Appendix of Chapter 6

24 Appendix

The Appendix is structured as follow :

- In Appendix 25, proof for fix point of `Oracle-TC` algorithm for can be found.
- In Appendix 26, proof for algorithm Vanilla `TC` and `Stacked-TC` can found about robust objective.
- In Appendix 29, the adversary training was sanity-checked within the time-constrained evaluation.
- In Appendix 28, all implementation details are provided.
- In Appendix 31, all raw results are presented.
- In Appendix 32, the computer resources and training wall clock time are detailed.
- In Appendix 33, the broader impact and limitations are discussed.

25 Proof of Theorem 6.2.1

Proof. The Proof is similar to Iyengar (2005), using the fact that $P_{\psi+b}$ belongs to the simplex, we get contraction of the operator and convergence to a fix point V_B^*. Not that to converge to the fix point, there is no need of rectangularity. □

Recall the recursion

$$V_{n+1}(s,\psi) = \max_{\pi(s,\psi)\in\Delta_A} \min_{b\in B} T_b^\pi V_n(s,\psi) := \max_{\pi(s,\psi)\in\Delta_A} \min_{b\in B} \mathbb{E}_{a\sim\pi(s)}[r(s,a) + \gamma \mathbb{E}_{s'\sim P_{\psi+b}} V_n(s',\psi')] \tag{A.328}$$

First we prove that the TC Robust Operator T_B^π is a contraction. Let $V_1, V_2 \in \mathbb{R}^n$. Fix $s \in S$, and assume that $T_B^\pi V_1(s,\psi) \geq T_B^\pi V_2(s,\psi)$. Then fix $\epsilon > 0$ and pick π s.t given $s \in S$,

$$\inf_{b\in B} \mathbb{E}_{p_{\psi+b}} \left[r\left(s,\pi(s)\right) + \gamma V_1\left(s',\psi'\right)\right] \geq T_B^\pi V_1(s,\psi') - \epsilon. \tag{A.329}$$

First we pick a probability measure P' such that $P' = P_{\psi+b}, b \in B$, such that

$$\mathbb{E}_{P'}\left[r\left(s,\pi(s)\right) + \gamma V_2\left(s',\psi'\right)\right] \leq \inf_{b\in B} \mathbb{E}_{P'}\left[r\left(s,\pi(s)\right) + \gamma V_2\left(s',\psi'\right)\right] + \epsilon. \tag{A.330}$$

Then it lead to

$$0 \leq \mathcal{T}_B^\pi V_1(s,\psi) - \mathcal{T}_B^\pi V_2(s,\psi) \leq \left(\inf_{P\in B} \mathbb{E}_P \left[r\left(s,\pi(s)\right) + \gamma V_1\left(s',\psi'\right)\right] + \epsilon \right) \tag{A.331}$$

$$- \left(\inf_{P\in B} \mathbb{E}_P \left[r\left(s,\pi(s)\right) + \gamma V_2\left(s',\psi'\right)\right] \right) \tag{A.332}$$

$$\leq \left(\mathbb{E}_{P'} \left[r\left(s,\pi(s)\right) + \gamma V_1\left(s',\psi'\right)\right] + \epsilon \right) - \tag{A.333}$$

$$\left(\mathbb{E}_{P'} \left[r\left(s,\pi(s)\right) + \gamma V_2\left(s',\psi'\right)\right] - \epsilon \right), \tag{A.334}$$

$$= \gamma \mathbb{E}_{P'}[V_1 - V_2] + 2\epsilon, \tag{A.335}$$

$$\leq \gamma \mathbb{E}_{P'}|V_1 - V_2| + 2\epsilon \tag{A.336}$$

$$\leq \gamma \|V_1 - V_2\|_\infty + 2\epsilon. \tag{A.337}$$

where last inequality is Holder's inequality between L_1 and L_∞ norms, use probability measure in the simplex such as $\|P'\|_1 = 1$. Doing the same thing but in the case where $\mathcal{T}_B^\pi V_1(s) \leq \mathcal{T}_B^\pi V_2(s)$, it holds

$$\forall s \in S, |\mathcal{T}_B^\pi V_1(s) - \mathcal{T}_B^\pi V_2(s)| \leq \gamma \|V_1 - V_2\|_\infty + 2\epsilon \ , \tag{A.338}$$

i.e. $\|\mathcal{T}_B^\pi V_1 - \mathcal{T}_B^\pi V_2\|_\infty \leq \gamma \|V_1 - V_2\|_\infty + 2\epsilon$. As we can choose ϵ arbitrary small, this establishes that the TC Bellman operator is a γ-contraction. Since T_B^π is a contraction operator on a Banach space, the Banach fixed point theorem implies that the operator equation $\mathcal{T}_B^\pi V = V$ has a unique solution $V = V_B^\pi$. A similar proof can be done for optimal operator $\mathcal{T}_B^*$. The only difference is the maximum operator which is $1-$Lipschitz. So $\mathcal{T}_B^*$ is also a contraction. Then, once proved that operators are $\gamma-$ contraction, following (Iyengar 2005) (Th. 5), we have that the fixed point of this recursion is exactly :

$$V_B^\pi(s,\psi) := \min_{\substack{(b_t)_{t\in\mathbb{N}},\\ b_t\in B}} \mathbb{E}\Big[\sum \gamma^t r_t | \psi_{-1} = \psi, s_0 = s, b_t \in B, \psi_t = \psi_{t-1} + b_t, a \sim \pi, s_t \sim P_{\psi_t} \Big], \tag{A.339}$$

$$V_B^*(s,\psi) = \max_{\pi(s,\psi)\in\Delta_A} V_B^\pi(s,\psi) \ . \tag{A.340}$$

for (optimal) TC Bellman Operator.

26 Guaranties for non-stationary Robust MDPS

Recall that we represent a non-stationary robust MDPs (NS-RMDP) as a stochastic sequence, $\{\mathcal{M} = \{M_i\}_{t=t_0}^\infty$, of stationary MDPs $M_t \in \mathcal{M}$, where $\mathcal{M}$ is the set of all stationary MDPs. Each M_t is a tuple, $(\mathcal{S}, \mathcal{A}, P_t, r_t, \gamma, \rho^0)$, where $\mathcal{S}$ The set of possible states is denoted by $\mathcal{S}$, the set of actions by $\mathcal{A}$, the discounting factor by γ, the start-state distribution by ρ^0, and the reward distribution by r_t. The reward distribution, denoted by $r_t : \mathcal{S} \times \mathcal{A} \to \Delta(\mathbb{R})$, is the probability distribution of rewards. The transition function, represented by $P_t : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, is the probability distribution of transitions between states. The symbol Δ denotes the simplex. For all $M_t \in \mathcal{M}$, we assume that the state space, action space, discount factor, and initial distribution remain fixed. A policy is represented as a function $\pi : \mathcal{S} \to \Delta(\mathcal{A})$. In general, we will use subscripts t to denote the time evolution during an episode and superscripts k to denote the time step assuming reward or kernel t which is stationary, assuming that the reward function is not changing as it is at time step t stationary. That r_t^k is the random variables corresponding to the state, action, and reward at time step t for stationary, but iterating with index k.

Definition 26.1 (Lipschitz of sequence of MDPs)**.** *We denote the sequence of kernel and reward function* $\mathcal{P} = \{P_t\}_{t=t_0}^{\infty}$ *and* $\mathcal{R} = \{r_t\}_{t=t_0}^{\infty}$*. We define a sequence of MDP is* $L = (L_r, L_P)$*-Lipchitz if* $m = \{m_t\}_{t=t_0}^{\infty} \in \mathcal{M}^L$ *with*

$$\mathcal{M}_t^L = \Big\{ \{M_t\}_{t'=t_0}^{t} ; \exists (L_r, L_P) \in \mathbb{R}_+^2 \forall t \in \mathbb{N}, \forall s \in \mathcal{S}, \forall a \in \mathcal{A}, \|P_{t'}(\cdot \mid s,a) - P_{t'+1}(\cdot \mid s,a)\|_1 \leq L_P \\ ; |r'_t(s,a) - r_{t'+1}(s,a)| \leq L_r \Big\}$$

Assuming that for a time steps the reward function is stationary, we can compute the average return as:

Definition 26.2. *Non-robust objective function, assuming that* $G(\pi, M_t) = \sum_{k\geq 0} \gamma^k r_t^k$*, the return is we assume stationary with reward function* r_t

$$J(\pi, t) = \mathbb{E}[G(\pi, M_t)] = (1-\gamma)^{-1} \sum_{s\in\mathcal{S}} d^{\pi}(s, M_t) \sum_{a\in\mathcal{A}} \pi(a \mid s) r_t(s,a). \tag{A.341}$$

with d^{π} *the state occupancy measure defined in* (A.342)*.*

Definition 26.3 (Robust (optimal) Return of NS-RMDPs)**.** *Let a return of* π *for any* $m_t \in M_t$ *be* $G(\pi, M_t) := \sum_{k=0}^{\infty} \gamma^k r_t^k$ *with kernel transition* P_t *following* π*, with* $\forall k, t, r_t^k \in [0,1]$*, and the Robust non-stationary expected return with variation of kernel*

Let the robust performance of π *for episode* t *be*

$$J^R(\pi, t) := \min_{m=\{m'_t\}_{t'=t_0}^{t} \in \mathcal{M}_t^L} \mathbb{E}\left[G(\pi, m)\right]$$

27 Proof Theom 6.6.1

$$\forall t \in \mathbb{N}^+, \forall t_0 \in \mathbb{N}^+, \quad |J^R(\pi, t_0) - J^R(\pi, t_0 + t)| \leq L' t.$$

with $L' := \left(\frac{\gamma}{(1-\gamma)^2} L_P + \frac{1}{1-\gamma} L_r\right)$

Proof of Theorem 6.6.1. First, this difference can be upper bounded in the non robust case as:

By definition, we can rewrite non-robust objective function and occupancy measure as.

$$d^{\pi}(s, M_t) = (1-\gamma) \sum_{k=0}^{\infty} \gamma^k \Pr(S_t = s \mid \pi, M_t), \tag{A.342}$$

$$J(\pi, M_t) = (1-\gamma)^{-1} \sum_{s\in\mathcal{S}} d^{\pi}(s, M_t) \sum_{a\in\mathcal{A}} \pi(a \mid s) r_t(s,a). \tag{A.343}$$

First, we can decompose the problem into sub-problems such that

$$\forall t \in \mathbb{N}^+, \forall t_0 \in \mathbb{N}^+, \quad |J(\pi, t_0) - J(\pi, t_0 + t)| \leq |\sum_{t'=t_0}^{t_0+t-1} |J(\pi, M_{t'}) - J(\pi, M_{t'+1})| \tag{A.344}$$

using triangular inequality. Looking at differences between two time steps:

$$
\begin{aligned}
&(1-\gamma)|J(\pi, M_t) - J(\pi, M_{t+1})| \\
&= \left| \sum_{s\in\mathcal{S}} d^\pi(s, M_t) \sum_{a\in\mathcal{A}} \pi(a \mid s) r_t(s,a) - \sum_{s\in\mathcal{S}} d^\pi(s, M_{t+1}) \sum_{a\in\mathcal{A}} \pi(a \mid s) r_{t+1}(s,a) \right| \\
&= \left| \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \pi(a \mid s) \left(d^\pi(s, M_t)\, r_t(s,a) - d^\pi(s, M_{t+1})\, r_{t+1}(s,a) \right) \right| \\
&= \left| \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \pi(a \mid s) \left(d^\pi(s, M_t) \left(r_{t+1}(s,a) + (r_t(s,a) - r_{t+1}(s,a)) \right) - d^\pi(s, M_{t+1})\, r_{t+1}(s,a) \right) \right| \\
&= \Big| \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \pi(a \mid s) \left(d^\pi(s, M_t) - d^\pi(s, M_{t+1}) \right) r_{t+1}(s,a) \\
&\quad + \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \pi(a \mid s) d^\pi(s, M_t) \left(r_t(s,a) - r_{t+1}(s,a) \right) \Big| \\
&\overset{(a)}{\leq} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \pi(a \mid s) \left| d^\pi(s, M_t) - d^\pi(s, M_{t+1}) \right| |r_{t+1}(s,a)| \\
&\quad + \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \pi(a \mid s) d^\pi(s, M_t) \left| r_t(s,a) - r_{t+1}(s,a) \right| \\
&\overset{(b)}{\leq} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \pi(a \mid s) \left| d^\pi(s, M_t) - d^\pi(s, M_{t+1}) \right| + L_R \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \pi(a \mid s) d^\pi(s, M_t) \\
&= \sum_{s\in\mathcal{S}} \left| d^\pi(s, M_t) - d^\pi(s, M_{t+1}) \right| + L_r
\end{aligned}
$$

where (a) is triangular inequality, (b) is definition of of supremum of reward in the assumptions and reward bounded by 1. Then, let $P_t^\pi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ be the transition matrix (s' in rows and s in columns) resulting due to π and P_t, i.e., $\forall t, P_t^\pi(s', s) := \Pr(S_{t+1} = s' \mid S_t = s, \pi, M_t)$, and let $d^\pi(\cdot, M_t) \in \mathbb{R}^{|\mathcal{S}|}$ denote the vector of probabilities for each state, then Finally we can easily bound the difference of occupation measure as :

$$
\begin{aligned}
&\sum_{s\in\mathcal{S}} \left| d^\pi(s, M_t) - d^\pi(s, M_{t+1}) \right| && \text{(A.345)} \\
&\overset{(d)}{\leq} \gamma(1-\gamma)^{-1} \sum_{s'\in\mathcal{S}} \left| \sum_{s\in\mathcal{S}} \left(P_t^\pi(s', s) - P_{t+1}^\pi(s', s) \right) d^\pi(s, M_t) \right| && \text{(A.346)} \\
&\leq \gamma(1-\gamma)^{-1} \sum_{s'\in\mathcal{S}} \sum_{s\in\mathcal{S}} \left| P_t^\pi(s', s) - P_{t+1}^\pi(s', s) \right| d^\pi(s, M_t) && \text{(A.347)} \\
&= \gamma(1-\gamma)^{-1} \sum_{s'\in\mathcal{S}} \sum_{s\in\mathcal{S}} \left| \sum_{a\in\mathcal{A}} \pi(a \mid s) \left(\Pr(s' \mid s, a, M_t) - \Pr(s' \mid s, a, M_{t+1}) \right) \right| d^\pi(s, M_t) && \text{(A.348)} \\
&\leq \gamma(1-\gamma)^{-1} \sum_{s'\in\mathcal{S}} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \pi(a \mid s) \left| \Pr(s' \mid s, a, M_t) - \Pr(s' \mid s, a, M_{t+1}) \right| d^\pi(s, M_t) && \text{(A.349)} \\
&= \gamma(1-\gamma)^{-1} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \pi(a \mid s) d^\pi(s, M_t) \sum_{s'\in\mathcal{S}} \left| \Pr(s' \mid s, a, M_t) - \Pr(s' \mid s, a, M_{t+1}) \right| && \text{(A.350)} \\
&\leq \gamma(1-\gamma)^{-1} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \pi(a \mid s) d^\pi(s, M_t)\, L_P && \text{(A.351)} \\
&= \frac{\gamma L_P}{(1-\gamma)}, && \text{(A.352)}
\end{aligned}
$$

which gives regrouping all terms:

$$|J(\pi, M_t) - J(\pi, M_{t+1})| \leq \frac{L_r}{1-\gamma} + \frac{\gamma L_P}{(1-\gamma)^2}. \tag{A.353}$$

where the stationary MDP M_{t+1} can be chosen as the minimum over the previous MDPs at time step t such as $|\Pr(s' \mid s, a, M_t) - \Pr(s' \mid s, a, M_{t+1})| \leq L_p$. Rewriting previous equation (A.353), it holds that

$$\left| \left[\mathbb{E}_{\pi,P}[G(\pi, m)] - \min_{m=\left\{m'_t\right\}_{t'=t}^{t+1}} \mathbb{E}_{\pi,P}[G(\pi, m)] \right] \right| \leq \frac{L_r}{1-\gamma} + \frac{\gamma L_P}{(1-\gamma)^2} = L'. \tag{A.354}$$

Now considering non robust objective :

$$\left| J^R(\pi, t) - J^R(\pi, t+1) \right| \tag{A.355}$$

$$= \left| \min_{m=\left\{m'_t\right\}_{t'=t_0}^{t} \in \mathcal{M}^L} \mathbb{E}[G(\pi, m)] - \min_{m=\left\{m'_t\right\}_{t'=t_0}^{t+1} \in \mathcal{M}^L_{t+1}} \mathbb{E}[G(\pi, m)] \right| \tag{A.356}$$

$$= \left| \min_{m=\left\{m'_t\right\}_{t'=t_0}^{t} \in \mathcal{M}^L_t} \left[\mathbb{E}[G(\pi, m)] - \min_{m=\left\{m'_t\right\}_{t'=t_0}^{t} \in \mathcal{M}^L_t} \min_{m=\left\{m'_t\right\}_{t'=t}^{t+1}} \mathbb{E}[G(\pi, m)] \right] \right| \tag{A.357}$$

$$\leq \max_{m=\{m_t\}_{t=t_0}^{t} \in \mathcal{M}^L_t} \left| \left[\mathbb{E}[G(\pi, m)] - \min_{m=\left\{m'_t\right\}_{t'=t}^{t+1}} \mathbb{E}[G(\pi, m)] \right] \right| \tag{A.358}$$

where first equality is the definition of the robust objective, second equality is decomposition of minimum across time steps and final inequality is simply a property of the min such as $|\min a - \min b| \leq \sup |a - b|$.

Finally plugging A.354 in (A.358), it holds that

$$\left| J^R(\pi, t) - J^R(\pi, t+1) \right| \tag{A.359}$$

$$= \left| \min_{m=\left\{m'_t\right\}_{t'=t_0}^{t} \in \mathcal{M}^L} \mathbb{E}_{\pi,P}[G(\pi, m)] - \min_{m=\left\{m'_t\right\}_{t'=t_0}^{t+1} \in \mathcal{M}^L} \mathbb{E}_{\pi,P}[G(\pi, m)] \right| \leq \frac{L_r}{1-\gamma} + \frac{\gamma L_P}{(1-\gamma)^2}. \tag{A.360}$$

$$:= L'. \tag{A.361}$$

Combining t times the previous equation gives the result:

$$\forall t \in \mathbb{N}^+, \forall t_0 \in \mathbb{N}^+, \quad |J^R(\pi, t_0) - J^R(\pi, t_0 + t)| \leq L' t.$$

with $L' := \left(\frac{\gamma}{(1-\gamma)^2} L_P + \frac{1}{1-\gamma} L_r \right)$ □

28 Implementation details

28.1 Algorithm

Algorithm 13: Time-constrained robust training

Input: Time-constrained MDP: $(S, A, \Psi, p_\psi, r, L)$, Agent π, Adversary $\bar{\pi}$

for *each interaction time step* t **do**

- $a_t \sim \pi_t(s_t, \psi_t)$ // Sample an action with Oracle-TC
- $a_t \sim \pi_t(s_t, a_{t-1}, s_{t-1})$ // Sample an action with Stacked-TC
- $a_t \sim \pi_t(s_t)$ // Sample an action with TC
- $\psi_{t+1} \sim \bar{\pi}_\phi(s_t, a_t, \psi_t)$ // Sample the worst TC parameter
- $s_{t+1} \sim p_{\psi_{t+1}}(s_t, a_t)$ // Sample a transition
- $\mathcal{B} \leftarrow \mathcal{B} \cup \{(s_t, a_t, r(s_t, a_t), \psi_t, \psi_{t+1}, s_{t+1})\}$ // Add transition to replay buffer
- $\{s_i, a_i, r(s_i, a_i), \psi_i, \psi_{i+1}, s_{i+1}\}_{i \in [1,N]} \sim \mathcal{B}$ // Sample a mini-batch of transitions
- $\theta_c \leftarrow \theta_c - \alpha \nabla_{\theta_c} L_Q(\theta_c)$ // Critic update phase
- $\theta_a \leftarrow \theta_a - \alpha \nabla_{\theta_a} L_\pi(\theta_a)$ // Actor update
- $\phi_c \leftarrow \phi_c + \alpha \nabla_{\phi_c} L_{\bar{Q}}(\phi_c)$ // Adversary Critic update phase
- $\phi_a \leftarrow \phi_a + \alpha \nabla_{\phi_a} L_{\bar{\pi}}(\phi_a)$ // Adversary update

Note that in Time-constrained robust training Algorithm in section 28.1, L_Q and L_π are as defined by (Fujimoto et al. 2018) double critics and target network updates are omitted here for clarity

In Table 9.8, for the stack algorithm, s_i is defined as $s_i \leftarrow s_i \cup s_{i-1} \cup a_{i-1}$ for Stacked-TC , and for the Oracle-TC version, $s_i \leftarrow s_i \cup \psi_i$.

Loss Function	Equation
$L_{Q_{\theta_c}}$ (TC-RARL)	$\mathbb{E}\left[Q_{\theta_c}(s_i, a_i) - r(s_i, a_i) + \gamma \min_{j=1,2} Q_{\theta_c}(s_{i+1}, \pi(s_{i+1}))\right]$
$L_\pi(\theta_a)$ (TC-RARL)	$-\mathbb{E}\left[Q_{\theta_c}(s_i, \pi_{\theta_a}(s_i))\right]$
$L_{\bar{\pi}}(\theta_a)$ (TC-RARL)	$\mathbb{E}\left[\bar{Q}_{\theta_c}(s_i, a_i, \bar{\pi}(s_i, a_i), \psi_i)\right]$
$L_{\bar{Q}}(\theta_c)$ (TC-RARL)	$\mathbb{E}\left[\bar{Q}_{\theta_c}(s_i, a_i) - r(s_i, a_i) + \gamma \min_{j=1,2} \bar{Q}_{\theta_c}(s_{i+1}, \pi_{\theta_a}(s_{i+1}), \bar{\pi}_{\theta_a}(s_{i+1}, a_{i+1}, \psi_{i+1}))\right]$
$L_{Q_{\theta_c}}$ Shared (TC-M2TD3)	$\mathbb{E}\left[Q_{\theta_c}(s_i, a_i) - r(s_i, a_i) + \gamma \min_{j=1,2} Q_{\theta_c}(s_{i+1}, \pi_{\theta_a}(s_{i+1}), \bar{\pi}_{\theta_a}(s_{i+1}, a_{i+1}, \psi_{i+1}))\right]$
$L_\pi(\theta_a)$ (TC-M2TD3)	$\mathbb{E}\left[Q_{\theta_c}(s_i, a_i, \bar{\pi}_{\theta_a}(s_i, a_i), \psi_i)\right]$
$L_{\bar{\pi}}(\theta_a)$ (TC-M2TD3)	$-\mathbb{E}\left[\bar{Q}_{\theta_c}(s_i, a_i, \bar{\pi}_{\theta_a}(s_i, a_i, \psi_i))\right]$

Table 9.8: Summary of Loss Functions for TD3 in TC-RARL and TC-M2TD3

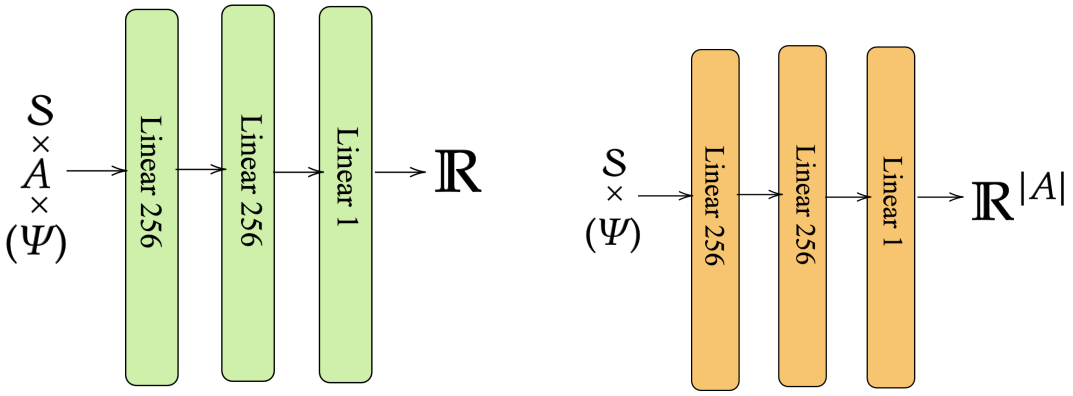
28.2 Neural network architecture

We employ a consistent neural network architecture for both the baseline and our proposed methods for the actor and the critic components. The architecture's design ensures uniformity and comparability across different models.

The critic network is structured with three layers, as depicted in Figure A37.30a, the critic begins with an input layer that takes the state and action as inputs, which then passes through two fully connected linear layers of 256 units each. The final layer is a single linear unit that

outputs a real-valued function, representing the estimated value of the state-action pair.

The actor neural network, shown in Figure A37.30b, also utilizes a three-layer design. It begins with an input layer that accepts the state as input. This is followed by two linear layers, each consisting of 256 units. The output layer of the actor neural network has a dimensionality equal to the number of dimensions of the action space.



(a) Critic neural network architecture **(b)** Actor neural network architecture

Figure A28.13: Actor critic neural network architecture

28.3 M2TD3

We utilized the official M2TD3 Tanabe et al. (2022a) implementation provided by the original authors, accessible via the GitHub repository for M2TD3 and Oracle M2TD3.

For the TC-M2TD3 or variants, we implemented the M2TD3 algorithm as specified. To simplify our approach, we omitted the implementation of the multiple $\hat{\psi}$ network and the system for resetting $\hat{\psi}$. We replace with an adversary which $\bar{\pi} : \mathcal{S} \times \mathcal{A} \times \Psi \to \Psi$ which minimize $Q(s, a, \psi)$.

28.4 TD3

We adopted the TD3 implementation from the CleanRL library, as detailed in Huang et al. (2022).

29 Sanity check on the adversary training in the time-constrained evaluation

A natural question arises regarding the worst time-constrained perturbation. Whether we adequately trained the adversary in the loop, or its suboptimal performance might lead to overestimating the trained agent reward against the worst-case time-constrained perturbation. We monitored the adversary's performance during its training against a fixed agent to address this. The attached figure shows the episodic reward (from the agent's perspective) during the adversary's training over 5 million timesteps, with a perturbation radius of $L = 0.001$. Each

Hyperparameter	**Default Value**
Policy Std Rate	0.1
Policy Noise Rate	0.2
Noise Clip Policy Rate	0.5
Noise Clip Omega Rate	0.5
Omega Std Rate	1.0
Min Omega Std Rate	0.1
Maximum Steps	5e6
Batch Size	100
Hatomega Number	5
Replay Size	1e6
Policy Hidden Size	256
Critic Hidden Size	256
Policy Learning Rate	3e-4
Critic Learning Rate	3e-4
Policy Frequency	2
Gamma	0.99
Polyak	5e-3
Hatomega Parameter Distance	0.1
Minimum Probability	5e-2
Hatomega Learning Rate (ho_lr)	3e-4
Optimizer	Adam

Table 9.9: Hyperparameters for the M2TD3 Agent

curve is an average of over 10 seeds. The plots show a rapid decline in reward during the initial stages of training, followed by quick stabilization. The episodic reward stabilizes early in the Ant (Figure A29.14a) environment, indicating quick convergence. Similarly, in the HalfCheetah

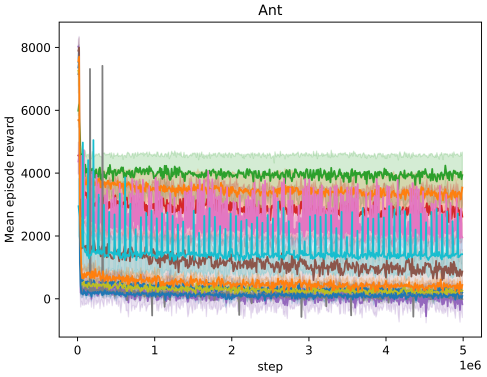
Hyperparameter	**Default Value**
Maximum Steps	5e6
Buffer Size	1×10^6
Learning Rate	3×10^{-4}
Gamma	0.99
Tau	0.005
Policy Noise	0.2
Exploration Noise	0.1
Learning Starts	2.5×10^4
Policy Frequency	2
Batch Size	256
Noise Clip	0.5
Action Min	-1
Action Max	1
Optimizer	Adam

Table 9.10: Hyperparameters for the TD3 Agent

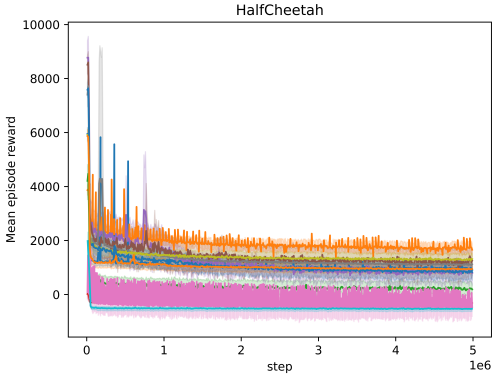
(Figure A29.14b) environment, the reward shows a sharp initial decline and stabilizes, suggesting effective training. For Hopper (Figure A29.14c), the reward decreases and then levels off, reflecting adversary convergence. Although the reward is more variable in the HumanoidStandup (Figure A29.14d) environment, it ultimately reaches a steady state, confirming adequate training. Finally, in the Walker environment, the reward pattern demonstrates a quick drop followed by stabilization, indicating convergence. These observations confirm that the adversaries were not undertrained. The rapid convergence to a stable performance across all environments ensures the accuracy of the worst time-constrained perturbations estimated during training.

30 Uncertainty set in MuJoCo environments

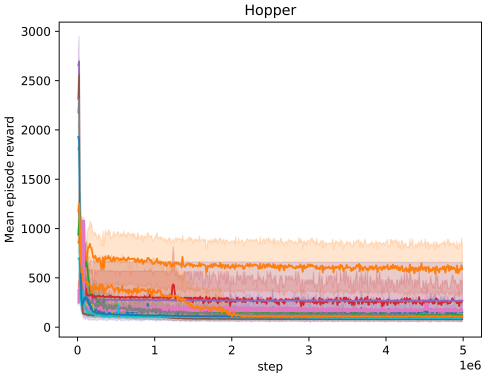
The experiments of Section 6.5 follow the evaluation protocol proposed by (Tanabe et al. 2022a) and based on MuJoCo environments (Todorov et al. 2012). These environments are designed with a 3D uncertainty sets. Table 9.11 lists all environments evaluated and their uncertainty sets. The uncertainty sets column defines the ranges of variation for the parameters within each environment. The reference parameters column indicates the nominal or default values. The



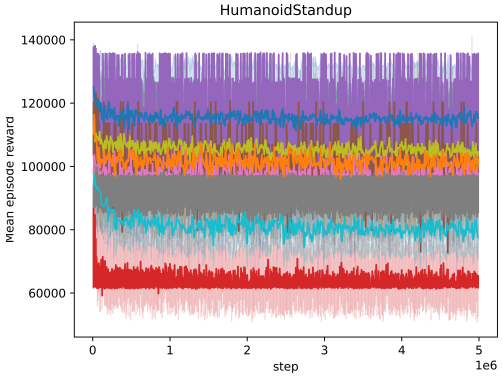
(a) Ant: Episodic reward of the trained agent during adversary training



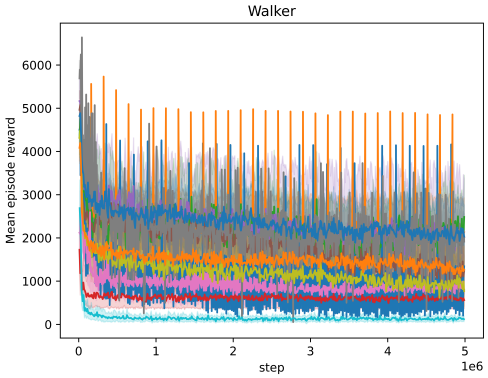
(b) HalfCheetah: Episodic reward of the trained agent during adversary training



(c) Hopper: Episodic reward of the trained agent during adversary training



(d) HumanoidStandup: Episodic reward of the trained agent during adversary training



(e) Walker: Episodic reward of the trained agent during adversary training

(f) Legend for algorithm

Figure A29.14: Episodic reward of the trained agent during the training of the adversary across different environments. Each plot represents the performance over 5 million timesteps, with rewards averaged across 10 seeds. The perturbation radius is set to $L = 0.001$ for all adversaries.

uncertainty parameters column describes the physical meaning of each parameter.

Table 9.11: List of environment and parameters for the experiments

Environment	Uncertainty set $\mathcal{P}$	Reference values	Uncertainty parameters
Ant	$[0.1, 3.0] \times [0.01, 3.0] \times [0.01, 3.0]$	$(0.33, 0.04, 0.06)$	torso mass; front left leg mass; front right leg mass
HalfCheetah	$[0.1, 4.0] \times [0.1, 7.0] \times [0.1, 3.0]$	$(0.4, 6.36, 1.53)$	world friction; torso mass; back thigh mass
Hopper	$[0.1, 3.0] \times [0.1, 3.0] \times [0.1, 4.0]$	$(1.00, 3.53, 3.93)$	world friction; torso mass; thigh mass
HumanoidStandup	$[0.1, 16.0] \times [0.1, 5.0] \times [0.1, 8.0]$	$(8.32, 1.77, 4.53)$	torso mass; right foot mass; left thigh mass
Walker	$[0.1, 4.0] \times [0.1, 5.0] \times [0.1, 6.0]$	$(0.7, 3.53, 3.93)$	world friction; torso mass; thigh mass

31 Raw results

Table 9.12 reports the non-normalized time-constrained (with a radius of $L = 0.001$) worst-case scores, averaged across 10 independent runs for each benchmark. Table 9.27 reports the static worst case score obtained by each agent across a grid of environments, also averaged across 10 independent runs for each benchmark. Table 9.28 reports the static average case score obtained by each agent across a grid of environments, also averaged across 10 independent runs for each benchmark.

31.1 Fixed adversary evaluation

At the beginning of each episode, $\psi_0 \sim \mathcal{U}(\Psi)$ is selected for every fixed adversary. The episode length is 1000 steps. To begin with, the random fixed adversary simulates stochastic changes. It selects a parameter ψ_t at each timestep within a radius of $L = 0.1$, which is 100 times larger than in our training methods. This tests the agents' adaptability to unexpected changes. In contrast, the cosine fixed adversary introduces deterministic changes using a cosine function. The radius of $L = 0.1$ scales the frequency of the cosine function, ensuring smooth and periodic variations. Additionally, a phase shift at the start of each episode ensures different starting points. Meanwhile, the linear fixed adversary employs a linear function. The parameters change linearly from the initial value to either one of a vertex of the uncertainty set Ψ over 1000 steps. Furthermore, the exponential fixed adversary uses an exponential function. Parameters change exponentially from the initial value to either of a vertex of the uncertainty set Ψ over 1000 steps. This ensures smooth and predictable variations. Similarly, the logarithmic fixed adversary uses a logarithmic function. Parameters change logarithmically from the initial value to either of a vertex of the uncertainty of the uncertainty set Ψ over 1000 steps, ensuring smooth and predictable variations. Agents trained under the time-constrained framework outperform all

Table 9.12: Avg. of time-constrained worst-case performance over 10 seeds for each method

Environment Method	Ant	HalfCheetah	Hopper	HumanoidStandup	Walker
Oracle M2TD3	5768 ± 395	3521 ± 187	1241 ± 125	116232 ± 1454	4559 ± 757
Oracle RARL	4387 ± 667	-50 ± 99	344 ± 113	68979 ± 10641	1811 ± 342
Oracle TC-M2TD3	7268 ± 704	7507 ± 284	$\mathbf{3386 \pm 323}$	114411 ± 16973	5344 ± 536
Oracle TC-RARL	$\mathbf{7534 \pm 781}$	$\mathbf{7526 \pm 311}$	3169 ± 311	101182 ± 12083	4783 ± 382
Stacked TC-M2TD3	6502 ± 450	6377 ± 517	3047 ± 394	85524 ± 11448	$\mathbf{5724 \pm 828}$
Stacked TC-RARL	6955 ± 690	5319 ± 223	1747 ± 153	107913 ± 5514	4152 ± 483
TC-M2TD3	7181 ± 591	6516 ± 232	2511 ± 45	$\mathbf{129183 \pm 9120}$	4964 ± 531
TC-RARL	7473 ± 361	4989 ± 284	1475 ± 158	108669 ± 17764	3971 ± 351
DR	7247 ± 925	4986 ± 363	1642 ± 104	109618 ± 11479	4380 ± 488
M2TD3	5622 ± 435	3671 ± 405	1120 ± 220	102839 ± 12987	4078 ± 644
RARL	4348 ± 574	382 ± 366	240 ± 104	106768 ± 4051	2388 ± 559
TD3	2259 ± 424	1808 ± 503	777 ± 407	104877 ± 12063	1893 ± 361

Table 9.13: Avg. of raw static worst-case performance over 10 seeds for each method

	Ant	HalfCheetah	Hopper	Humanoid	Walker
dr	19.78 ± 394.84	2211.48 ± 915.64	245.01 ± 167.21	64886.87 ± 30048.79	1318.36 ± 777.51
m2td3	2322.73 ± 649.3	2031.9 ± 409.7	273.6 ± 131.9	71900.97 ± 24317.35	2214.16 ± 1330.4
oracle m2td3	2370.93 ± 473.56	319.67 ± 599.26	267.41 ± 111.47	93123.84 ± 26696.17	736.59 ± 944.76
oracle rarl	1396.88 ± 777.46	-278.84 ± 54.36	167.5 ± 38.2	45635.24 ± 15974.44	459.74 ± 437.02
oracle tc m2td3	120.74 ± 618.23	4273.31 ± 246.91	168.7 ± 217.94	58687.26 ± 22321.77	710.99 ± 799.08
oracle tc rarl	1328.27 ± 890.49	3458.52 ± 893.22	150.54 ± 33.12	73276.78 ± 9110.33	1299.88 ± 812.63
rarl	960.11 ± 744.01	-211.8 ± 218.73	170.46 ± 45.73	67821.86 ± 21555.24	360.31 ± 186.06
stacked tc m2td3	-242.98 ± 212.98	3467.34 ± 418.64	289.37 ± 182.18	58515.04 ± 19186.25	2475.58 ± 1057.03
stacked tc rarl	37.77 ± 320.71	1414.37 ± 876.91	344.37 ± 190.1	77357.17 ± 18186.34	1518.86 ± 668.13
td3	-123.64 ± 824.35	-546.21 ± 158.81	69.3 ± 42.77	64577.24 ± 16606.51	114.41 ± 211.05
tc m2td3	-271.34 ± 191.15	3286.67 ± 603.14	333.36 ± 60.04	73428.2 ± 17879.28	2603.59 ± 706.63
tc rarl	209.04 ± 575.89	1738.59 ± 782.71	376.01 ± 155.4	74840.68 ± 33496.45	1513.65 ± 1239.3

Table 9.14: Avg. of raw static average case performance over 10 seeds for each method

env name algo-name	Ant	HalfCheetah	Hopper	HumanoidStandup	Walker
dr	7500.88 ± 143.38	6170.33 ± 442.57	1688.36 ± 225.59	110939.89 ± 22396.41	4611.24 ± 463.42
m2td3	5577.41 ± 316.95	4000.98 ± 314.76	1193.32 ± 254.9	109598.43 ± 12992.35	4311.2 ± 877.89
oracle m2td3	5958.21 ± 237.32	4930.18 ± 390.96	1249.62 ± 212.74	118273.54 ± 13891.06	4616.05 ± 407.94
oracle rarl	4684.83 ± 648.14	36.19 ± 216.52	380.39 ± 110.14	76920.58 ± 26135.3	1451.39 ± 1132.87
oracle-tc m2td3	7739.65 ± 254.65	9536.92 ± 429.14	3281.92 ± 61.79	119737.21 ± 12697.2	5442.85 ± 499.78
oracle-tc-rarl	7889.1 ± 56.0	9474.0 ± 341.69	3071.17 ± 220.39	104348.01 ± 26249.98	5220.2 ± 318.07
rarl	4650.55 ± 395.03	206.71 ± 887.25	276.37 ± 52.42	104764.87 ± 17400.85	2493.26 ± 1113.74
stacked tc m2td3	6912.76 ± 1116.81	8583.55 ± 479.97	3124.06 ± 133.27	88039.74 ± 15138.11	5809.54 ± 703.92
stacked-tc-rarl	7123.07 ± 332.33	6130.71 ± 384.05	2072.75 ± 306.48	110843.2 ± 19887.32	4596.79 ± 619.2
vanilla	2600.43 ± 1468.87	2350.58 ± 357.12	733.18 ± 382.06	100533.0 ± 12298.37	2965.47 ± 685.39
vanilla-tcm2td3	7366.9 ± 169.58	8467.64 ± 397.42	2756.5 ± 273.91	130305.38 ± 22865.1	5070.71 ± 315.7
vanilla-tc-rarl	7558.58 ± 198.37	6092.61 ± 365.68	1558.26 ± 242.17	108635.71 ± 19848.21	4325.42 ± 283.04

Environment Method	Ant	HalfCheetah	Hopper	HumanoidStandup	Walker
Oracle TC-M2TD3	7782 ± 915	$\mathbf{8805 \pm 165}$	$\mathbf{2365 \pm 199}$	116791 ± 12572	5148 ± 558
Oracle TC-RARL	$\mathbf{8041 \pm 470}$	8727 ± 227	2120 ± 96	107733 ± 11975	4896 ± 326
Oracle M2TD3	5830 ± 542	4445 ± 186	1222 ± 111	118861 ± 1365	4584 ± 787
Oracle RARL	4628 ± 514	-51 ± 60	370 ± 141	81583 ± 16526	1829 ± 356
Stacked TC-M2TD3	6888 ± 738	7400 ± 385	2114 ± 138	88436 ± 10750	$\mathbf{5278 \pm 845}$
Stacked TC-RARL	7045 ± 904	5992 ± 427	1940 ± 93	106213 ± 6770	4430 ± 389
TC-M2TD3	7156 ± 692	7530 ± 185	2157 ± 112	$\mathbf{129599 \pm 13556}$	4931 ± 568
TC-RARL	7554 ± 948	5751 ± 482	1445 ± 203	105144 ± 16813	4112 ± 329
DR	7572 ± 629	6048 ± 349	1416 ± 168	105677 ± 16333	4371 ± 431
M2TD3	5588 ± 516	4180 ± 70	1018 ± 271	107692 ± 10414	4176 ± 783
RARL	4347 ± 567	240 ± 250	390 ± 130	103583 ± 9217	1925 ± 501
TD3	4017 ± 518	2028 ± 1250	1944 ± 246	91205 ± 11350	2860 ± 419

Table 9.15: Avg. performance against time-constrained fixed random adversary with a radius $L = 0.1$ over 10 seeds for each method

baselines in all environments for each fixed adversary, except when compared to the oracle TC method, which has access to ψ. In this case, the stacked-TC or TC methods outperform all baselines in all environments for the cosine, logarithmic, and exponential adversaries and outperform the fixed adversary baseline in 4 out of 5 instances for the random and linear fixed adversaries.

31.2 Agents training curve

We conducted training for each agent over a duration of 5 million steps, closely monitoring the cumulative rewards obtained over a trajectory spanning 1,000 steps. To enhance the reliability of our results, we averaged the performance curves across 10 different seeds. The graphs in Figures A35.26 to A31.25 illustrate how different training methods, including Domain Randomization, M2TD3, RARL, Oracle RARL ,Oracle M2TD3, TC RARL, TC M2TD3, Stacked TC RARL and Stacked TC M2TD3, impact agent performance across various environments.

Table 9.16: Avg. performance against time-constrained fixed cosine adversary with a radius $L = 0.1$ over 10 seeds for each method

Environment Method	Ant	HalfCheetah	Hopper	HumanoidStandup	Walker
Oracle M2TD3	5528 ± 637	3453 ± 266	1016 ± 48	119813 ± 3281	3589 ± 863
Oracle RARL	4550 ± 626	-79 ± 34	371 ± 140	74116 ± 7890	1593 ± 326
Oracle TC-M2TD3	$\mathbf{7586 \pm 1345}$	$\mathbf{8174 \pm 383}$	$\mathbf{1946 \pm 104}$	115506 ± 12470	4464 ± 781
Oracle TC-RARL	7522 ± 1435	7838 ± 810	1735 ± 138	110535 ± 12702	4442 ± 591
Stacked TC-M2TD3	6269 ± 849	7173 ± 509	1734 ± 157	88157 ± 10654	$\mathbf{4888 \pm 567}$
Stacked TC-RARL	6510 ± 1395	5385 ± 445	1519 ± 118	105696 ± 5243	3848 ± 404
TC-M2TD3	6350 ± 769	6797 ± 609	1413 ± 167	$\mathbf{130892 \pm 11544}$	4611 ± 632
TC-RARL	7124 ± 912	5109 ± 348	1172 ± 129	102864 ± 13308	3548 ± 545
DR	6975 ± 992	5490 ± 384	1091 ± 169	109227 ± 17068	3851 ± 612
M2TD3	5330 ± 684	3634 ± 321	938 ± 158	108136 ± 9755	4126 ± 644
RARL	4153 ± 602	154 ± 261	363 ± 58	103366 ± 7604	1689 ± 465
TD3	4025 ± 557	2784 ± 370	1317 ± 189	94352 ± 10101	2020 ± 355

Table 9.17: Avg. performance against a fixed linear adversary over 10 seeds for each method

Environment Method	Ant	HalfCheetah	Hopper	HumanoidStandup	Walker
Oracle M2TD3	5811 ± 121	3560 ± 167	1216 ± 326	118829 ± 846	4431 ± 615
Oracle RARL	4447 ± 600	-122 ± 64	308 ± 62	81498 ± 12860	1503 ± 450
Oracle TC-M2TD3	7919 ± 595	$\mathbf{7495 \pm 268}$	$\mathbf{2983 \pm 252}$	117610 ± 11682	4952 ± 415
Oracle TC-RARL	$\mathbf{8069 \pm 151}$	7443 ± 236	2805 ± 352	110314 ± 9354	4613 ± 257
Stacked TC-M2TD3	7003 ± 812	6365 ± 335	2714 ± 198	89556 ± 11115	$\mathbf{5256 \pm 675}$
Stacked TC-RARL	7328 ± 251	5301 ± 86	1616 ± 137	105137 ± 7903	4234 ± 385
TC-M2TD3	7622 ± 413	6451 ± 246	2228 ± 131	$\mathbf{129501 \pm 10326}$	4844 ± 417
TC-RARL	7675 ± 143	4881 ± 251	1277 ± 288	105566 ± 15551	3906 ± 381
DR	7713 ± 412	5290 ± 103	1419 ± 122	108711 ± 16696	4307 ± 309
M2TD3	5444 ± 225	3810 ± 69	970 ± 323	106311 ± 9771	4128 ± 727
RARL	4651 ± 446	218 ± 138	346 ± 22	101477 ± 8947	1894 ± 515
TD3	3493 ± 475	1462 ± 1246	1722 ± 366	89934 ± 10644	2396 ± 416

Table 9.18: Avg. performance against a fixed logarithmic adversary over 10 seeds for each method

Environment Method	Ant	HalfCheetah	Hopper	HumanoidStandup	Walker
Oracle M2TD3	5561 ± 580	3086 ± 163	957 ± 165	119214 ± 2525	4148 ± 630
Oracle RARL	4911 ± 177	-145 ± 67	293 ± 49	79522 ± 13470	1618 ± 142
Oracle TC-M2TD3	7963 ± 796	$\mathbf{6625 \pm 204}$	$\mathbf{2577 \pm 171}$	116664 ± 11798	4818 ± 451
Oracle TC-RARL	$\mathbf{8061 \pm 821}$	6532 ± 304	2572 ± 177	108213 ± 10684	4375 ± 382
Stacked TC-M2TD3	7315 ± 478	5863 ± 290	2283 ± 122	87691 ± 11133	$\mathbf{4931 \pm 735}$
Stacked TC-RARL	7514 ± 62	4770 ± 145	1426 ± 197	104193 ± 8030	3939 ± 369
TC-M2TD3	7910 ± 90	5657 ± 280	1702 ± 226	$\mathbf{128467 \pm 10762}$	4664 ± 412
TC-RARL	7686 ± 208	4475 ± 238	1082 ± 298	104835 ± 16040	3636 ± 428
DR	7883 ± 67	4721 ± 146	1166 ± 332	106171 ± 16867	3995 ± 313
M2TD3	5371 ± 279	3565 ± 105	802 ± 271	104002 ± 11606	4206 ± 712
RARL	4620 ± 763	231 ± 110	340 ± 44	102004 ± 9925	1919 ± 499
TD3	3678 ± 623	576 ± 983	1389 ± 327	88952 ± 11367	1956 ± 360

Table 9.19: Avg. performance against a fixed exponential adversary over 10 seeds for each method

Environment Method	Ant	HalfCheetah	Hopper	HumanoidStandup	Walker
Oracle M2TD3	5860 ± 93	3780 ± 137	1271 ± 224	119205 ± 1217	4767 ± 815
Oracle RARL	4585 ± 674	-88 ± 79	302 ± 41	82063 ± 13274	1611 ± 342
Oracle TC-M2TD3	7491 ± 624	$\mathbf{8256 \pm 269}$	2894 ± 244	118476 ± 11683	5161 ± 289
Oracle TC-RARL	$\mathbf{7724 \pm 368}$	8000 ± 250	$\mathbf{3036 \pm 293}$	110092 ± 10754	4650 ± 503
Stacked TC-M2TD3	6903 ± 365	7041 ± 302	2721 ± 214	91077 ± 11945	$\mathbf{5310 \pm 882}$
Stacked TC-RARL	7061 ± 222	5741 ± 249	1825 ± 145	104793 ± 6758	4376 ± 342
TC-M2TD3	7318 ± 299	7139 ± 387	2408 ± 113	$\mathbf{129966 \pm 10823}$	4910 ± 663
TC-RARL	7441 ± 133	5326 ± 220	1457 ± 163	106491 ± 14605	4017 ± 439
DR	7389 ± 206	5691 ± 121	1564 ± 99	106290 ± 17502	4224 ± 660
M2TD3	5466 ± 318	3909 ± 332	1062 ± 272	107097 ± 9551	4274 ± 582
RARL	4556 ± 729	228 ± 181	351 ± 24	102096 ± 8291	2053 ± 493
TD3	3771 ± 228	2302 ± 343	2201 ± 219	90496 ± 9487	2768 ± 538

Table 9.20: Average wall-clock time for each algorithm

	Wall-clock time
TD3	14h
M2TD3	16h
RARL	18h
TC	16h
Stacked TC	16h
Oracle TC	16h

32 Computer ressources

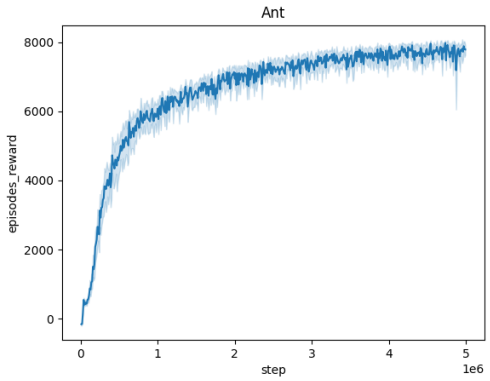
All experiments were run on a desktop machine (Intel i9, 10th generation processor, 64GB RAM) with a single NVIDIA RTX 4090 GPU. Averages and standard deviations were computed from 10 independent repetitions of each experiment.

33 Broader impact

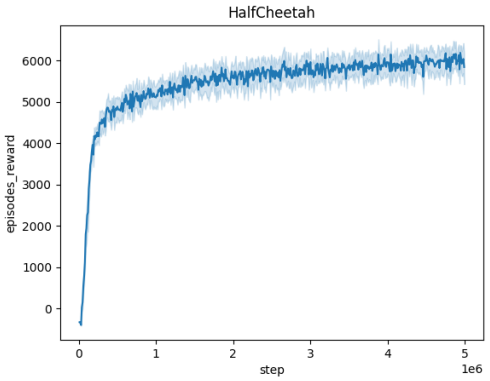
This paper aims to advance robust reinforcement learning. It addresses general mathematical and computational challenges. These challenges may have societal and technological impacts, but we do not find it necessary to highlight them here.

33.1 Limitations

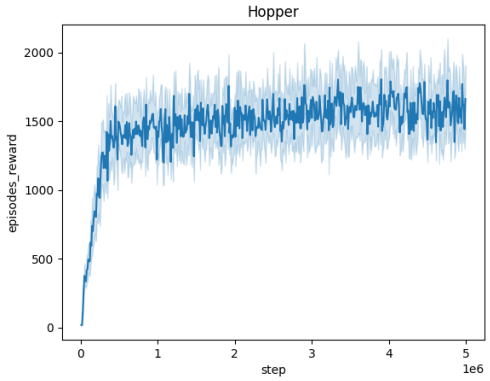
While our proposed Time-Constrained Robust Markov Decision Process (TC-RMDP) framework significantly advances robust reinforcement learning by addressing multifactorial, correlated, and time-dependent disturbances, several limitations must be acknowledged. The TC-RMDP framework assumes that the parameter vector ψ that governs environmental disturbances is known during training. In real-world applications, obtaining such detailed information may not always be feasible. This reliance on precise parameter knowledge limits the practical deployment of our algorithms in environments where ψ cannot be accurately measured or inferred. Our approach assumes that the environment's dynamics can be accurately parameterized and that these parameters remain within a predefined uncertainty set Ψ. This assumption might not hold in more complex or highly dynamic environments where disturbances are not easily parameterized or when the uncertainty set Ψ cannot comprehensively capture all possible variations. Consequently, the robustness of the learned policies might degrade when facing disturbances outside the considered parameter space. Addressing these limitations in future work.



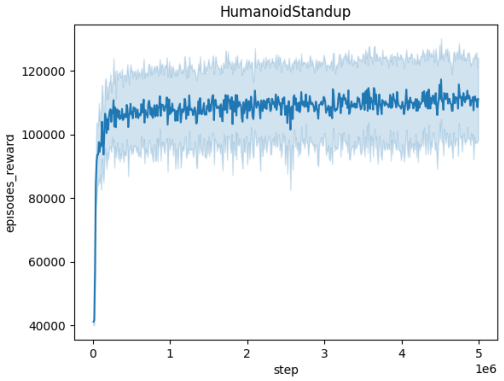
(a) Training curve on Ant with Domain Randomization



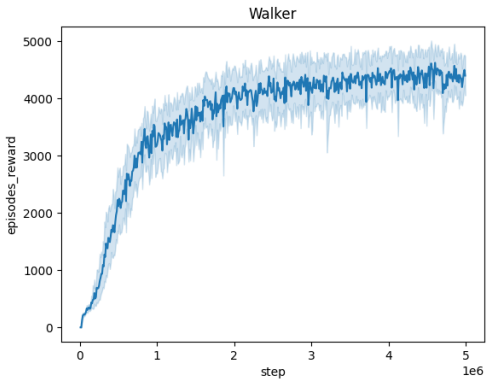
(b) Training curve on HalfCheetah with Domain Randomization



(c) Training curve on Hopper with Domain Randomization

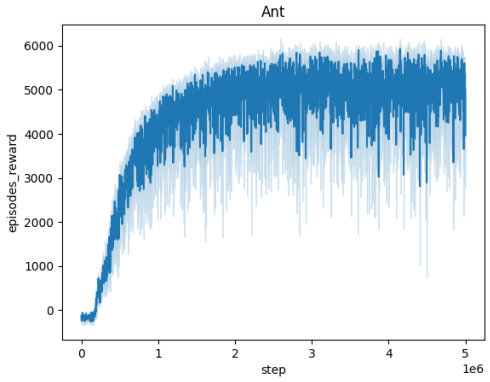


(d) Training curve on HumanoidStandup with Domain Randomization

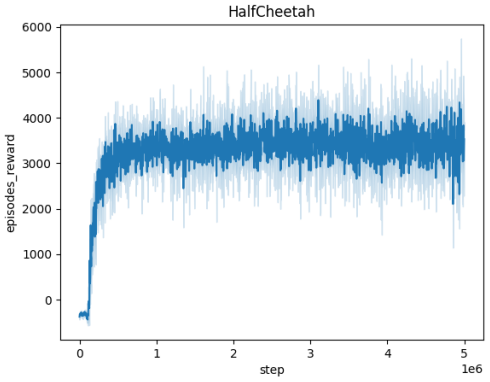


(e) Training curve on Walker with Domain Randomization

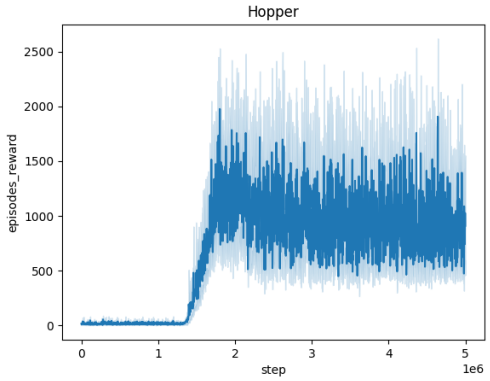
Figure A31.15: Averaged training curves for the Domain Randomization method over 10 seeds



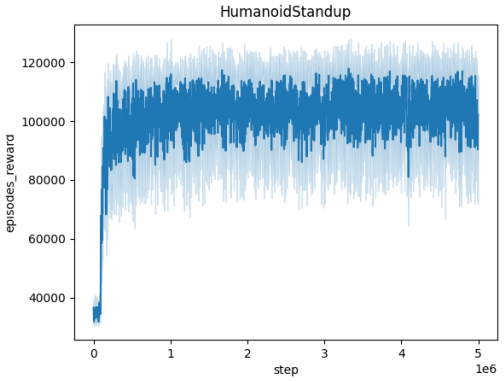
(a) Training curve on Ant with M2TD3



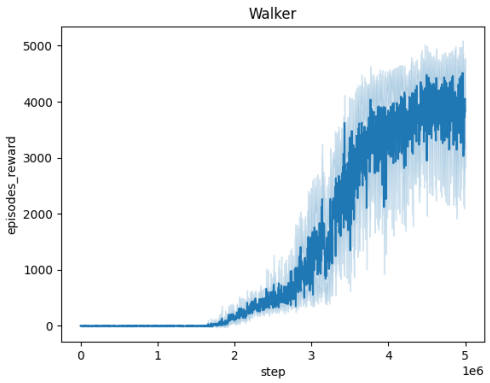
(b) Training curve on HalfCheetah with M2TD3



(c) Training curve on Hopper with M2TD3

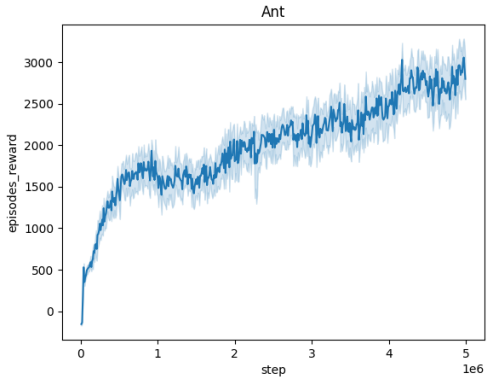


(d) Training curve on HumanoidStandup with M2TD3

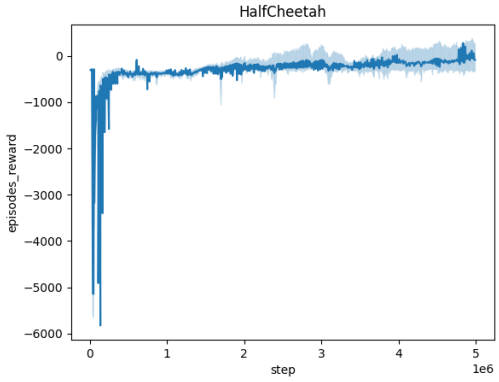


(e) Training curve on Walker with M2TD3

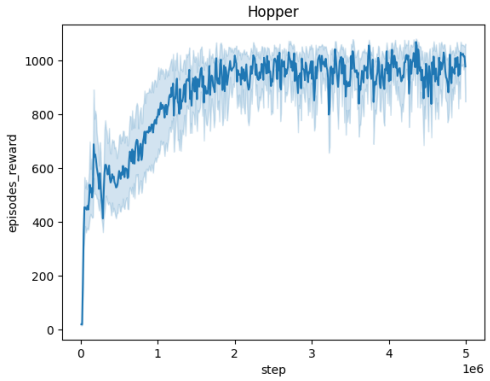
Figure A31.16: Averaged training curves for the M2TD3 method over 10 seeds



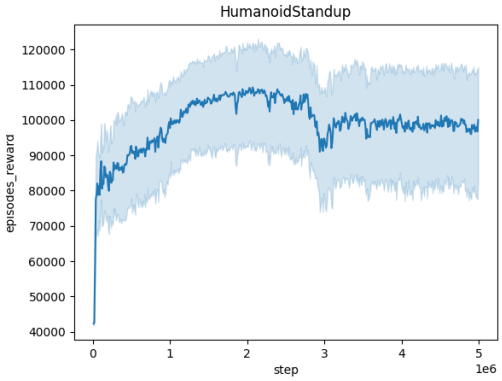
(a) Training curve on Ant with RARL



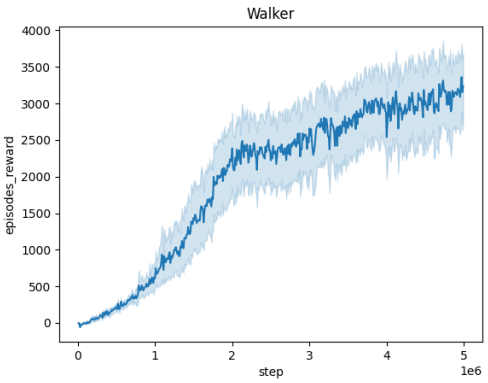
(b) Training curve on HalfCheetah with RARL



(c) Training curve on Hopper with RARL

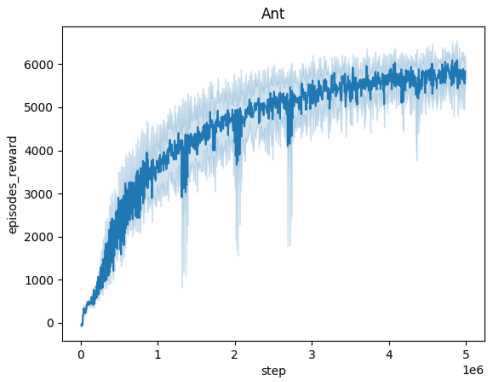


(d) Training curve on HumanoidStandup with RARL

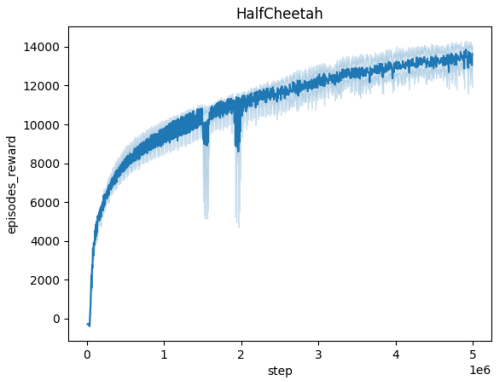


(e) Training curve on Walker with RARL

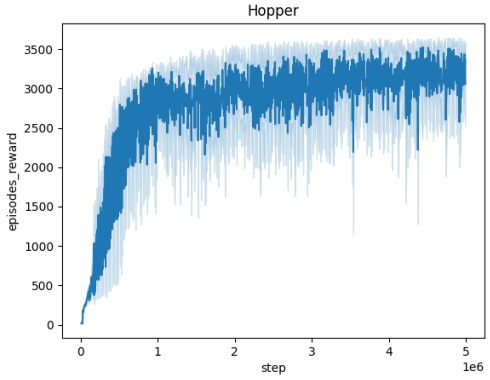
Figure A31.17: Averaged training curves for the RARL method over 10 seeds



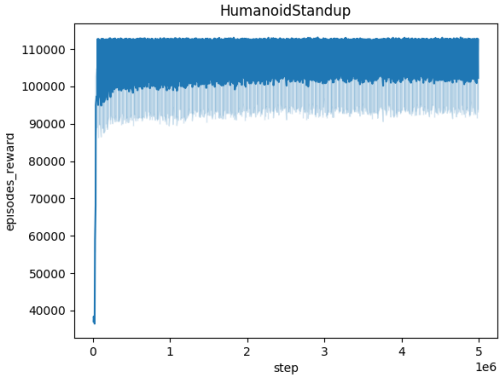
(a) Training curve on Ant with TD3



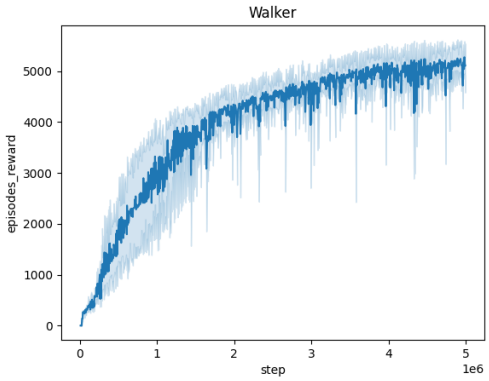
(b) Training curve on HalfCheetah with TD3



(c) Training curve on Hopper with TD3

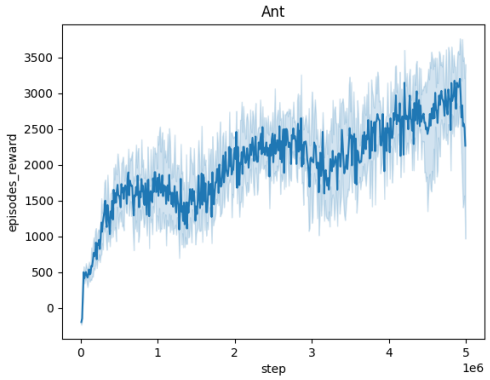


(d) Training curve on HumanoidStandup with TD3

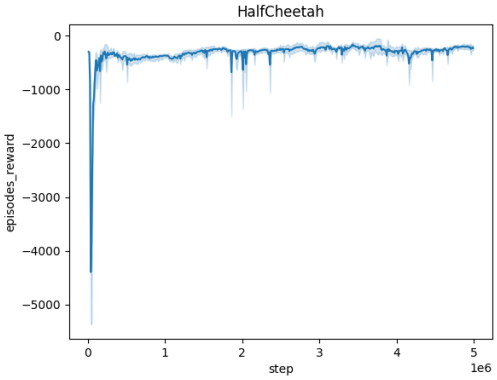


(e) Training curve on Walker with TD3

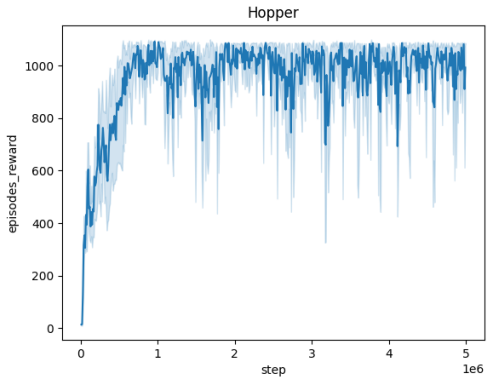
Figure A31.18: Averaged training curves for the TD3 method over 10 seeds



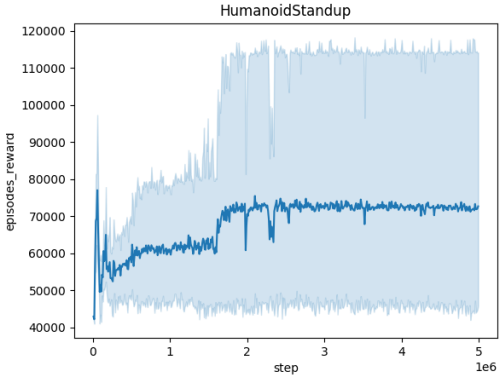
(a) Training curve on Ant with Oracle RARL



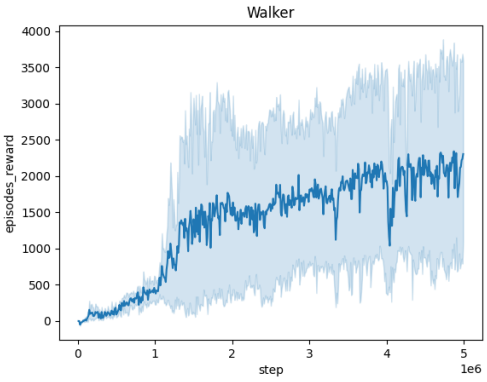
(b) Training curve on HalfCheetah with Oracle RARL



(c) Training curve on Hopper with Oracle RARL

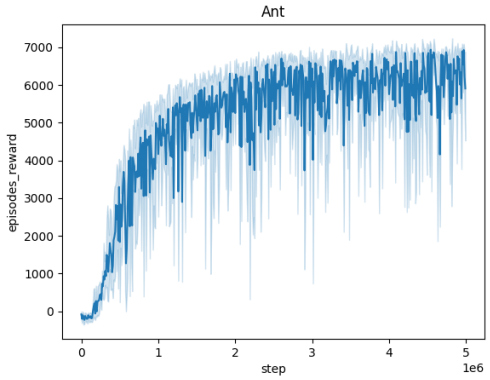


(d) Training curve on HumanoidStandup with Oracle RARL

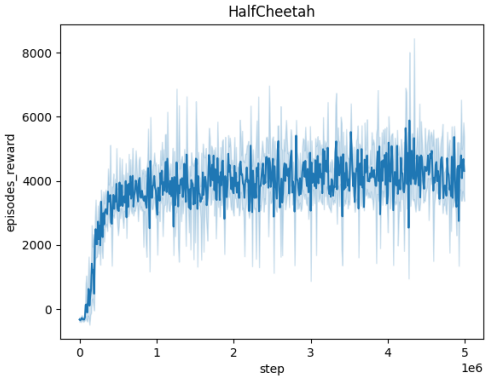


(e) Training curve on Walker with Oracle RARL

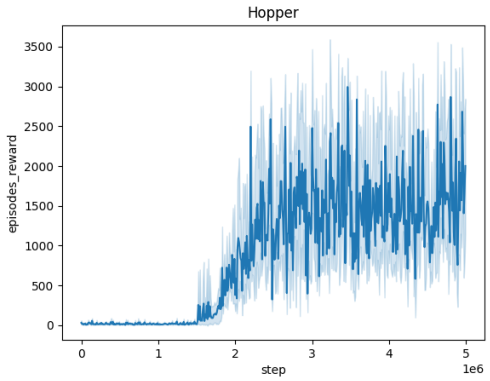
Figure A31.19: Averaged training curves for the Oracle RARL method over 10 seeds



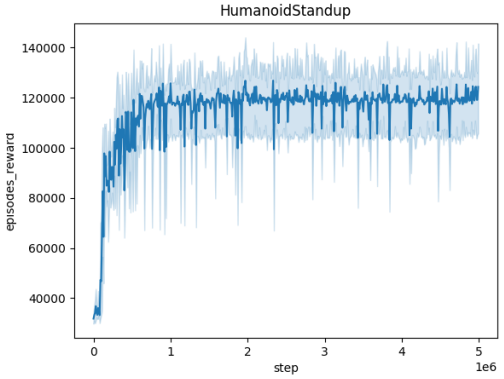
(a) Training curve on Ant with Oracle M2TD3



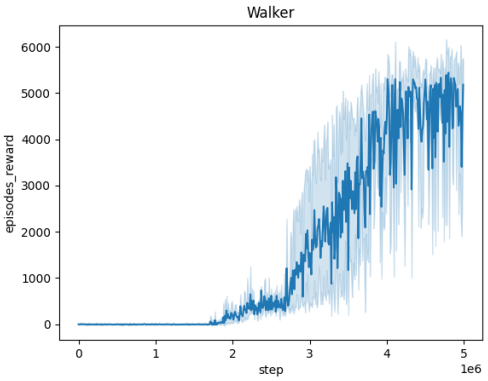
(b) Training curve on HalfCheetah with Oracle M2TD3



(c) Training curve on Hopper with Oracle M2TD3

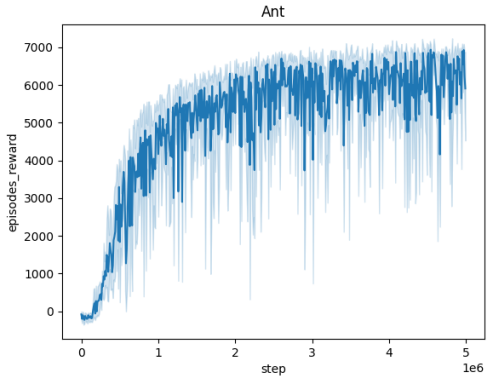


(d) Training curve on HumanoidStandup with Oracle M2TD3

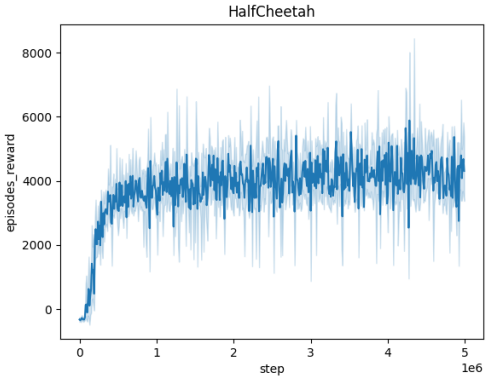


(e) Training curve on Walker with Oracle M2TD3

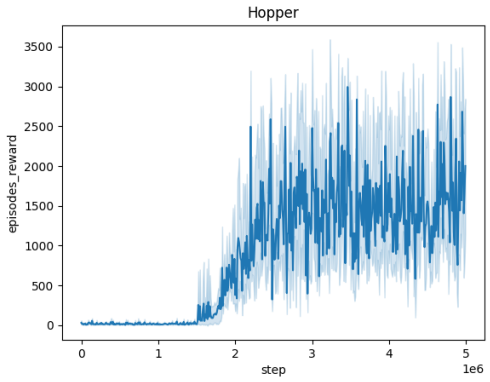
Figure A31.20: Averaged training curves for the Oracle M2TD3 method over 10 seeds



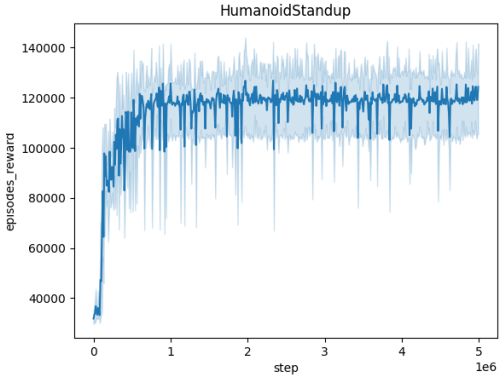
(a) Training curve on Ant with Oracle M2TD3



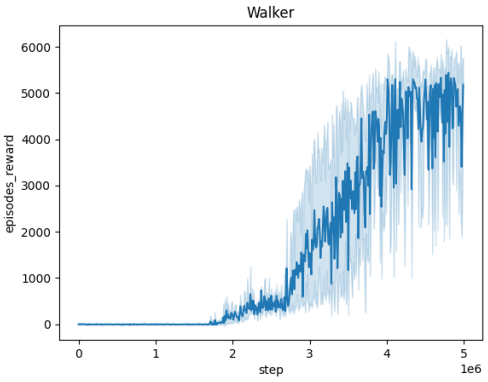
(b) Training curve on HalfCheetah with Oracle M2TD3



(c) Training curve on Hopper with Oracle M2TD3

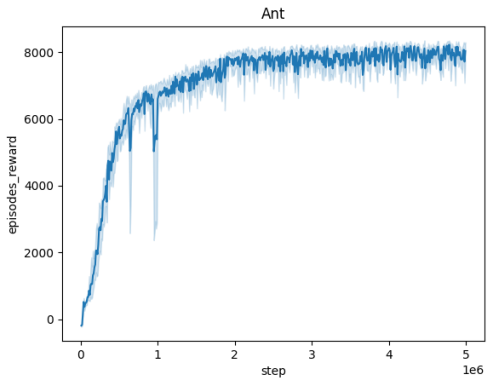


(d) Training curve on HumanoidStandup with Oracle M2TD3

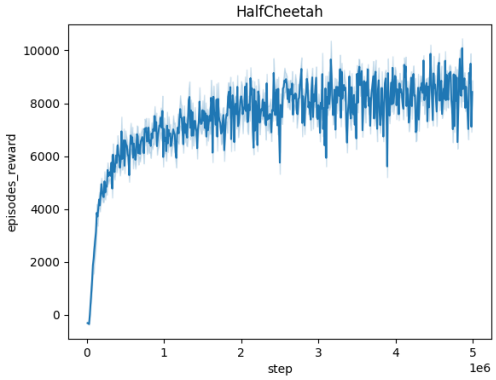


(e) Training curve on Walker with Oracle M2TD3

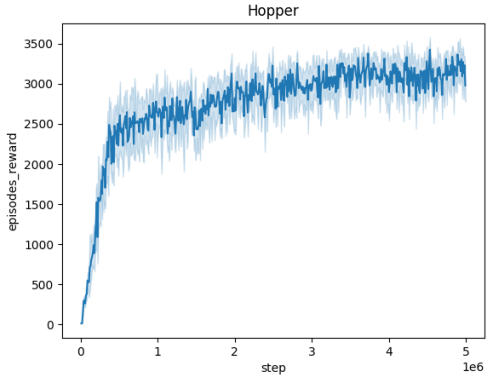
Figure A31.21: Averaged training curves for the Oracle M2TD3 method over 10 seeds



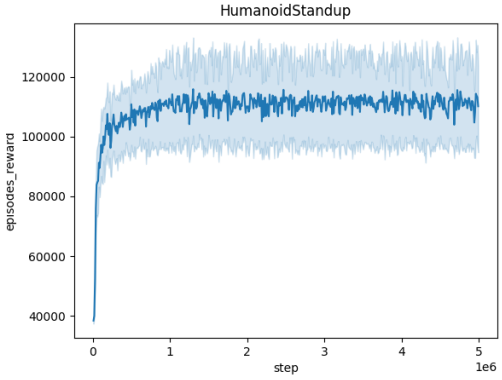
(a) Training curve on Ant with Oracle TC-RARL



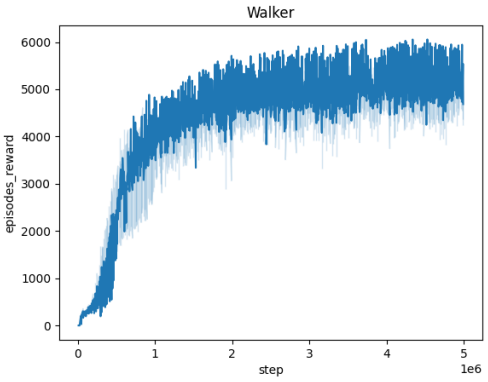
(b) Training curve on HalfCheetah with Oracle TC-RARL



(c) Training curve on Hopper with Oracle TC-RARL

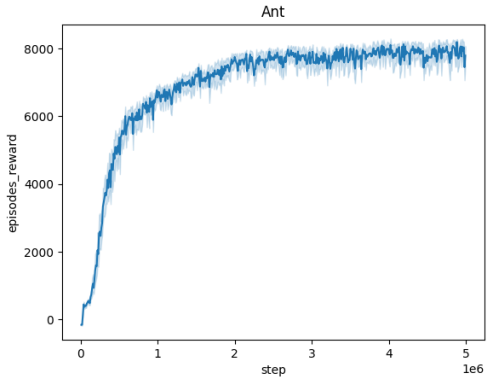


(d) Training curve on HumanoidStandup with Oracle TC-RARL

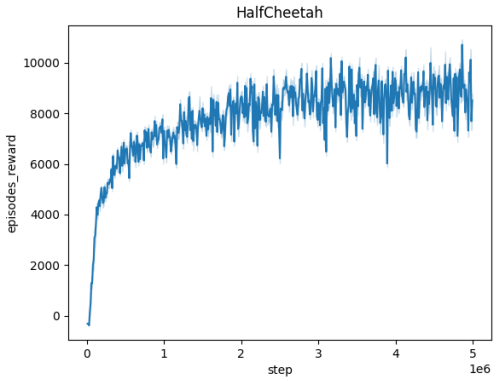


(e) Training curve on Walker with Oracle TC-RARL

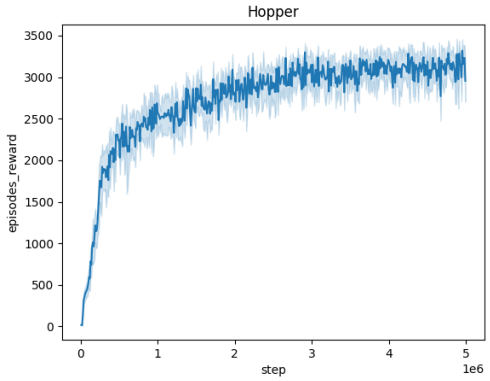
Figure A31.22: Averaged training curves for the Oracle TC-RARL method over 10 seeds



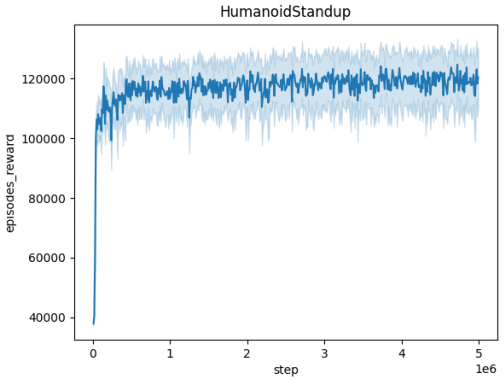
(a) Training curve on Ant with Oracle TC-M2TD3



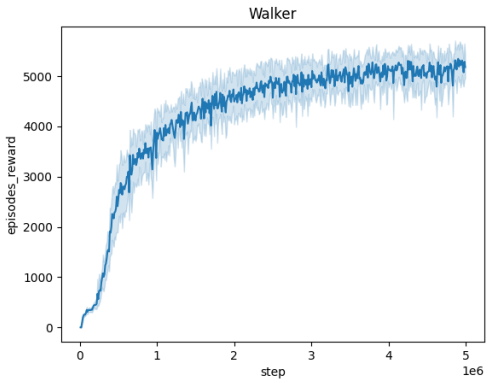
(b) Training curve on HalfCheetah with Oracle TC-M2TD3



(c) Training curve on Hopper with Oracle TC-M2TD3

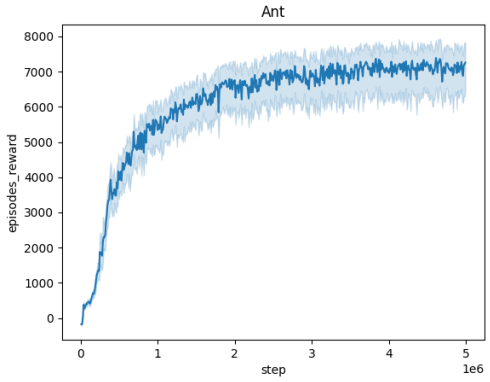


(d) Training curve on HumanoidStandup with Oracle TC-M2TD3

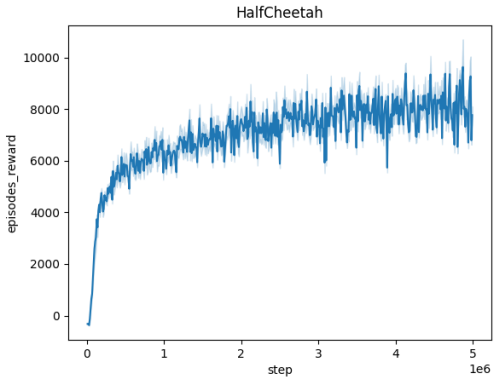


(e) Training curve on Walker with Oracle TC-M2TD3

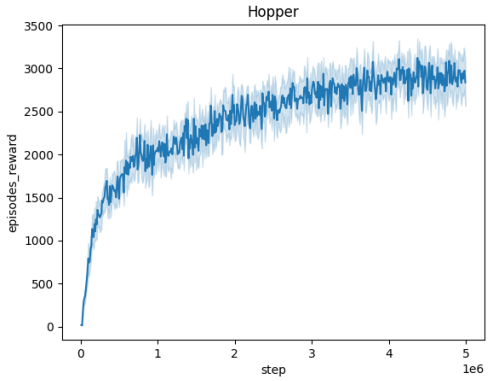
Figure A31.23: Averaged training curves for the Oracle TC-M2TD3 method over 10 seeds



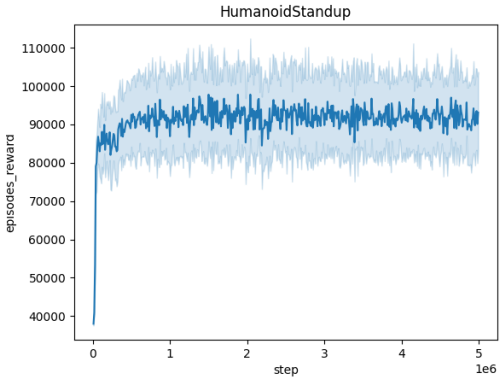
(a) Training curve on Ant with Stacked TC-M2TD3



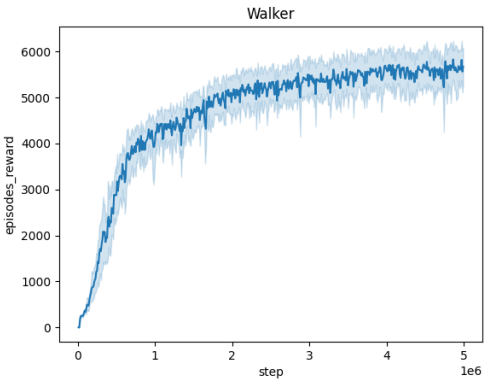
(b) Training curve on HalfCheetah with Stacked TC-M2TD3



(c) Training curve on Hopper with Stacked TC-M2TD3

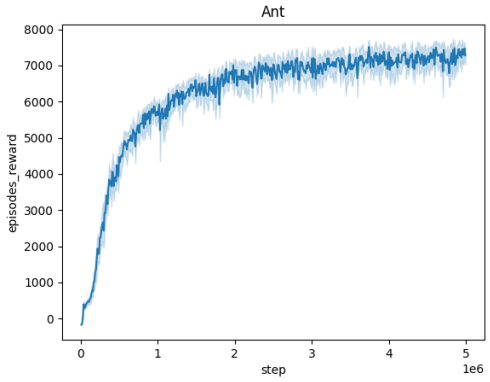


(d) Training curve on HumanoidStandup with Stacked TC-M2TD3

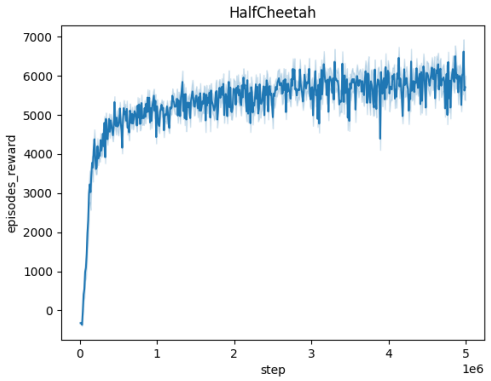


(e) Training curve on Walker with Stacked TC-M2TD3

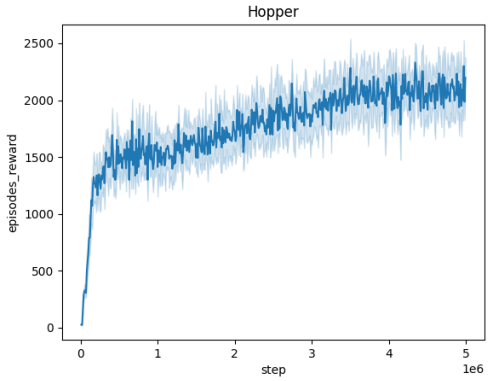
Figure A31.24: Averaged training curves for the Stacked TC-M2TD3 method over 10 seeds



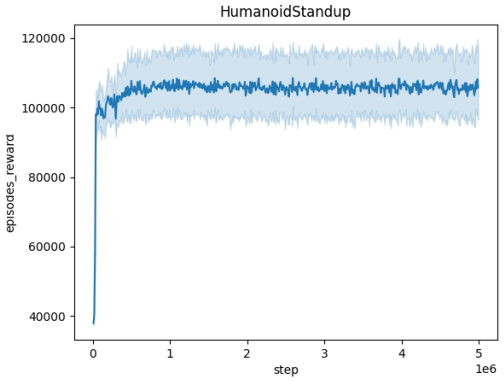
(a) Training curve on Ant with Stacked TC-RARL



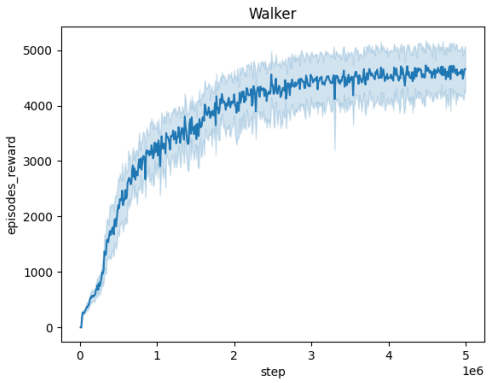
(b) Training curve on HalfCheetah with Stacked TC-RARL



(c) Training curve on Hopper with Stacked TC-RARL



(d) Training curve on HumanoidStandup with Stacked TC-RARL



(e) Training curve on Walker with Stacked TC-RARL

Figure A31.25: Averaged training curves for the Stacked TC-RARL method over 10 seeds

Appendix of Chapter 7

34 Modifiable parameters

The following tables list the parameters that can be modified in different MuJoCo environments used in the Robust Reinforcement Learning Suite. These parameters are accessed and modified through the `set_params` and `get_params` methods in the `ModifiedParamsEnv` interface.

Parameter Name
Torso Mass
Front Left Leg Mass
Front Left Leg Auxiliary Mass
Front Left Leg Ankle Mass
Front Right Leg Mass
Front Right Leg Auxiliary Mass
Front Right Leg Ankle Mass
Back Left Leg Mass
Back Left Leg Auxiliary Mass
Back Left Leg Ankle Mass
Back Right Leg Mass
Back Right Leg Auxiliary Mass
Back Right Leg Ankle Mass

Table 9.21: Modifiable parameters from Ant environment

35 Training curves

We conducted training for each agent over a duration of 5 million steps, closely monitoring the cumulative rewards obtained over a trajectory spanning 1,000 steps. To enhance the reliability of our results, we averaged the performance curves across 10 different seeds.The graphs in Figures A35.26 to A35.29 illustrate how different training methods, including Domain Randomization,

Parameter Name
World Friction
Torso Mass
Back Thigh Mass
Back Shin Mass
Back Foot Mass
Forward Thigh Mass
Forward Shin Mass
Forward Foot Mass

Table 9.22: Modifiable parameters from Halfcheetah environment

Parameter Name
World Friction
Torso Mass
Thigh Mass
Leg Mass
Foot Mass

Table 9.23: Modifiable parameters from Hopper environment

M2TD3, RARL, and TD3 impact agent performance across various environments.

36 Non-normalized results

Table 9.27 reports the non-normalized worst case scores, averaged across 10 independent runs for each benchmark. Table 9.28 reports the average score obtained by each agent across a grid of environments, also averaged across 10 independent runs for each benchmark.

37 Implementation details

37.1 Neural network architecture

We employ the same neural network architecture for all baselines for the actor and the critic components. The architecture's design ensures uniformity and comparability across different models.

The critic network is structured with three layers, as depicted in Figure A37.30a, the critic

Parameter Name
Torso Mass
Lower Waist Mass
Pelvis Mass
Right Thigh Mass
Right Shin Mass
Right Foot Mass
Left Thigh Mass
Left Shin Mass
Left Foot Mass
Right Upper Arm Mass
Right Lower Arm Mass
Left Upper Arm Mass
Left Lower Arm Mass

Table 9.24: Modifiable parameters from Humanoid Stand Up environment

Parameter Name
World Friction
Torso Mass
Thigh Mass
Leg Mass
Foot Mass
Left Thigh Mass
Left Leg Mass
Left Foot Mass

Table 9.25: Modifiable parameters from Walker environment

begins with an input layer that takes the state and action as inputs, then passes through two fully connected linear layers of 256 units each. The final layer is a single linear unit that outputs a real-valued function, representing the estimated value of the state-action pair.

The actor neural network, shown in Figure A37.30b, also utilizes a three-layer design. It

Parameter Name
Pole Mass
Cart Mass

Table 9.26: Modifiable parameters from Inverted Pendulum environment

Table 9.27: Avg. of raw static worst-case performance over 10 seeds for each method

	Ant	HalfCheetah	Hopper	Humanoid StandUp	Walker
DR	19.78 ± 394.84	2211.48 ± 915.64	245.01 ± 167.21	64886.87 ± 30048.79	1318.36 ± 777.51
M2TD3	2322.73 ± 649.3	2031.9 ± 409.7	273.6 ± 131.9	71900.97 ± 24317.35	2214.16 ± 1330.4
RARL	960.11 ± 744.01	-211.8 ± 218.73	170.46 ± 45.73	67821.86 ± 21555.24	360.31 ± 186.06
NR-MDP	-744.94 ± 484.65	-818.64 ± 63.21	5.73 ± 8.87	48318.45 ± 11092.99	16.42 ± 3.5
TD3	-123.64 ± 824.35	-546.21 ± 158.81	69.3 ± 42.77	64577.24 ± 16606.51	114.41 ± 211.05

begins with an input layer that accepts the state as input. This is followed by two linear layers, each consisting of 256 units. The output layer of the actor neural network has a dimensionality equal to the number of dimensions of the action space.

37.2 M2TD3

We use the official M2TD3 Tanabe et al. (2022a) implementation provided by the original authors, accessible via the GitHub repository for M2TD3.

Table 9.28: Avg. of raw static average case performance over 10 seeds for each method

env name algo-name	Ant	HalfCheetah	Hopper	Humanoid Standup	Walker
DR	7500.88 ± 143.38	6170.33 ± 442.57	1688.36 ± 225.59	110939.89 ± 22396.41	4611.24 ± 463.42
M2TD3	5577.41 ± 316.95	4000.98 ± 314.76	1193.32 ± 254.9	109598.43 ± 12992.35	4311.2 ± 877.89
RARL	4650.55 ± 395.03	206.71 ± 887.25	276.37 ± 52.42	104764.87 ± 17400.85	2493.26 ± 1113.74
NR-MDP	4197.80 ± 90.66	1388.90 ± 283.25	340.15 ± 3.65	92972.45 ± 2251.18	1501.05 ± 453.96
TD3	2600.43 ± 1468.87	2350.58 ± 357.12	733.18 ± 382.06	100533.0 ± 12298.37	2965.47 ± 685.39

Hyperparameter	Default Value
Policy Std Rate	0.1
Policy Noise Rate	0.2
Noise Clip Policy Rate	0.5
Noise Clip Omega Rate	0.5
Omega Std Rate	1.0
Min Omega Std Rate	0.1
Maximum Steps	5e6
Batch Size	100
Hatomega Number	5
Replay Size	1e6
Policy Hidden Size	256
Critic Hidden Size	256
Policy Learning Rate	3e-4
Critic Learning Rate	3e-4
Policy Frequency	2
Gamma	0.99
Polyak	5e-3
Hatomega Parameter Distance	0.1
Minimum Probability	5e-2
Hatomega Learning Rate (ho_lr)	3e-4
Optimizer	Adam

Table 9.29: Hyperparameters for the M2TD3 Agent

37.3 TD3

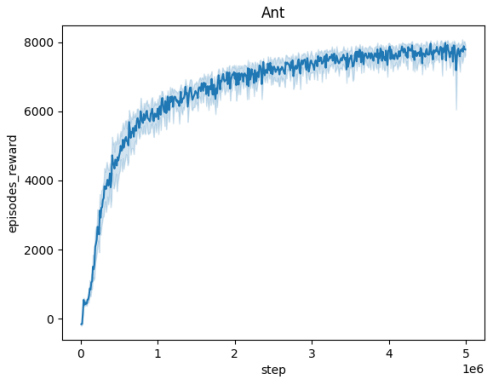
We adopted the TD3 implementation from the CleanRL library, as detailed in Huang et al. (2022).

Hyperparameter	Default Value
Maximum Steps	5e6
Buffer Size	1×10^6
Learning Rate	3×10^{-4}
Gamma	0.99
Tau	0.005
Policy Noise	0.2
Exploration Noise	0.1
Learning Starts	2.5×10^4
Policy Frequency	2
Batch Size	256
Noise Clip	0.5
Action Min	-1
Action Max	1
Optimizer	Adam

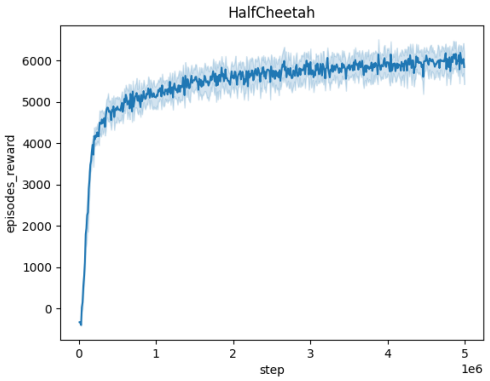
Table 9.30: Hyperparameters for the TD3 Agent

38 Computer ressources

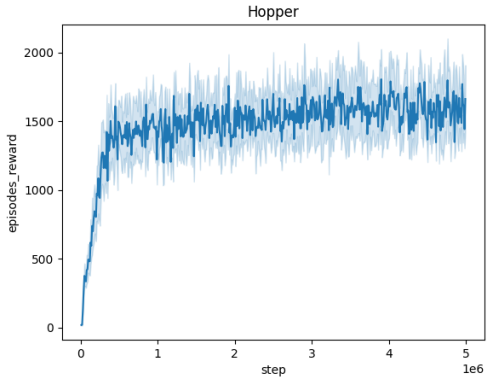
All experiments were run on a desktop machine (Intel i9, 10th generation processor, 64GB RAM) with a single NVIDIA RTX 4090 GPU. Averages and standard deviations were computed from 10 independent repetitions of each experiment.



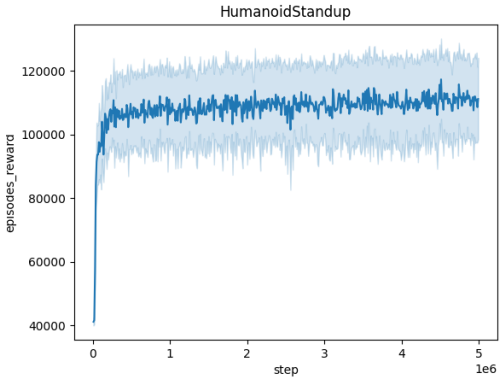
(a) Training curve on Ant with Domain Randomization



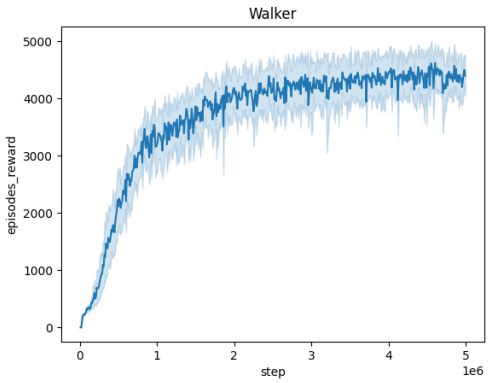
(b) Training curve on HalfCheetah with Domain Randomization



(c) Training curve on Hopper with Domain Randomization

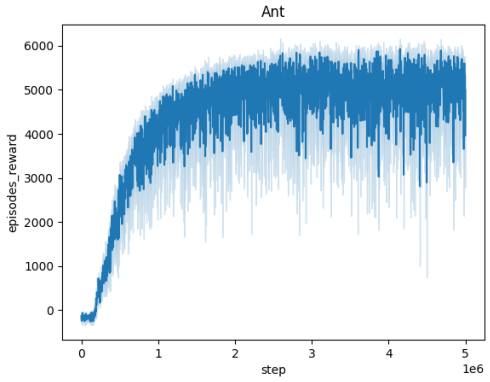


(d) Training curve on HumanoidStandup with Domain Randomization

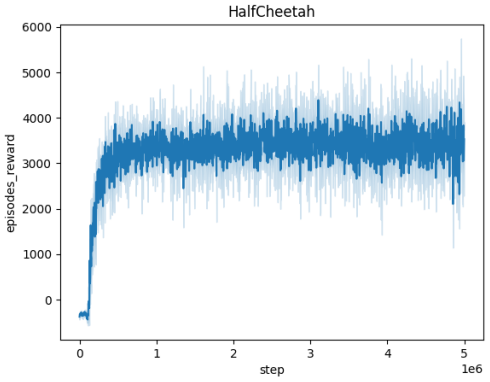


(e) Training curve on Walker with Domain Randomization

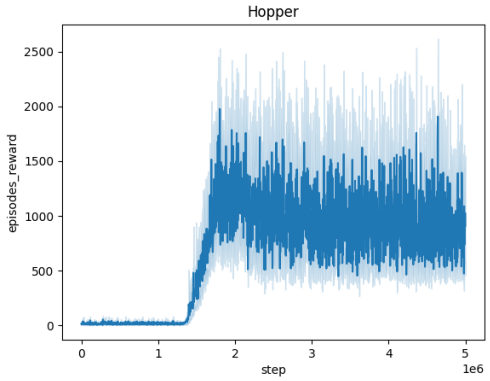
Figure A35.26: Averaged training curves for the Domain Randomization method over 10 seeds



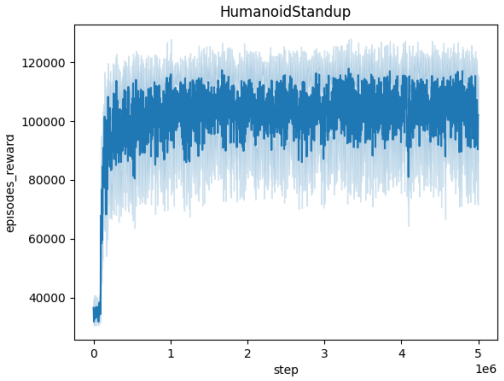
(a) Training curve on Ant with M2TD3



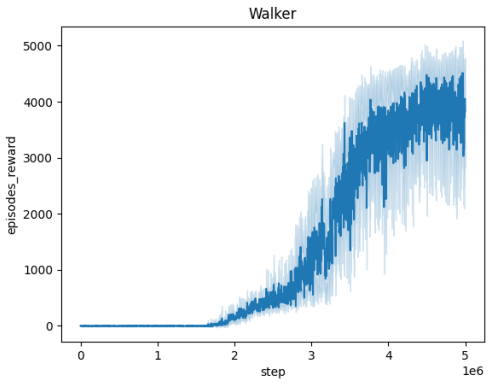
(b) Training curve on HalfCheetah with M2TD3



(c) Training curve on Hopper with M2TD3

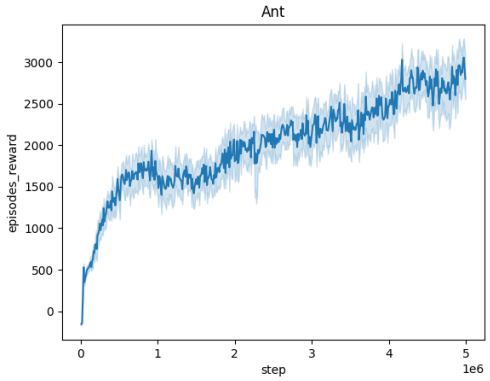


(d) Training curve on HumanoidStandup with M2TD3

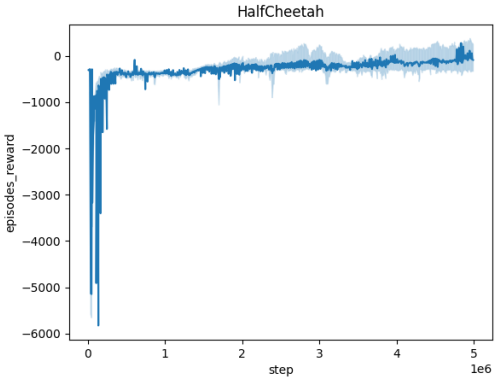


(e) Training curve on Walker with M2TD3

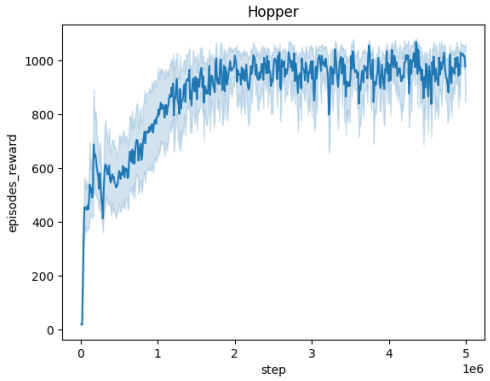
Figure A35.27: Averaged training curves for the M2TD3 method over 10 seeds



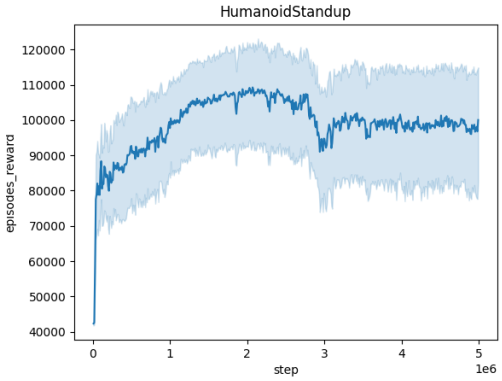
(a) Training curve on Ant with RARL



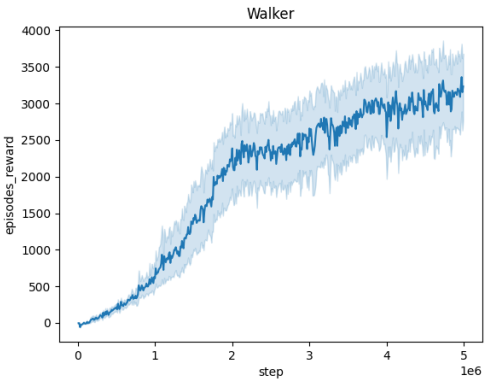
(b) Training curve on HalfCheetah with RARL



(c) Training curve on Hopper with RARL

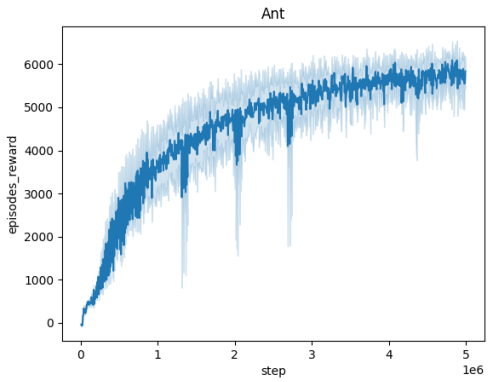


(d) Training curve on HumanoidStandup with RARL

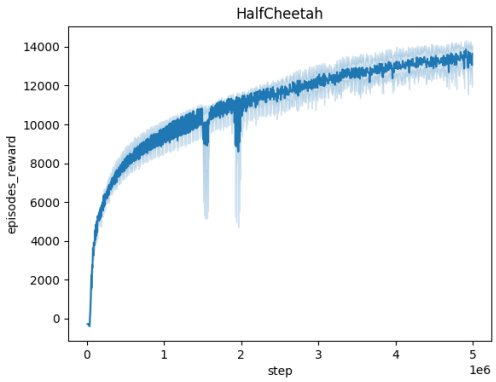


(e) Training curve on Walker with RARL

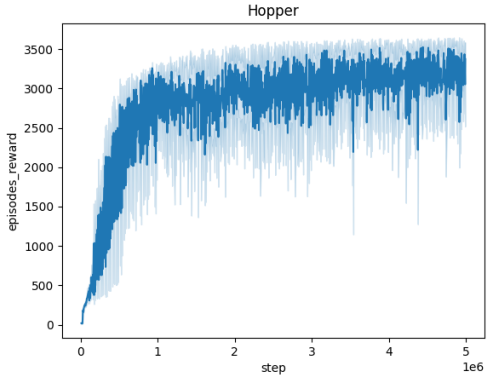
Figure A35.28: Averaged training curves for the RARL method over 10 seeds



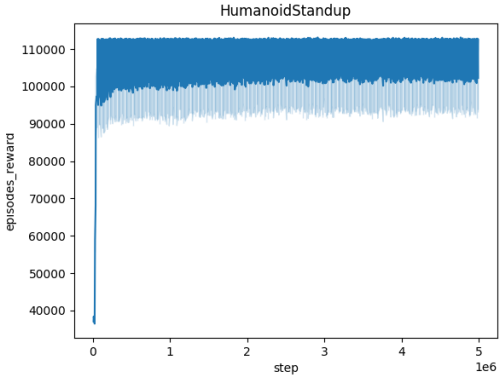
(a) Training curve on Ant with TD3



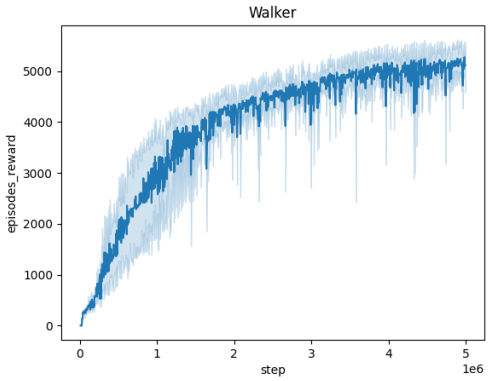
(b) Training curve on HalfCheetah with TD3



(c) Training curve on Hopper with TD3

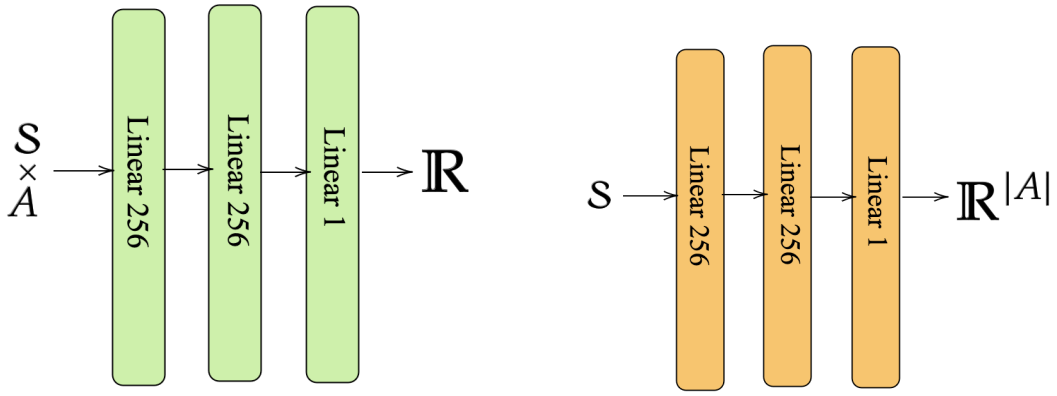


(d) Training curve on HumanoidStandup with TD3



(e) Training curve on Walker with TD3

Figure A35.29: Averaged training curves for the TD3 method over 10 seeds



(a) Critic neural network architecture

(b) Actor neural network architecture

Figure A37.30: Actor critic neural network architecture

Appendix of Chapter 8

39 Proof of the regret bound

39.1 Proof of Theorem 8.3.5

While Lambert et al. (2022) establishes quantitative bounds on the bias introduced by Algorithm 8 for the VI of the posterior. Combining this result with the one derived in Agrawal and Goyal (2013) for TS leads to sub-optimal regret bounds. It is similar to LMC-TS Xu et al. (2022) which had to make a clever adaptation of Agrawal and Goyal (2013). Similar to this work, we need here to revise the proof of Agrawal and Goyal (2013) to `VITS`. We give in this section the main steps of our proofs. Each step is based on Lemmas which are stated and proved in the next sections. First, we define the filtration $(\mathcal{F}_t)_{t\in\{0,\dots,T-1\}}$ such that for any $t \in [T]$, $\mathcal{F}_{t-1}$ is the σ-field generated by $\mathcal{H}_{t-1}$ and x_t where $\mathcal{H}_{t-1} = \{(x_s, a_s, r_s)\}_{s\leq t-1}$ is the observations up to $t-1$ and x_t is the contextual vector at step t. For some feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and for any $t \in [T]$, we denote by

$$\overset{*}{\phi}_t = \phi(x_t, a_t^\star) \ , \ \phi_t = \phi(x_t, a_t) \ ,$$

the features vector of the best arm $a_t^\star$ and the features vector of the arm a_t chosen by `VITS` at time t respectively. the difference between the best expected reward and the expected reward obtained by `VITS` is denoted by

$$\Delta_t = \overset{*}{\phi}_t \theta^\star - \phi_t^\top \theta^\star \ .$$

At each round $t \in [T]$, we consider the set of saturated arms $\mathcal{S}_t$ and unsaturated arms $\mathcal{U}_t$ defined by

$$\mathcal{S}_t = \bigcap_{a\in\mathsf{A}(x_t)} \{\Delta_t(a) > g(t) \ \|\phi(x_t, a)\|_{V_t^{-1}}\} \ , \tag{A.362}$$

and $\mathcal{U}_t = \mathsf{A}(x_t) \backslash \ \mathcal{S}_t$ where V_t^{-1} is defined in (8.15) and

$$g(t) = CR^2 d\sqrt{\log(t)\log(T)}/\lambda^{3/2} \ ,$$

for some constant $C \geq 0$ independent of d, t and T. In addition, consider the events $\mathrm{E}_t^{\mathrm{true}}$ and $\mathrm{E}_t^{\mathrm{var}}$ such that

$$\bigcap_{a\in\mathsf{A}(x_t)} \{|\phi(x_t, a)^\top \hat{\mu}_t - \phi(x_t, a)^\top \theta^\star| \leq \ g_1(t) \ \ \|\phi(x_t, a)\|_{V_t^{-1}}\} \subset \mathrm{E}_t^{\mathrm{true}}$$

$$\mathrm{E}_t^{\mathrm{var}} = \bigcap_{a\subset\mathsf{A}(x_t)} \{|\phi(x_t, a)^\top \tilde{\theta}_t - \phi(x_t, a)^\top \hat{\mu}_t| \leq \ g_2(t) \ \ \|\phi(x_t, a)\|_{V_t^{-1}}\} \ ,$$

where $\hat{\mu}_t$ is given by $\hat{\mu}_t = V^{-1} b_t$ and b_t is given in (8.15). The specific definitions of $\mathrm{E}_t^{\mathrm{true}}$, g, g_1 and g_2 are given in Section 39.3 of the supplementary. Nevertheless, by definition, it holds that $g_1(t) + g_2(t) \leq g(t)$.

1. For ease of notation, the conditional expectation $\mathbb{E}_{\pi_{1:T}\sim\mathbb{Q}_{1:T}}[\cdot]$ and probabilities $\mathbb{P}_{\pi_{1:T}\sim\mathbb{Q}_{1:T}}(\cdot)$ with respect to the σ-field $\mathcal{F}_{t-1}$ are denoted by $\mathbb{E}_t[\cdot]$ and $\mathbb{P}_t(\cdot)$ respectively. Therefore, with these notations, we have by definition of the cumulative regret:

$$\mathrm{CRegret}(\tilde{\mathbb{Q}}_{1:T}) = \sum_{t=1}^{T} \Delta_t \ .$$

We now bound for any $t \in [T]$, with high probability, $\Delta_t(a_t)$. To this end, in the next step of the proof, we show that the stochastic process $(X_t)_{t\in[T]}$ defined below is a $(\mathcal{F}_t)_{t\in[T]}$ super-martingale.

$$X_t = \sum_{s=1}^{t} Y_s$$

with

$$Y_s = \Delta_s - cg(s)\|\phi_s\|_{V_s^{-1}}/p - 2/s^2 \ ,$$

where $p \in (0,1)$ and c is a sufficiently large real number, independent of d, T and s.

2. **Showing that $(X_t)_{t\in[T]}$ is a super-martingale.** We consider the following decomposition

$$\begin{aligned}\mathbb{E}_t[\Delta_t(a_t)] &= \mathbb{E}_t[\Delta_t(a_t)\mathbb{1}_{\mathrm{E}_t^{\mathrm{true}}}] + \ \mathbb{E}_t[\Delta_t(a_t)|\bar{\mathrm{E}}_t^{\mathrm{true}}]\mathbb{P}_t(\bar{\mathrm{E}}_t^{\mathrm{true}}) \\ &\leq \mathbb{E}_t[\Delta_t(a_t)\mathbb{1}_{\mathrm{E}_t^{\mathrm{true}}}] + \mathbb{P}_t(\bar{\mathrm{E}}_t^{\mathrm{true}}) \ , \end{aligned} \tag{A.363}$$

where we used for the last inequality that $\|\theta^\star\|_2 \leq 1$ and Assumption 8.3.3. Then, since $\mathrm{E}_t^{\mathrm{true}} \in \mathcal{F}_{t-1}$, we have,

$$\begin{aligned}\mathbb{E}_t[\Delta_t(a_t)\mathbb{1}_{\mathrm{E}_t^{\mathrm{true}}}] &= \mathbb{1}_{\mathrm{E}_t^{\mathrm{true}}}\mathbb{E}_t[\Delta_t(a_t)|\mathrm{E}_t^{\mathrm{var}}]\mathbb{P}_t(\mathrm{E}_t^{\mathrm{var}}) + \mathbb{1}_{\mathrm{E}_t^{\mathrm{true}}}\mathbb{E}_t[\Delta_t(a_t)|\bar{\mathrm{E}}_t^{\mathrm{var}}]\mathbb{P}_t(\bar{\mathrm{E}}_t^{\mathrm{var}}) \\ &\leq \mathbb{1}_{\mathrm{E}_t^{\mathrm{true}}}[\mathbb{E}_t[\Delta_t(a_t)|\mathrm{E}_t^{\mathrm{var}}] + \mathbb{P}_t(\bar{\mathrm{E}}_t^{\mathrm{var}})] \end{aligned} \tag{A.364}$$

where in the last line we have used that $\Delta_t(a_t) \leq 1$ again. Denote by $\bar{a}_t = \arg\min_{a\in\mathcal{U}_t} \|\phi(x_t,a)\|_{V_t^{-1}}$ and $\overline{\phi}_t = \phi(x_t, \bar{a}_t)$. Then, given $\mathrm{E}_t^{\mathrm{true}}$ and $\mathrm{E}_t^{\mathrm{var}}$ we have

$$\begin{aligned}\Delta_t(a_t) &= \overset{*}{\phi}_t^\top\theta^\star - \phi_t^\top\theta^\star \\ &= \overset{*}{\phi}_t^\top\theta^\star - \overline{\phi}_t^\top\theta^\star + \overline{\phi}_t^\top\theta^\star - \phi_t^\top\theta^\star \\ &\overset{(a)}{\leq} g(t)\|\overline{\phi}_t\|_{V_t^{-1}} + \overline{\phi}_t^\top\theta^\star - \phi_t^\top\theta^\star \\ &\overset{(b)}{\leq} g(t)\|\overline{\phi}_t\|_{V_t^{-1}} + (\overline{\phi}_t^\top\tilde{\theta}_t + g(t)\|\overline{\phi}_t\|_{V_t^{-1}}) - (\phi_t^\top\tilde{\theta}_t - g(t)\|\phi_t\|_{V_t^{-1}}) \\ &\overset{(c)}{\leq} (2\|\overline{\phi}_t\|_{V_t^{-1}} + \|\phi_t\|_{V_t^{-1}})g(t) \end{aligned} \tag{A.365}$$

where inequality (a) is due to $\bar{a}_t \in \mathcal{U}_t$, and therefore $\Delta_t(\bar{a}_t) \leq g(t)\|\overline{\phi}_t\|_{V_t^{-1}}$, inequality (b) uses that given $\mathrm{E}_t^{\mathrm{true}}$ and $\mathrm{E}_t^{\mathrm{var}}$, for any $\phi \in \mathbb{R}^d$, $|\phi^\top\tilde{\theta}_t - \phi^\top\theta^\star| \leq g(t)\|\phi\|_{V_t^{-1}}$ since by definition $g_1(t) + g_2(t) \leq g(t)$; finally, the arm a_t maximizes the quantity $\phi(x_t,a_t)^\top\tilde{\theta}_t$, $\overline{\phi}_t^\top\tilde{\theta}_t - \phi_t^\top\tilde{\theta}_t$ is obviously negative, which implies inequality (c).

Moreover, given $\mathrm{E}_t^{\mathrm{true}}$ and $\mathrm{E}_t^{\mathrm{var}}$,

$$\begin{aligned}\mathbb{E}_t[\|\phi_t\|_{V_t^{-1}}] &= \mathbb{E}_t[\|\phi_t\|_{V_t^{-1}}|a_t \in \mathcal{U}_t]\mathbb{P}_t(a_t \in \mathcal{U}_t) + \mathbb{E}_t[\|\phi_t\|_{V_t^{-1}}|a_t \in \mathcal{S}_t]\mathbb{P}_t(a_t \in \mathcal{S}_t) \\ &\overset{(a)}{\geq} \|\overline{\phi}_t\|_{V_t^{-1}}\mathbb{P}_t(a_t \in \mathcal{U}_t) \\ &\overset{(b)}{\geq} (p - 1/t^2)\|\overline{\phi}_t\|_{V_t^{-1}} \end{aligned}$$

where (a) is due to the definition of $\overline{\phi}_t$, i.e. for any $a \in \mathcal{U}_t$, $\|\overline{\phi}_t\|_{V_t^{-1}} \leq \|\phi(x_t, a)\|_{V_t^{-1}}$, and (b) uses Lemma 39.2 with $p \in (0, 1)$. Here is one of the main differences with the proof conducted by Agrawal and Goyal (2013). Indeed, to obtain such a bound, we need to carefully dig into the convergence of the the sequence of means $\{\tilde{\mu}_{t,K_t}\}_{k\in[1,K_t]}$ and covariance matrices $\{\tilde{\Sigma}_{t,K_t}\}_{k\in[1,K_t]}$ to obtain a fine-grained analysis of the distribution of $\tilde{\mathrm{q}}_t$. Therefore, using equations (A.364) and (A.365)

$$\mathbb{1}_{\mathrm{E}_t^{\mathrm{true}}}\mathbb{E}_t[\Delta_t(a_t)] \leq \Big(\frac{2}{p - 1/t^2} + 1\Big) g(t)\mathbb{E}_t[\|\phi_t\|_{V_t^{-1}}] + \frac{1}{t^2} \leq \frac{cg(t)}{p}\mathbb{E}_t[\|\phi_t\|_{V_t^{-1}}] + \frac{1}{t^2} \,,$$

where c is a sufficiently large real number independent of the problem. Plugging this bounds in A.363, we obtain

$$\mathbb{E}_t[\Delta_t(a_t)] \leq \frac{cg(t)}{p}\mathbb{E}_t[\|\phi\|_{V_t^{-1}}] + \frac{1}{t^2} + \mathbb{P}_t(\overline{\mathrm{E}}_t^{\mathrm{true}})$$

Applying Lemma 39.1 yields

$$\mathbb{E}_t[\Delta_t(a_t)] \leq \frac{cg(t)}{p}\mathbb{E}_t[\|\phi\|_{V_t^{-1}}] + \frac{2}{t^2} \,.$$

This is another important difference with the original proof of Agrawal and Goyal (2012) which uses our precise convergence study for $\{\tilde{\mu}_{t,K_t}\}_{k\in[1,K_t]}$. Then, it follows that $(X_t)_{t\in[T]}$ is a $(\mathcal{F}_t)_{t\in[T]}$-super martingale.

3. **Concentration for $(X_t)_{t\in[T]}$.** Note that $(X_t)_{t\in[T]}$ is a super-martingale with bounded increments: for any $t \in [T]$

$$\begin{aligned}
|X_{t+1} - X_t| &= |Y_{t+1}| \\
&= |\Delta_t(a_t) - \frac{cg(t)}{p}\|\phi_t\|_{V_t^{-1}} - \frac{2}{t^2}| \\
&\overset{(a)}{\leq} |\Delta_t(a_t)(a_t) - \frac{cg(t)}{\sqrt{\lambda}p} - \frac{2}{t^2}| \\
&\leq \frac{3cg(t)}{\sqrt{\lambda}p} \,,
\end{aligned}$$

where in (a) we have used that

$$\|\phi_t\|_{V_t^{-1}} \leq \|\phi_t\|_{V_1^{-1}} \leq 1/\sqrt{\lambda}$$

and inequality (b) is due to $\Delta_t(a_t) \leq 1$, $2/t^2 \leq 2$ and $3cg(t)/(p\sqrt{\lambda}) > 2$ for an appropriate choice of the numerical constant c. Therefore, applying Azuma-Hoeffding inequality (Lemma (40.3)), with probability $1 - \delta$ it holds that

$$X_T \leq \sqrt{2\log(1/\delta)\sum_{s=1}^{T}\frac{9c^2g(s)^2}{p^2\lambda}} \leq \sqrt{18\log(1/\delta)\frac{c^2}{p^2\lambda}g(T)^2T} \,,$$

using that $g(T) \geq g(t)$.

4. **Conclusion.** The super-martingale $(X_t)_{t=1}^T$ is directly linked to the cumulative regret by

$$\begin{aligned} X_T &= \sum_{t=1}^{T} Y_t \\ &= \sum_{s=1}^{T} \Delta_t - cg(t)\|\phi_t\|_{V_t^{-1}}/p - 2/t^2 \\ &= \mathrm{CRegret}(\tilde{\mathbb{Q}}_{1:T}) - \sum_{t=1}^{T} cg(t)\|\phi_t\|_{V_t^{-1}}/p + 2/t^2 \end{aligned}$$

then taking the expectation and using the super-martingale previous argument of the proof, we obtain the following upper bound for the cumulative regret :

$$\mathrm{CRegret}(\tilde{\mathbb{Q}}_{1:T}) \leq \sum_{t=1}^{T} \frac{cg(t)}{p}\|\phi_t\|_{V_t^{-1}} + \sqrt{18\log(1/\delta)\frac{c^2}{p^2\lambda}g(T)^2T} + \frac{\pi^2}{3} \ .$$

using that $\sum_{t=1}^{+\infty} 1/t^2 \leq \pi^2/6$. As a result, applying Lemma 39.3 yields

$$\mathrm{CRegret}(\tilde{\mathbb{Q}}_{1:T}) \leq \frac{cg(T)}{p}\sqrt{2dT\log\Big(1+T/(\lambda d)\Big)} + \frac{cg(T)}{p\sqrt{\lambda}}\sqrt{18\log(1/\delta)T} + \frac{\pi^2}{3} \ .$$

Using the definition of $g(T)$ in (A.363), we get

$$\mathrm{CRegret}(\tilde{\mathbb{Q}}_{1:T}) \leq \frac{CR^2d}{\lambda^2}\log\Big(3T^3\Big)\sqrt{dT\log\left(1+T/(\lambda d)\right)\log\left(1/\delta\right)} \ ,$$

where $C \geq 0$ is a constant, independent of the problem, which completes the proof.

39.2 Hyperparameters choice and values

In this section, we define and discuss the values of the main hyperparameters.

Parameter $\boldsymbol{\eta}$: is the inverse of the temperature. The lower is η, the better is the exploration. It is fixed to

$$\eta = 4\lambda^2/(81R^2d\log\Big(3T^3\Big)) \leq 1 \tag{A.366}$$

Parameter $\boldsymbol{\lambda}$: is the inverse of the standard deviation of the prior distribution. It controls the regularization. The lower is λ, the better is the exploitation. This parameter is fixed but lower than 1.

Parameter $\boldsymbol{h_t}$: is the step size used in all Algorithms. It is fixed to

$$h_t = \lambda_{\min}(V_t)/(2\eta(\lambda_{\min}(V_t)^2 + 2\lambda_{\max}(V_t)^2)) \tag{A.367}$$

Parameter $\boldsymbol{K_t}$: is the number of gradient descent steps performed. It is fixed to

$$K_t = 1 + 2(1+2\kappa_t^2)\log\Big(2R\kappa_t d^2T^2\log^2(3T^3)\Big) \ . \tag{A.368}$$

Therefore the number of gradient descent steps is $K_t \leq \mathcal{O}(\kappa_t^2\log(dT\log(T)))$.

39.3 Useful definitions

Definition 39.1. ***(Variational approximation)*** *Recall that* $\hat{\mathrm{p}}_t(\theta) \propto \exp(-U_t(\theta))$ *is the posterior distribution. And* $\tilde{\mathrm{q}}_t$ *is the variational posterior distribution in the sense that*

$$\tilde{\mathrm{q}}_t = \arg\min_{\mathrm{p}\in\mathcal{G}} \mathrm{KL}(\mathrm{p}|\hat{\mathrm{p}}_t) \ ,$$

where $\mathcal{G}$ *is a variational family. In this paper we focus on the Gaussian variational family and we denote by* $\tilde{\mu}_t$ *and* B_t *respectively the mean and the square root covariance matrix of the variational distribution, ie,*

$$\tilde{\mathrm{q}}_t = \mathrm{N}(\tilde{\mu}_t, B_t B_t^\top) \ .$$

The values of $\tilde{\mu}_t$ *and* B_t *are obtained after running* K_t *steps of algorithm 8 or 9. Note that the sequence of means* $\{\tilde{\mu}_{t,k}\}_{k=1}^{K_t}$ *is defined recursively by*

$$\begin{aligned}\tilde{\mu}_{t,k+1} &= \tilde{\mu}_{t,k} - h_t \nabla U_t(\tilde{\theta}_{t,k}) \\ &= \tilde{\mu}_{t,k} - h_t \eta V_t(\tilde{\theta}_{t,k} - \hat{\mu}_t)\end{aligned}$$

where $\tilde{\theta}_{t,k} \sim \mathrm{N}(\tilde{\mu}_{t,k}, B_{t,k}^\top B_{t,k})$ *and we have used that* $\nabla U_t(\theta) = \eta(V_t\theta - b_t)$ *(see equation (8.15)). Consequently,* $\tilde{\mu}_{t,k}$ *is also Gaussian and we denote by* $\tilde{m}_{t,k}$ *and* $\tilde{W}_{t,k}$ *its mean and covariance matrix, ie,* $\tilde{\mu}_{t,k} \sim \mathrm{N}(\tilde{m}_{t,k}, \tilde{W}_{t,k})$*. Furthermore, the sequence of square root covariance matrix* $\{B_{t,k}\}_{k=1}^{K_t}$ *is defined recursively in Algorithm 8 by*

$$\begin{aligned}B_{t,k+1} &= \left\{\mathrm{I} - h_t \nabla^2(U_t(\tilde{\theta}_{t,k}))\right\} B_{t,k} + (B_{t,k}^\top)^{-1} \\ &= \{\mathrm{I} - \eta h_t V_t\} B_{t,k} + h_t (B_{t,k}^\top)^{-1}\end{aligned}$$

where we have used that $\nabla^2(U_t(\theta)) = \eta V_t$ *for the linear bandit case (see (8.15)). Let denote by* $\tilde{\Sigma}_{t,k} = B_{t,k}B_{t,k}^\top$ *the covariance of the variational posterior* $\tilde{\mathrm{q}}_{t,k}$*. For ease of notation we denote by* $A_t = \mathrm{I} - \eta h_t V_t$*, it follows that*

$$\tilde{\Sigma}_{t,k+1} = A_t \tilde{\Sigma}_{t,k} A_t + 2h_t A_t + h_t^2 \tilde{\Sigma}_{t,k}^{-1}$$

If $\Lambda_{t,k} = \tilde{\Sigma}_{t,k} - 1/\eta V_t^{-1}$ *denotes the difference between the covariance matrix of the varational posterior and the true posterior, therefore it holds that*

$$\begin{aligned}\Lambda_{t,k+1} &= A_t \tilde{\Sigma}_{t,k} A_t + 2h_t A_t + h_t^2 \tilde{\Sigma}_{t,k}^{-1} - 1/\eta V_t^{-1} \\ &= A_t \Lambda_{t,k} A_t + 2h_t A_t - 2h_t \mathrm{I} + \eta h_t^2 V_t + h_t^2 \tilde{\Sigma}_{t,k}^{-1} \\ &= A_t \Lambda_{t,k} A_t - \eta h_t^2 V_t + h_t^2 \tilde{\Sigma}_{t,k}^{-1} \\ &= A_t \Lambda_{t,k} A_t - h_t^2 \eta V_t \Lambda_{t,k} \tilde{\Sigma}_{t,k}^{-1}\end{aligned}$$

In the case of VITS − II, the sequence of square root covariance matrix $\{B_{t,k}\}_{k\le K_t}$ *and the sequence of inverse square root covariance matrix* $\{C_{t,k}\}_{k\le K_t}$ *are defined recursively in Algorithm 9 by*

$$\begin{aligned}C_{t,k+1} &= C_{t,k}\{\mathrm{I} - h_t(C_{t,k}^\top C_{t,k} - \nabla^2 U_t(\tilde{\theta}_{t,k}))\} \\ B_{t,k+1} &= (\mathrm{I} - h_t \nabla^2 U_t(\tilde{\theta}_{t,k}))B_{t,k} + h_t C_{t,k}^\top \ .\end{aligned}$$

Nevertheless, we will show that this approximation does not impact the cumulative regret bound. Note that all sequences are deterministic in the specific setting of linear bandit, because the Hessian of $\nabla^2 U_t(\tilde{\theta})$ *does not depend on* $\tilde{\theta}$*. In VITS-II, we obtain the following form for* $\Lambda_{t,k}$*, see Lemma* **??**.

$$\Lambda_{t,k+1} = A_t B_{t,k}^2 A_t + 2h_t A_t (B_{t,k} C_{t,k} + C_{t,k}^T B_{t,k}^T) + h_t^2 C_{t,k}^2 - 1/\eta V_t^{-1}$$

Definition 39.2. ***(Concentration events)***

The main challenge for the proof of Theorem 8.3.5, is to control the probability of the following events: for any $t \in [T]$ we define

- $\widehat{\mathrm{E}_t^{true}} = \Big\{ \text{for any } a \in \mathsf{A}(x_t) : \ |\phi(x_t,a)^\top \hat{\mu}_t - \phi(x_t,a)^\top \theta^\star| \leq \ g_1(t) \ \ \|\phi(x_t,a)\|_{V_t^{-1}} \Big\}$
- $\mathrm{E}_t^{true} = \widehat{\mathrm{E}_t^{true}} \bigcap \Big\{ |\xi_t| < R\sqrt{1+\log 3t^2} \Big\} \bigcap \Big\{ \|\tilde{W}_{t,K_t}^{-1/2}(\tilde{\mu}_{t,K_t} - \tilde{m}_{t,K_t})\| \leq \sqrt{4d \log 3t^3} \Big\}$
- $\mathrm{E}_t^{var} = \Big\{ \text{for any } a \in \mathsf{A}(x_t) : \ |\phi(x_t,a)^\top \tilde{\theta}_t - \phi(x_t,a)^\top \hat{\mu}_t| \leq \ g_2(t) \ \ \|\phi(x_t,a)\|_{V_t^{-1}} \Big\} \ ,$

where $g_1(t) = R\sqrt{d\log(3t^3)} + \sqrt{\lambda}$ and $g_2(t) = 10\sqrt{d\log(3t^3)/(\eta\lambda)}$ and ξ_t is the R-sub Gaussian noise of the reward definition defined by the relation

$$r_t = \phi_t^\top \theta^\star + \xi_t \ . \tag{A.369}$$

The first event $\widehat{\mathrm{E}_t^{true}}$ controls the concentration of $\phi(x_t,a)^\top \hat{\mu}_t$ around its mean. Similarly, event E_t^{var} controls the concentration of $\phi(x_t,a)^\top \tilde{\theta}_t$ around its mean. Note that compared to Agrawal and Goyal (2013), in our case, it is important to include within E_t^{true}, the concentration of the distributions ξ_t and $\tilde{W}_{t,K_t}^{-1/2}(\tilde{\mu}_{t,K_t} - \tilde{m}_{t,K_t})$. Consequently, conditionally on E_t^{true} and E_t^{var} it holds that: for any $a \in \mathsf{A}(x_t)$

$$\begin{aligned}
|\phi(x_t,a)^\top \tilde{\theta}_t - \phi(x_t,a)^\top \theta^\star| &\leq \left(R\sqrt{d\log(3t^3)} + \sqrt{\lambda} + 10\sqrt{d\log(3t^3)/(\eta\lambda)} \right) \|\phi(x_t,a)\|_{V_t^{-1}} \\
&\leq 12R\sqrt{d\log(3t^3)/(\eta\lambda)} \ \|\phi(x_t,a)\|_{V_t^{-1}} \\
&\overset{(a)}{=} \frac{108dR^2}{\lambda^{3/2}} \sqrt{\log(3t^3)\log(3T^3)} \|\phi(x_t,a)\|_{V_t^{-1}} \\
&:= g(t)\|\phi(x_t,a)\|_{V_t^{-1}} \ ,
\end{aligned} \tag{A.370}$$

where in (a), we have used that $\eta = 4\lambda/\Big(81R^2 d\log(3T^3)\Big)$ and in the last inequality we have used that $g(t) = CR^2 d\sqrt{\log(t)\log(T)}/\lambda^{3/2}$.

39.4 Main lemmas

Lemma 39.1. ***(Concentration lemma for $\hat{\mu}_t$)***

Recall the definition of the event E_t^{true} in (39.2). Therefore, for any $t \in [T]$, it holds that

$$\mathbb{P}(\mathrm{E}_t^{true}) \geq 1 - \frac{1}{t^2} \tag{A.371}$$

This lemma shows that the mean of the posterior distribution $\hat{\mu}_t$ is concentrated around the true parameter $\theta^\star$ with high probability.

Proof. Firstly, we apply Lemma 40.4, with $m_t = \phi_t/\sqrt{\lambda} = \phi(x_t,a_t)/\sqrt{\lambda}$ and $\epsilon_t = (r_{a_t}(t) - \phi_t^\top \theta^\star)/\sqrt{\lambda}$, where $r_{a_t}(t)$ is sampled from the R-sub-Gaussian reward distribution of mean $\phi_t^\top \theta^\star$. Let's define the filtration $\mathcal{F}'_t = \{a_{\tau+1}, m_{\tau+1}, \epsilon_\tau\}_{\tau \leq t}$. By the definition of $\mathcal{F}'_t$, m_t is $\mathcal{F}'_{t-1}$-measurable. Moreover, ϵ_t is conditionally $R/\sqrt{\lambda}$-sub-Gaussian due to Assumption 8.3.1 and is a martingale difference process because $\mathbb{E}[\epsilon_t|\mathcal{F}'_{t-1}] = 0$. If we denote by

$$M_t = \mathrm{I}_d + 1/\lambda \sum_{\tau=1}^{t} m_\tau m_\tau^\tau = 1/\lambda V_{t+1} \ ,$$

and

$$\zeta_t = \sum_{\tau=1}^{t} m_\tau \epsilon_\tau \ ,$$

Then, Lemma 40.4 shows that $\|\zeta_t\|_{M_t^{-1}} \leq R/\sqrt{\lambda}\sqrt{d\log\left(\frac{t+1}{\delta'}\right)}$ with probability at least $1-\delta'$. Moreover, note that

$$\begin{aligned} M_{t-1}^{-1}(\zeta_{t-1} - \theta^\star) &= M_t^{-1}(1/\lambda b_t - 1/\lambda \sum_{\tau=1}^{t-1} \phi_\tau \phi_\tau^\top \theta^\star - \theta^\star) \\ &= M_{t-1}^{-1}(1/\lambda b_t - M_{t-1}\theta^\star) \\ &= \hat{\mu}_t - \theta^\star \ . \end{aligned}$$

Note that $||\theta^\star||_{M_{t-1}^{-1}} = \|\theta^\star M_{t-1}^{-1/2}\|_2 \leq \|\theta^\star\|_2 \|M_{t-1}^{-1/2}\|_2 \leq \|\theta^\star\|_2$, where the last inequality is due to Assumption 8.3.2. Then, for any arm $a \in \mathsf{A}(x_t)$ we have

$$\begin{aligned} |\phi(x_t,a)^\top \hat{\mu}_t - \phi(x_t,a)^\top \theta^\star| &= |\phi(x_t,a) M_{t-1}^{-1}(\xi_{t-1} - \theta^\star)| \\ &\leq \|\phi(x_t,a)\|_{M_{t-1}^{-1}} \|\xi_{t-1} - \theta^\star\|_{M_{t-1}^{-1}} \\ &\leq \|\phi(x_t,a)\|_{M_{t-1}^{-1}} (\|\xi_{t-1}\|_{M_{t-1}^{-1}} + \|\theta^\star\|_{M_{t-1}^{-1}}) \\ &\leq \sqrt{\lambda}\left(R/\sqrt{\lambda}\sqrt{d\log\left(\frac{t}{\delta'}\right)} + 1\right) \|\phi(x_t,a)\|_{V_t^{-1}} \\ &= \sqrt{\lambda}\left(R/\sqrt{\lambda}\sqrt{d\log(3t^3)} + 1\right) \|\phi(x_t,a)\|_{V_t^{-1}} \\ &= \left(R\sqrt{d\log(3t^3)} + \sqrt{\lambda}\right) \|\phi(x_t,a)\|_{V_t^{-1}} \\ &:= g_1(t) \|\phi(x_t,a)\|_{V_t^{-1}} \ . \end{aligned}$$

This inequality holds with probability at least $\delta' = 1/(3t^2)$.

Moreover, recall the definition of the R-subGaussian noise of the reward definition in section 39.2

$$r_t = \phi_t^\top \theta^\star + \xi_t$$

Then it holds that $\mathbb{P}(|\xi_t| > x) \leq \exp(1 - x^2/R^2)$. It follows that $\mathbb{P}(|\xi_t| \leq R\sqrt{1+\log 3t^2}) \geq 1 - 1/(3t^2)$, for any $t \leq 1$. Finally, recall the definition of $\tilde{W}_{t,k}$, $\tilde{\mu}_{t,k}$ and $\tilde{m}_{t,k}$ in section 39.1. Consequently, the term $\tilde{W}_{t,K_t}^{-1/2}(\tilde{\mu}_{t,K_t} - \tilde{m}_{t,K_t})$ is gaussian with mean 0 and an identity covariance matrix. Therefore, it holds that

$$\mathbb{P}\left(\|\tilde{W}_{t,K_t}^{-1/2}(\tilde{\mu}_{t,K_t} - \tilde{m}_{t,K_t})\| \leq \sqrt{4d\log 3t^3}\right) \geq 1 - 1/(3t^2) \tag{A.372}$$

Consequently, we have

$$\mathbb{P}\left(\widehat{\mathrm{E}_t^{\mathrm{true}}} \bigcap \left\{|\xi_i| < R\sqrt{1+\log 3t^2}\right\} \bigcap \left\{\|\tilde{W}_{t,K_t}^{-1/2}(\tilde{\mu}_{t,K_t} - \tilde{m}_{t,K_t})\| \leq \sqrt{4d\log 3t^3}\right\}\right) \geq 1 - \frac{1}{t^2} \ ,$$

where $\widehat{\mathrm{E}_t^{\mathrm{true}}}$ is defined is 39.2

□

Lemma 39.2. ***Probability of playing an unsaturated arm***

Given E_t^{true} *defined in section* (39.2)*, the conditional probability of playing an unsaturated arm is strictly positive and is lower bounded as*

$$\mathbb{1}_{\mathrm{E}_t^{true}}\mathbb{P}_t(a_t \in \mathcal{U}_t) := \mathbb{P}(a_t \in \mathcal{U}_t|\mathcal{F}_{t-1}) \geq \mathbb{1}_{\mathrm{E}_t^{true}}(p - 1/t^2) \;, \tag{A.373}$$

where $p = 1/\sqrt{2\pi \mathrm{e}}$ *and* $\mathcal{U}_t$ *is defined in* (A.362)*.*

Proof. If we suppose that $\forall a \in \mathcal{S}_t, \;\; \phi(x_t, a_t^\star)^\top \tilde{\theta}_t \geq \phi(x_t, a)^\top \tilde{\theta}_t$, then $a_t \in \mathcal{U}_t$. Indeed, The optimal arm $a_t^\star$ is obviously in the unsaturated arm set $(\mathcal{U}_t)$ and $\phi(x_t, a_t)^\top \tilde{\theta}_t \geq \phi(x_t, a_t^\star)^\top \tilde{\theta}_t$ by construction of the algorithm. Hence we have

$$\mathbb{P}(a_t \in \mathcal{U}_t|\mathcal{F}_{t-1}) \geq \mathbb{P}(\overset{*}{\phi}{}^\top \tilde{\theta}_t \geq \phi(x_t, a)^\top \tilde{\theta}_t, \forall a \in \mathcal{S}_t|\mathcal{F}_{t-1})$$

Subsequently, given events $\mathrm{E}_t^{\text{true}}$ and $\mathrm{E}_t^{\text{var}}$ we have

$$\left\{\overset{*}{\phi}{}^\top \tilde{\theta}_t \geq \phi(x_t, a)^\top \tilde{\theta}_t, \forall a \in \mathcal{S}_t\right\} \supset \left\{\overset{*}{\phi}{}^\top \tilde{\theta}_t \geq \overset{*}{\phi}{}^\top \theta^\star\right\} \;.$$

Indeed, for any $a \in \mathcal{S}_t$,

$$\begin{aligned}\phi(x_t, a)^\top \tilde{\theta}_t &\overset{(a)}{\leq} \phi(x_t, a)^\top \theta^\star + g(t)\|\phi(x_t, a)\|_{\hat{\Sigma}_t} \\ &\overset{(b)}{\leq} \overset{*}{\phi}{}^\top \theta^\star \;,\end{aligned}$$

where (a) uses that $\mathrm{E}_t^{\text{true}}$ and $\mathrm{E}_t^{\text{var}}$ hold. And in inequality (b) we have used that $a \in \mathcal{S}_t$, ie, $\overset{*}{\phi}{}_t^\top \theta^\star - \phi(x_t, a)^\top \theta^\star := \Delta_t(a) > g(t)\|\phi(x_t, a)\|_{\hat{\Sigma}_t}$.

Consequently,

$$\begin{aligned}\mathbb{P}(\overset{*}{\phi}{}^\top \tilde{\theta}_t \geq \overset{*}{\phi}{}^\top \theta^\star|\mathcal{F}_{t-1}) &= \mathbb{P}(\overset{*}{\phi}{}^\top \tilde{\theta}_t \geq \overset{*}{\phi}{}^\top \theta^\star|\mathcal{F}_{t-1}, \mathrm{E}_t^{\text{var}})\mathbb{P}(\mathrm{E}_t^{\text{var}}) + \mathbb{P}(\overset{*}{\phi}{}^\top \tilde{\theta}_t \geq \overset{*}{\phi}{}^\top \theta^\star|\mathcal{F}_{t-1}, \overline{\mathrm{E}_t^{\text{var}}})\mathbb{P}(\overline{\mathrm{E}_t^{\text{var}}}) \\ &\leq \mathbb{P}(\overset{*}{\phi}{}^\top \tilde{\theta}_t \geq \phi(x_t, a)^\top \tilde{\theta}_t, \forall a \in \mathcal{S}_t|\mathcal{F}_{t-1}) + \mathbb{P}(\overline{\mathrm{E}_t^{\text{var}}})\end{aligned}$$

Therefore,

$$\begin{aligned}\mathbb{P}(a_t \in \mathcal{U}_t|\mathcal{F}_{t-1}) &\geq \mathbb{P}(\overset{*}{\phi}{}_t^\top \tilde{\theta}_t \geq \overset{*}{\phi}{}_t^\star \theta^\star|\mathcal{F}_{t-1}) - \mathbb{P}(\overline{\mathrm{E}_t^{\text{var}}}) \\ &\geq p - \frac{1}{t^2} \;,\end{aligned}$$

where the last inequality is due to Lemma 40.2 and Lemma 40.1 with $p = 1/(2\sqrt{2\pi e})$. □

Lemma 39.3. ***(Upper bound of*** $\sum_{t=1}^T \|\phi_t\|_{\hat{\Sigma}_t}$***)*** *The following lemma we will be useful in the derivation of the regret bound later in the proof.*

$$\sum_{t=1}^T \|\phi_t\|_{V_t^{-1}} \leq \sqrt{2dT \log\left(1 + \frac{T}{\lambda d}\right)}$$

Proof. Recall the relation between the 1-norm and 2-norm for a d-dimensional vector, ie, $\|\cdot\|_1 \leq \sqrt{d}\|\cdot\|_2$. Hence, it follows that

$$\sum_{t=1}^T \|\phi_t\|_{V_t^{-1}} \leq \sqrt{T \sum_{t=1}^T \|\phi_t\|^2_{V_t^{-1}}}$$

Firs, recall the definition of $V_t = \lambda \mathrm{I} \sum_{s=1}^{t-1} \phi_s \phi_s^\top$ in (8.15). Therefore, we apply Lemma 11 and Lemma 10 of Abbasi-Yadkori et al. (2011), then we have

$$\begin{aligned}\sum_{t=1}^{T} \|\phi_t\|^2_{V_t^{-1}} &\leq 2 \log \frac{\det V_t}{\det \lambda \mathrm{I}} \\ &\leq 2 \log \frac{(\lambda + T/d)^d}{\lambda^d} \\ &= 2d \log \left(1 + \frac{T}{\lambda d}\right) .\end{aligned}$$

Consequently,

$$\sum_{t=1}^{T} \|\phi_t\|_{V_t^{-1}} \leq \sqrt{2dT \log \left(1 + \frac{T}{\lambda d}\right)}$$

□

39.5 Technical Lemmas

39.5.1 Upper bound of variational mean concentration term

In this section the objective is to bound the mean variational concentration term, ie, $|\phi^\top(\tilde{m}_{t,k} - \hat{\mu})|$.

Lemma 39.4. *Given E_t^{true} defined in section (39.2), the expected mean of the variational posterior at time step t after K_t steps of gradient descent $\tilde{m}_{t,K_t}$, defined in section (39.1), is equal to:*

$$\tilde{m}_{t,K_t} = \sum_{j=1}^{t-1} \prod_{i=j}^{t-1} A_i^{K_i - 1} (\hat{\mu}_j - \hat{\mu}_{j+1}) + \hat{\mu}_t \tag{A.374}$$

where $A_i = \mathrm{I} - \eta h_i V_i$.

Proof. Recall the definitions of $\tilde{\mu}_{t,k}$ and $\tilde{m}_{t,k}$ in section 39.1. Moreover, this section also presents the sequence $\{\tilde{\mu}_{t,k}\}_{k=1}^{K_t}$ defined recursively in Algorithm 7 by:

$$\tilde{\mu}_{t,k+1} = \tilde{\mu}_{t,k} - h_t \eta V_t (\tilde{\theta}_{t,k} - \hat{\mu}_t) .$$

Note that $\{\tilde{\mu}_{t,k}\}_{k=1}^{K_t}$ is a sequence of Gaussian samples with mean and covariance matrix $\tilde{m}_{t,k}$ and $\tilde{W}_{t,k+1}$ respectively (see (39.1)). Then, we have,

$$\begin{aligned}\tilde{m}_{t,k+1} &= \mathbb{E}[\tilde{\mu}_{t,k+1}] \\ &= \tilde{m}_{t,k} - \eta h_t V_t (\tilde{m}_{t,k} - \hat{\mu}_t) \\ &= (\mathrm{I} - h_t \eta V_t) \tilde{m}_{t,k} + \eta h_t V_t \hat{\mu}_t\end{aligned}$$

Now, we recognise an arithmetico-geometric sequence, therefore the solution is:

$$\tilde{m}_{t,k} = (\mathrm{I} - h_t \eta V_t)^{k-1} (\tilde{m}_{t,1} - \hat{\mu}_t) + \hat{\mu}_t$$

Moreover, in the algorithm we use that $\tilde{\mu}_{t,1} = \tilde{\mu}_{t-1,K_{t-1}}$, which implies that $\tilde{m}_{t,1} = \tilde{m}_{t-1,k_{t-1}}$ and $W_{t,1} = W_{t-1,K_{t-1}}$. Hence, we have

$$\tilde{m}_{t,K_t} = \prod_{i=1}^{t} (\mathrm{I} - \eta h_i V_i)^{K_i - 1} (\tilde{m}_{1,1} - \hat{\mu}_1) + \sum_{j=1}^{t-1} \prod_{i=j+1}^{t} (\mathrm{I} - \eta h_i V_i)^{K_i - 1} (\hat{\mu}_j - \hat{\mu}_{j+1}) + \hat{\mu}_t \tag{A.375}$$

Moreover, the mean of the variational posterior is initialized at $\tilde{\mu}_{1,1} = 0_d$, then the expected mean of the variational posterior $\tilde{m}_1 = \hat{\mu}_1 = 0_d$. Therefore the first term of (A.375) is null. □

Lemma 39.5. *Given* E_t^{true}*, for any* $\phi \in \mathbb{R}^d$*, it holds that*

$$|\phi(\tilde{m}_{t,K_t} - \hat{\mu}_t)| \leq \sum_{j=1}^{t-1}\prod_{i=j}^{t-1}(1-\eta h \lambda_{min}(V_i))^{K_i-1}\|\phi\|_{V_j^{-1}}\|\phi_j\|_{V_j^{-1}}\left(g_1(t)/\sqrt{\lambda} + R\sqrt{1+\log(3t^2)}\right)$$

where $\tilde{m}_{t,K_t}$ *is the expected mean of the variational posterior at time step* t *after* K_t *steps of gradient descent, ie,* $\tilde{m}_{t,K_t} = \mathbb{E}[\tilde{\mu}_{t,K_t}]$*, (see section 39.1). Recall that* $g_1(t) = R\sqrt{d\log(3t^3)} + \sqrt{\lambda}$ *(see section: 39.2).*

Proof. Lemma 39.4 gives us that $\tilde{m}_{t,K_t} = \sum_{j=1}^{t-1}\prod_{i=j}^{t-1} A_i^{K_i-1}(\hat{\mu}_j - \hat{\mu}_{j+1}) + \hat{\mu}_t$ where $A_i = \mathrm{I} - \eta h_i V_i$. Then, for any $\phi \in \mathbb{R}^d$, the term we want to upper bound is:

$$|\phi^\top(\tilde{m}_{t,K_t} - \hat{\mu}_t)| \leq \sum_{j=1}^{t-1}|\phi^\top \prod_{i=j}^{t-1} A_i^{K_i-1}(\hat{\mu}_j - \hat{\mu}_{j+1})|\,, \tag{A.376}$$

We can notice that the previous term only depends on the difference between the mean posterior at time j and the one at time j+1, which can be upper bounded. Recall the different relations between V_j, b_j, r_j, ϕ_j and $\hat{\Sigma}_j$ in the linear bandit setting (see equation (8.15)): $V_{j+1} = V_j + \phi_j\phi_j^\top$, $b_{j+1} = b_j + r_j\phi_j$ and $\hat{\mu}_j = V_j^{-1}b_j$, then by Sherman–Morrison formula we have:

$$V_{j+1}^{-1} = (V_j + \phi_j\phi_j^\top)^{-1} = V_j^{-1} - \frac{V_j^{-1}\phi_j\phi_j^\top V_j^{-1}}{1+\phi_j^\top V_j^{-1}\phi_j} \tag{A.377}$$

The difference between the mean posterior at time $j+1$ and the one at time j becomes:

$$\begin{aligned}
\hat{\mu}_{j+1} - \hat{\mu}_j &= V_{j+1}^{-1}b_{j+1} - V_j^{-1}b_j \\
&= (V_j^{-1} - \frac{V_j^{-1}\phi_j\phi_j^\top V_j^{-1}}{1+\phi_j^\top V_j^{-1}\phi_j})(b_j + r_j\phi_j) - V_j^{-1}b_j \\
&= r_j V_j^{-1}\phi_j - \frac{V_j^{-1}\phi_j\phi_j^\top V_j^{-1}}{1+\phi_j^\top V_j^{-1}\phi_j}(b_j + r_j\phi_j) \\
&= \frac{V_j^{-1}\phi_j}{1+\phi_j^\top V_j^{-1}\phi_j}\left\{-\phi_j^\top\hat{\mu}_j - r_j\phi_j^\top V_j^{-1}\phi_j + r_j(1+\phi_j^\top V_j^{-1}\phi_j^\top)\right\} \\
&= \frac{V_j^{-1}\phi_j(r_j - \phi_j^\top\hat{\mu}_j)}{1+\phi_j^\top V_j^{-1}\phi_j} \\
&\overset{(a)}{=} \frac{V_j^{-1}\phi_j(\phi_j^\top\theta^\star + \xi_j - \phi_j^\top\hat{\mu}_j)}{1+\phi_j^\top V_j^{-1}\phi_j} \\
&\overset{(b)}{\leq} V_j^{-1}\phi_j(\phi_j^\top\theta^\star + \xi_j - \phi_j^\top\hat{\mu}_j)
\end{aligned} \tag{A.378}$$

where in (a) we have used that $r_j = \phi_j^\top\theta^\star + \xi_j$ with ξ_j is sampled from a R-Subgaussian distribution. Inequality (b) is due to $\phi_j^\top V_j^{-1}\phi_i = \|\phi_j\|^2_{V_j^{-1}} > 0$.

Subsequently, combining equations (A.376) and (A.378), we obtain the following upper bound

$$
\begin{aligned}
|\phi(\tilde{m}_{t,K_t} - \hat{\mu}_t)| &\leq \sum_{j=1}^{t-1}\prod_{i=j}^{t-1} |\phi^\top A_i^{K_i-1}(\hat{\mu}_j - \hat{\mu}_{j+1})| \\
&\overset{(a)}{\leq} \sum_{j=1}^{t-1}\prod_{i=j}^{t-1} \left|\phi^\top A_i^{K_i-1} V_j^{-1}\phi_j(\phi_j^\top\theta^\star + \xi_j - \phi_j^\top\hat{\mu}_j)\right| \\
&\overset{(b)}{\leq} \sum_{j=1}^{t-1}\prod_{i=j}^{t-1} (1-\eta h\lambda_{\min}(V_i))^{K_i-1}\phi^\top V_j^{-1/2}V_j^{-1/2}\phi_j|(\phi_j^\top\theta^\star + \xi_j - \phi_j^\top\hat{\mu}_j)| \\
&\overset{(c)}{\leq} \sum_{j=1}^{t-1}\prod_{i=j}^{t-1} (1-\eta h\lambda_{\min}(V_i))^{K_i-1}\|\phi\|_{V_j^{-1}}\|\phi_j\|_{V_j^{-1}}|(\phi_j^\top\theta^\star + \xi_j - \phi_j^\top\hat{\mu}_j)| \\
&\overset{(d)}{\leq} \sum_{j=1}^{t-1}\prod_{i=j}^{t-1} (1-\eta h\lambda_{\min}(V_i))^{K_i-1}\|\phi\|_{V_j^{-1}}\|\phi_j\|_{V_j^{-1}}\left(g_1(t)/\sqrt{\lambda} + R\sqrt{1+\log(3t^2)}\right)
\end{aligned}
$$

In the inequality (a) we have used equation (A.378), in (b) the relation $A_i^{K_i-1} = (\mathrm{I} - \eta h_i V_i)^{K_i-1} \preceq (1-\eta h_i\lambda_{\min}(V_i))^{K_i-1}\,\mathrm{I}_d$, in (c) the definition of $\|\phi\|_{V_t^{-1}} = \sqrt{\phi^\top V_t^{-1}\phi} = \sqrt{\phi^\top V_t^{-1/2}}\sqrt{V_t^{-1/2}\phi} = \phi^\top V_t^{-1/2}$, and finally (d) is due to $|\xi_i| < R\sqrt{1+\log 3t^2}$ as $\mathrm{E}_t^{\text{true}}$ holds and $|\phi_i^\top(\theta^\star - \hat{\mu}_t)| \leq g_1(t)\left\|\phi_i^\top\right\|_{V_t^{-1}} \leq g_1(t)/\sqrt{\lambda}$ □

Lemma 39.6. *Given E_t^{true}, for any $\phi \in \mathbb{R}^d$, if the number of gradient descent of Algorithm 8 is such that for $t \geq 2$*

$$K_t \geq 1 + 2(1+2\kappa_t^2)\log\left(4R\sqrt{dT\log(3T^3)}\right),$$

then it holds that

$$|\phi(\tilde{m}_{t,K_t} - \hat{\mu}_t)| \leq \frac{2\|\phi\|_{V_t^{-1}}}{\lambda}$$

This lemma provides the upper bound for variational mean concentration term.

Proof. Firstly, we can apply Lemma 39.5, it gives us

$$|\phi(\tilde{m}_{t,K_t} - \hat{\mu}_t)| \leq \sum_{j=1}^{t-1}\prod_{i=j}^{t-1} (1-\eta h_t\lambda_{\min}(V_t))^{K_i-1}\|\phi\|_{V_j^{-1}}\|\phi_j\|_{V_j^{-1}}\left(g_1(t)/\sqrt{\lambda} + R\sqrt{1+\log(3t^2)}\right)$$

where $g_1(t) = R\sqrt{d\log(3t^3)} + \sqrt{\lambda}$. Moreover, for $t \geq 2$,

$$
\begin{aligned}
R\sqrt{1+\log 3t^2} + g_1(t)/\sqrt{\lambda} &\leq R\sqrt{\log 3t^2} + R\sqrt{d\log\left(3t^3\right)/\lambda} + R + 1 && \text{(A.379)} \\
&\leq 4R\sqrt{d\log 3t^2/\lambda} && \text{(A.380)}
\end{aligned}
$$

where we have used that $R \geq 1$ and $\lambda \leq 1$. Moreover, for any $j \in [1,t]$ we have

$$
\begin{aligned}
\|\phi\|_{V_j^{-1}} &\leq \|\phi\|_2/\sqrt{\lambda} && \text{(A.381)} \\
&\leq \lambda_{\max}(V_t)^{1/2}\|\phi\|_{V_t^{-1}}/\sqrt{\lambda} && \text{(A.382)} \\
&= \lambda_{\max}(V_t)^{1/2}\|\phi\|_{V_t^{-1}}/\lambda^{1/2} && \text{(A.383)}
\end{aligned}
$$

Let's define $\epsilon = \left(4R\sqrt{dt\log(3t^2)}\right)^{-1} \leq 1/2$ and let's take K_i such that $(1-h_t\lambda_{\min}(V_t))^{K_i-1} \leq \epsilon$, this condition will be explained later in the proof. It follows that

$$
\begin{aligned}
|\phi(\tilde{m}_{t,K_t} - \hat{\mu}_t)| &\leq \sum_{j=1}^{t-1} \prod_{i=j}^{t-1} (1 - \eta h_t \lambda_{\min}(V_t))^{K_i - 1} \|\phi\|_{V_j^{-1}} \|\phi_j\|_{V_j^{-1}} \epsilon^{-1} \\
&\overset{(a)}{\leq} \frac{\|\phi\|_{V_t^{-1}} \lambda_{\max}(V_t)^{1/2}}{\lambda} \sum_{j=1}^{t-1} \prod_{i=j}^{t-1} (1 - h_t \lambda_{\min}(V_T))^{K_i - 1} \epsilon^{-1} \\
&\overset{(b)}{\leq} \frac{\|\phi\|_{V_t^{-1}}}{\lambda} \sum_{j=1}^{t-1} \epsilon^{t-j-1} \\
&\overset{(c)}{\leq} \frac{\|\phi\|_{V_t^{-1}}}{\lambda} \times \frac{1}{1-\epsilon} \\
&\overset{(d)}{\leq} \frac{2\|\phi\|_{V_t^{-1}}}{\lambda} ,
\end{aligned}
$$

where (a) comes from equations (A.381) and (A.380). The point (b) comes that $\lambda_{\max}(V_t) \leq \sqrt{t}$ because $\lambda \leq 1$ and definition of ϵ, then (c) from the geometric series formula. Finally, in (d), we have used $\epsilon \leq 1/2$.

Now, let's focus on that condition on K_i presented previously. For any $i \in [t]$, recall the definition of the step size h_t in 39.1.

$$
h_i = \frac{\lambda_{\min}(V_i)}{2\eta(\lambda_{\min}(V_i)^2 + 2\lambda_{\max}(V_i)^2)} ,
$$

and define $\kappa_i = \lambda_{\max}(V_i)/\lambda_{\min}(V_i)$. Therefore, it holds that

$$
(1 - \eta h_i \lambda_{\min}(V_i))^{K_i - 1} = \left(1 - \frac{1}{2(1 + 2\kappa_i^2)}\right)^{K_i - 1}
$$

For any $\epsilon > 0$, we want that $(1 - h_i \lambda_{\min}(V_i))^{K_i - 1} \leq \epsilon$. Hence we deduce that

$$
K_i \geq 1 + \frac{\log(1/\epsilon)}{\log(1 - 1/(2(1 + 2\kappa_i^2)))} .
$$

Moreover, if $0 < x < 1$ then we have $-x > \log(1 - x)$, if follows that

$$
K_i \geq 1 + 2(1 + 2\kappa_i^2) \log(1/\epsilon) .
$$

We note that,

$$
\begin{aligned}
\log(1/\epsilon) &= \log\left(4R\sqrt{dt \log 3t^3}\right) \\
&\leq \log\left(4R\sqrt{dT \log 3T^3}\right) .
\end{aligned}
$$

Finally, taking $K_i \geq 1 + 2(1 + 2\kappa_i^2) \log\left(4R\sqrt{dT \log(3T^3)}\right)$, we obtain the condition

$$
(1 - \eta h_i \lambda_{\min}(V_t))^{K_i - 1} \leq \epsilon ,
$$

which concludes the proof. □

39.5.2 Control the variational covariance matrix

The objective of this section is to control the following term: $|\boldsymbol{\phi}^\top \left(\tilde{\boldsymbol{\theta}}_{\boldsymbol{t,k}} - \tilde{\boldsymbol{\mu}}_{\boldsymbol{t,k}}\right)|$. As $\tilde{\theta}_{t,k}$ is a sample from a Gaussian distribution with mean $\tilde{\mu}_{t,k}$, the previous term will be controlled using Gaussian concentration and an upper bound of the norm of the variational covariance matrix $\tilde{\Sigma}_{t,k}$. Recall the definitions of parameters $\hat{\Sigma}_t$, $B_{t,k}$, $\tilde{\theta}_{t,k}$ and $\tilde{\mu}_{t,k}$ in section 39.1.

Lemma 39.7. *For any $t \in [T]$ and $k \in [K_t]$, the following relation holds:*

$$(\mathrm{H}): \ \tilde{\Sigma}_{t,k} \succeq V_t^{-1}/(2\eta) \tag{A.384}$$

Proof. The sequence $\{\tilde{\Sigma}_{t,n}\}_{t\in[T],\, n\in[k_t]}$ is initialized by $\tilde{\Sigma}_{1,1} = \mathrm{I}/(\lambda\eta) = V_1^{-1}/\eta \succeq V_t^{-1}/(2\eta)$. Hence, (H) holds for the pair $t = 1$ and $k = 1$. Therefore, to conclude the proof, we have to show that the following transitions are true:

- for any $t \in [T]$, if (H) holds at step (t, K_t) then it stays true at step $(t+1, 1)$ (`recursion in t`),
- for any $k \in [K_t]$, if (H) holds at step (t, k) then it stays true at step $(t, k+1)$ (`recursion in k`).

Firstly, let's focus on the first implication and suppose that (H) holds at step (t, K_t). Therefore we have

$$\tilde{\Sigma}_{t+1,1} \overset{(a)}{=} \tilde{\Sigma}_{t,K_t} \overset{(b)}{\succeq} V_t^{-1}/(2\eta) \overset{(c)}{\succeq} V_{t+1}^{-1}/(2\eta)$$

where (a) comes from the initialization of the sequence $\{\tilde{\Sigma}_{t,k}\}_{k\in[k_t]}$, (b) from the hypothesis (H) at step (t, K_t). And finally, (c) is due to $V_{t+1} = V_t + \phi_t\phi_t^\top \succeq V_t$. Then we can conclude that (H) holds at step $(t+1, 1)$.

Now we focus on the second implication and we suppose that (H) holds at step (t, k). For ease of notation we denote by $Z_{t,k} := \tilde{\Sigma}_{t,k} - V_t^{-1}/(2\eta)$. Therefore using the recursive definition of $\tilde{\Sigma}_{t,k}$ given in section (39.1), we have

$$\begin{aligned}
Z_{t,k+1} &= A_t\tilde{\Sigma}_{t,k}A_t + 2h_tA_t + h_t^2\tilde{\Sigma}_{t,k}^{-1} - V_t^{-1}/(2\eta) \\
&= A_tZ_{t,k}A_t + 2h_tA_t + h_t^2\tilde{\Sigma}_{t,k}^{-1} - h_t\mathrm{I} + \eta h_t^2V_t/2 \\
&= A_tZ_{t,k}A_t + h_t\mathrm{I} - 3h_t^2\eta V_t/4 + h_t^2\tilde{\Sigma}_{t,k}^{-1}
\end{aligned}$$

where in the last inequalities we have used that $A_t = (\mathrm{I} - \eta h_t^2 V_t)$. Moreover, all terms in the previous inequality are positive semi-definite. Indeed, as (H) holds at step (t, k), we know that $Z_{t,k} \succeq 0$ and then that $A_tZ_{t,k}A_t \succeq 0$. Moreover, $\tilde{\Sigma}_{t,k} \succeq V_t^{-1}/(2\eta) \succeq 0$, so $\tilde{\Sigma}_{t,k}^{-1} \succeq 0$. Finally, recall the definition of h_t in section (39.1)

$$\begin{aligned}
h_t &\leq \frac{\lambda_{\min}(V_t)}{2\eta(\lambda_{\min}(V_t)^2 + 2\lambda_{\max}(V_t)^2)} \\
&= \frac{1/\kappa_t}{2\eta\lambda_{\max}(V_t)((1/\kappa_t)^2 + 1)} \\
&= \frac{4}{3\eta\lambda_{\max}(V_t)} \times \frac{3/\kappa_t}{8(1 + (1/\kappa_t^2))} \\
&\leq \frac{4}{3\eta\lambda_{\max}(V_t)} \ ,
\end{aligned}$$

where $\kappa_t = \lambda_{\max}(V_t)/\lambda_{\min}(V_t) \geq 1$. Consequently, the matrix $\mathrm{I} - 3\eta h_t V_t/4$ is also positive semi-definite. Subsequently, we have

$$Z_{t,k+1} \succeq 0 \ .$$

□

Lemma 39.8. *For any $\phi \in \mathbb{R}^d$, let $B_{t,k}$ the square root of the covariance matrix defined in Algorithm (8). It holds that*

$$\|B_{t,K_t}\phi\|_2 \leq 1/\sqrt{\eta}\Big(1 + \sqrt{\|V_t\|_2 C_t}\Big)\|\phi\|_{V_t^{-1}}$$
$$\|B_{t,K_t}\phi\|_2 \geq 1/\sqrt{\eta}\Big(1 - \sqrt{\|V_t\|_2 C_t}\Big)\|\phi\|_{V_t^{-1}}$$

where $C_t = 1/\lambda \sum_{j=1}^{t-1} \prod_{i=j+1}^{t} \Big(1 - \frac{3h_t\eta}{2}\lambda_{min}(V_i)\Big)^{K_i - 1}$.

Proof. Recall the recursive relation of $\Lambda_{t,k}$ defined in Section (39.1).

$$\Lambda_{t,k+1} = A_t \Lambda_{t,k} A_t - \eta h_t^2 V_t \Lambda_{t,k} \tilde{\Sigma}_{t,k}^{-1} \ ,$$

Hence, we have the following relation on the norm of $\Lambda_{t,k+1}$:

$$\begin{aligned}
\|\Lambda_{t,k+1}\|_2 &\leq \|A_t\|_2 \|\Lambda_{t,k}\|_2 \|A_t\|_2 + \eta h_t^2 \|V_t\|_2 \|\Lambda_{t,k}\|_2 \|\tilde{\Sigma}_{t,k}^{-1}\|_2 \\
&= \Big(\lambda_{\max}(A_t)^2 + \eta h_t^2 \lambda_{\max}(V_t)\lambda_{\max}(\tilde{\Sigma}_{t,k}^{-1})\Big)\|\Lambda_{t,k}\|_2 \\
&\overset{(a)}{=} \Big(1 - 2\eta h_t \lambda_{\min}(V_t) + \eta^2 h_t^2 \lambda_{\min}(V_t)^2 + \eta h_t^2 \lambda_{\max}(V_t)\lambda_{\max}(\tilde{\Sigma}_{t,k}^{-1})\Big)\|\Lambda_{t,k}\|_2 \\
&= \left(1 - \frac{3h_t\eta}{2}\lambda_{\min}(V_t) + \eta h_t\{h_t(\eta\lambda_{\min}(V_t)^2 + \lambda_{\max}(V_t)\lambda_{\max}(\tilde{\Sigma}_{t,k}^{-1})) - \lambda_{\min}(V_t)/2\}\right)\|\Lambda_{t,k}\|_2 \\
&\overset{(b)}{\leq} \left(1 - \frac{3h_t\eta}{2}\lambda_{\min}(V_t)\right)\|\Lambda_{t,k}\|_2 \ ,
\end{aligned}$$

where (a) uses that $\lambda_{\max}(A_t) = 1 - \eta h \lambda_{\min}(V_t)$. Finally, inequality (b) is due to: $\tilde{\Sigma}_{t,k} \succeq V_t^{-1}/(2\eta)$ (Lemma 39.7). Indeed it implies that

$$\begin{aligned}
h_t(\eta\lambda_{\min}(V_t)^2 + \lambda_{\max}(V_t)\lambda_{\max}(\tilde{\Sigma}_{t,k}^{-1})) &\leq h_t(\eta\lambda_{\min}(V_t)^2 + 2\eta\lambda_{\max}(V_t)^2) \\
&\leq \lambda_{\min}(V_t)/2 \ ,
\end{aligned}$$

where the inequality comes from the definition of the step size: $h_t \leq \lambda_{\min}(V_t)/\Big(2\eta(\lambda_{\min}(V_t)^2 +$

$2\lambda_{\max}(V_t)^2)\Big)$. Subsequently,

$$\begin{aligned}
\|\Lambda_{t,K_t}\|_2 &\leq \left(1-\frac{3h_t\eta}{2}\lambda_{\min}(V_t)\right)\|\Lambda_{t,K_t-1}\|_2 \\
&\leq \left(1-\frac{3h_t\eta}{2}\lambda_{\min}(V_t)\right)^{K_t-1}\|\tilde{\Sigma}_{t,1}-1/\eta V_t^{-1}\|_2 \\
&= \left(1-\frac{3h_t\eta}{2}\lambda_{\min}(V_t)\right)^{K_t-1}\|\tilde{\Sigma}_{t-1,k_{t-1}}-1/\eta V_t^{-1}\|_2 \\
&\leq \prod_{i=1}^{t-1}\left(1-\frac{3h_t\eta}{2}\lambda_{\min}(V_i)\right)^{K_i-1}\|\tilde{\Sigma}_{1,1}-1/\eta V_1^{-1}\|_2 \qquad \text{(A.385)} \\
&+ \sum_{j=1}^{t-1}\prod_{i=j+1}^{t}\left(1-\frac{3h_t\eta}{2}\lambda_{\min}(V_i)\right)^{K_i-1}\|1/\eta V_j^{-1}-1/\eta V_{j+1}^{-1}\|_2 \\
&\overset{(a)}{\leq} \frac{1}{\eta}\sum_{j=1}^{t-1}\prod_{i=j+1}^{t}\left(1-\frac{3h_t\eta}{2}\lambda_{\min}(V_i)\right)^{K_i-1}\|V_j^{-1}-V_{j+1}^{-1}\|_2 \\
&\overset{(b)}{\leq} \frac{1}{\lambda\eta}\sum_{j=1}^{t-1}\prod_{i=j+1}^{t}\left(1-\frac{3h_t\eta}{2}\lambda_{\min}(V_i)\right)^{K_i-1}, \\
&:= C_t/\eta
\end{aligned}$$

where in (a) we have used that $\tilde{\Sigma}_{1,1}=\frac{1}{\lambda\eta}\mathrm{I}=1/\eta V_1^{-1}$. Moreover $\|V_j^{-1}-V_{j+1}^{-1}\|_2=\|(V_j^{-1}\phi_j\phi_j^\top V_j^{-1})/(1-\phi_j^\top V_j^{-1}\phi_j)\|_2$ see result (A.377). It implies that $\|V_j^{-1}-V_{j+1}^{-1}\|_2\leq\|V_j^{-1}\|_2^2\leq\|V_1^{-1}\|_2^2=1/\lambda$.

Finally, for any $\phi\in\mathbb{R}^d$,

$$\begin{aligned}
\|B_{t,K_t}\phi\|_2 &= \sqrt{\phi^\top B_{t,K_t}^\top B_{t,K_t}\phi} \\
&= \sqrt{\phi^\top\tilde{\Sigma}_{t,K_t}\phi} \\
&= \sqrt{\phi^\top(\tilde{\Sigma}_{t,K_t}-1/\eta V_t^{-1})\phi+1/\eta\phi^\top V_t^{-1}\phi} \\
&\leq \|\phi\|_2\sqrt{\|\tilde{\Sigma}_{t,K_t}-1/\eta V_t^{-1}\|_2}+1/\sqrt{\eta}\|\phi\|_{V_t^{-1}}
\end{aligned}$$

where the last inequality comes from the fact that for $a,b>0$, $\sqrt{a+b}<\sqrt{a}+\sqrt{b}$, Moreover,

$$\begin{aligned}
\|\phi\|_2 &= \|\phi V_t^{-1/2}V_t^{1/2}\|_2 \\
&\leq \|\phi\|_{V_t^{-1}}\|V_t^{1/2}\|_2
\end{aligned}$$

Consequently, we have

$$\|B_{t,K_t}\phi\|_2\leq 1/\sqrt{\eta}\Big(1+\sqrt{\|V_t\|_2 C_t}\Big)\|\phi\|_{V_t^{-1}}$$

The lower bound of this lemma

$$\|B_{t,K_t}\phi\|_2\geq 1/\sqrt{\eta}\Big(1-\sqrt{\|V_t\|_2 C_t}\Big)\|\phi\|_{V_t^{-1}}$$

is obtained because

$$\begin{aligned}\|B_{t,K_t}\phi\|_2 &= \sqrt{\phi^\top B_{t,K_t}^\top B_{t,K_t}\phi} \\ &= \sqrt{\phi^\top \tilde{\Sigma}_{t,K_t}\phi} \\ &= \sqrt{\phi^\top(\tilde{\Sigma}_{t,K_t} - 1/\eta V_t^{-1})\phi + 1/\eta\phi^\top V_t^{-1}\phi} \\ &\geq -\|\phi\|_2\sqrt{\|\tilde{\Sigma}_{t,K_t} - 1/\eta V_t^{-1}\|_2} + 1/\sqrt{\eta}\|\phi\|_{V_t^{-1}} \quad \leq \|B_{t,K_t}\phi\|_2 \geq 1/\sqrt{\eta}\Big(1 - \sqrt{\|V_t\|_2 C_t}\Big)\|\phi\|_{V_t^{-1}},\end{aligned}$$

where the first inequality comes from remarkable identity $\sqrt{a} - \sqrt{b} < \sqrt{a+b}$ for $a, b > 0$.

□

Lemma 39.9. *For any $t \in [T]$ and $a \in \mathsf{A}(x_t)$, if the number of gradient descent steps of Algorithm 8 is $K_t \geq 1 + 4(1 + 2\kappa_t^2)\log(2T)/3$, then with probability at least $1 - 1/t^2$, we have*

$$|\phi(x_t, a)^\top \left(\tilde{\theta}_{t,K_t} - \tilde{\mu}_{t,K_t}\right)| \leq \sqrt{4d\log(t^3)/\eta}\Big(1 + 1/\sqrt{\lambda}\Big)\|\phi(x_t, a)\|_{V_t^{-1}}$$

Proof. For any $a \in \mathsf{A}(x_t)$, if $K_t \geq 1 + 4(1 + 2\kappa_t^2)\log(2T)/(3\eta)$, Lemma **??** gives us that

$$\begin{aligned}|\phi(x_t, a)^\top \left(\tilde{\theta}_{t,K_t} - \tilde{\mu}_{t,K_t}\right)| &\leq \|B_{t,K_t}^{-1}(\tilde{\theta}_{t,K_t} - \tilde{\mu}_{t,K_t})\|_2\|\phi(x_t, a)^\top B_{t,K_t}\|_2 \\ &\leq \|B_{t,K_t}^{-1}(\tilde{\theta}_{t,K_t} - \tilde{\mu}_{t,K_t})\|_2(1/\sqrt{\eta})\Big(1 + 1/\sqrt{\lambda}\Big)\|\phi\|_{V_t^{-1}} .\end{aligned}$$

where first inequality comes from classical matrix norm inequality and the second one is previous lemma **??**, recall that $\tilde{\theta}_{t,K_t} \sim \mathrm{N}(\tilde{\mu}_{t,K_t}, B_{t,K_t}B_{t,K_t}^\top)$, hence $B_{t,K_t}^{-1}(\tilde{\theta}_{t,K_t} - \tilde{\mu}_{t,K_t}) \sim \mathrm{N}(0, \mathrm{I}_d)$. Therefore, with probability $1 - 1/t^2$ we have

$$B_{t,K_t}^{-1}(\tilde{\theta}_{t,K_t} - \tilde{\mu}_{t,K_t}) \leq \sqrt{4d\log(t^3)} .$$

Finally, we conclude that with probability $1 - 1/t^2$, it holds that

$$|\phi(x_t, a)^\top \left(\tilde{\theta}_{t,K_t} - \tilde{\mu}_{t,K_t}\right)| \leq \sqrt{4d\log(t^3)/\eta}\left(1 + 1/\sqrt{\lambda}\right)\|\phi(x_t, a)\|_{V_t^{-1}} .$$

□

39.5.3 Concentration of the mean of the Variational posterior around its mean

In this section, the objective is the show to concentration of $\tilde{\mu}_{t,k}$ around its mean $\tilde{m}_{t,k}$ More precisely, we want an upper bound of $|\boldsymbol{\phi}^\top (\tilde{\boldsymbol{\mu}}_{\boldsymbol{t,k}} - \tilde{\boldsymbol{m}}_{\boldsymbol{t,k}})|$.

Lemma 39.10. *For any $t \in [T]$ and $k \in [K_t]$, we have the following relation*

$$\tilde{W}_{t,k+1} = (\mathrm{I} - \eta h_t V_t)\tilde{W}_{t,k}(\mathrm{I} - \eta h_t V_t)^T + \eta^2 h_t^2 V_t \mathbb{E}[\tilde{\Sigma}_{t,k}]V_t$$

where the sequence $\{\tilde{W}_{t,k}\}_{k=1}^{K_t}$ is introduced in section 39.1. (Recall : $\tilde{\mu}_{t,k} \sim \mathrm{N}(\tilde{m}_{t,k}, \tilde{W}_{t,k})$)

Proof. In this section, we focus on the covariance matrix $\tilde{W}_{t,k}$ (see definition 39.1), by definition we have

$$\begin{aligned}\tilde{W}_{t,k+1} &= \mathbb{E}[(\tilde{\mu}_{t,k+1} - \tilde{m}_{t,k+1})(\tilde{\mu}_{t,k+1} - \tilde{m}_{t,k+1})^\top] \\ &= \mathbb{E}[a_{t,k+1}a_{t,k+1}^\top] ,\end{aligned}$$

where $a_{t,k}$ is the difference between $\tilde{\mu}_{t,k}$ and its mean. For ease of notation, let's define $\Omega_{t,k} := \tilde{\theta}_{t,k} - \tilde{m}_{t,k}$, then we have

$$a_{t,k+1} = \tilde{\mu}_{t,k} - \tilde{m}_{t,k} - \eta h_t V_t(\tilde{\theta}_{t,k} - \tilde{m}_{t,k}) = \tilde{\mu}_k - \tilde{m}_{t,k} - \eta h_t V_t \Omega_{t,k}.$$

Consequently,

$$\begin{aligned} a_{t,k+1}a_{t,k+1}^\top = (\tilde{\mu}_{t,k} - \tilde{m}_{t,k})(\tilde{\mu}_{t,k} - \tilde{m}_{t,k})^\top - \eta h_t V_t \Omega_{t,k}(\tilde{\mu}_{t,k} - \tilde{m}_{t,k})^\top \\ - \eta h_t(\tilde{\mu}_{t,k} - \tilde{m}_{t,k})\Omega_{t,k}^\top V_t + \eta^2 h_t^2 V_t \Omega_{t,k}\Omega_{t,k}^\top V_t \end{aligned}$$

Moreover, note that $\tilde{\theta}_{t,k} = \tilde{\mu}_{t,k} + B_{t,k}\epsilon_{t,k}$ where $\epsilon_{t,k} \sim \mathrm{N}(0, \mathrm{I})$. Subsequently we have $\Omega_{t,k} = \tilde{\mu}_{t,k} - \tilde{m}_{t,k} + \tilde{\Sigma}_{t,k}^{1/2}\epsilon_{t,k}$. Then we have $\mathbb{E}[\Omega_{t,k}\Omega_{t,k}^\top] = \tilde{W}_{t,k} + \mathbb{E}[B_{t,k}B_{t,k}^\top]$, $\mathbb{E}[\Omega_{t,k}(\tilde{\mu}_{t,k} - \tilde{m}_{t,k})^\top] = W_{t,k}$, and $\mathbb{E}[(\tilde{\mu}_{t,k} - \tilde{m}_{t,k})\Lambda_{t,k}^\top] = \tilde{W}_{t,k}$. Finally we obtain that

$$\begin{aligned} \tilde{W}_{t,k+1} &= \mathbb{E}[a_{t,k+1}a_{t,k+1}^\top] \\ &= \tilde{W}_{t,k} - \eta h_t V_t \tilde{W}_{t,k} - \eta h_t \tilde{W}_{t,k} V_t + \eta^2 h_t^2 V_t \tilde{W}_{t,k} V_t + \eta^2 h_t^2 V_t \mathbb{E}[B_{t,k}B_{t,k}^\top] V_t \\ &= (\mathrm{I} - \eta h_t V_t)\tilde{W}_{t,k}(\mathrm{I} - \eta h_t V_t)^\top + \eta^2 h_t^2 V_t \mathbb{E}[B_{t,k}B_{t,k}^\top] V_t \ . \end{aligned}$$

□

Lemma 39.11. *Recall that $\tilde{\mu}_{t,K_t}$, the mean of the variational posterior after K_t steps of gradient descent, is a sample from the Gaussian with mean $\tilde{m}_{t,K_t}$ and covariance matrix $\tilde{W}_{t,K_t}$, ie, $\tilde{\mu}_{t,K_t} \sim \mathrm{N}(\tilde{m}_{t,K_t}, \tilde{W}_{t,K_t})$. Recall the definition of $\Lambda_{t,k} = \tilde{\Sigma}_{t,k} - 1/\eta V_t^{-1}$, and let denote by $\Gamma_{t,k} = \tilde{W}_{t,k} - J_t V_t^{-1}$, where $J_t = h_t(2\mathrm{I} - \eta h_t V_t)^{-1} V_t$.*

This Lemma shows that the 2-norm of Γ_{t,K_t} is controlled by

$$\|\Gamma_{t,K_t}\|_2 \leq \sum_{j=1}^{t} \frac{6\kappa_j h_j \|V_j\|_2}{\lambda} \sum_{r=1}^{j} \prod_{i=r}^{t} D_i^{K_i - 1} \ ,$$

where $\kappa_j = \lambda_{max}(V_j)/\lambda_{min}(V_j)$ and $D_i = \mathrm{I} - 3\eta h_i \lambda_{min}(V_i)/2$.

Proof. Lemma (39.10) gives us that

$$\tilde{W}_{t,k+1} = A_t \tilde{W}_{t,k} A_t + \eta^2 h_t^2 V_t \tilde{\Sigma}_{t,k} V_t \ ,$$

where $A_t = \mathrm{I} - \eta h_t V_t$.

Note that J_t and V_t commute, therefore we have

$$A_t J_t V_t^{-1} A_t = J_t V_t^{-1} - 2h_t \eta J_t + \eta^2 h_t^2 J_t V_t \ .$$

Consequently, by combining the two previous equations we obtain

$$\begin{aligned} \Gamma_{t,k+1} &= A_t \Gamma_{t,k} A_t - 2h_t \eta J_t + \eta^2 h_t^2 J_t V_t + \eta h_t^2 V_t + \eta^2 h_t^2 V_t \Lambda_{t,k} V_t \\ &= A_t \Gamma_{t,k} A_t + \eta^2 h_t^2 V_t \Lambda_{t,k} V_t - h_t \eta J_t(2\mathrm{I} - \eta h_t V_t) + \eta h_t^2 V_t \\ &= A_t \Gamma_{t,k} A_t + \eta^2 h_t^2 V_t \Lambda_{t,k} V_t \ . \end{aligned}$$

It follows that

$$\|\Gamma_{t,k+1}\|_2 \leq \|A_t\|_2^2 \|\Gamma_{t,k}\|_2 + \eta^2 h_t^2 \|V_t\|_2^2 \|\Lambda_{t,k}\|_2$$

Therefore, iterating over k gives us

$$\|\Gamma_{t,k+1}\|_2 \leq \|A_t\|_2^{2k}\|\Gamma_{t,k}\|_2 + \eta^2 h_t^2 \sum_{j=0}^{k-1} \|A_t\|_2^{2j}\|V_t\|_2^2\|\Lambda_{t,k-j}\|_2 \ .$$

Moreover, Equation (A.385) is used to controls the following quantity

$$\|\Lambda_{t,k}\|_2 \leq \left(1 - \frac{3\eta h_t}{2}\lambda_{\min}(V_t)\right)^{k-1} \|\Lambda_{t,1}\|_2 \ .$$

Let's denote by $D_t = 1 - 3\eta h_t \lambda_{\min}(V_t)/2$, Subsequently

$$\|\Gamma_{t,k+1}\|_2 \leq \|A_t\|_2^{2k}\|\Gamma_{t,k}\|_2 + \eta^2 h_t^2 \sum_{j=0}^{k-1} \|A_t\|_2^{2j}\|V_t\|_2^2 D_t^{k-j-1}\|\Lambda_{t,1}\|_2 \ .$$

However, $\|A_t\|_2^2 = (1 - \eta h_t \lambda_{\min}(V_t))^2 < (1 - \frac{3\eta h_t}{2}\lambda_{\min}(V_t))$, because $\eta h_t \leq 1/(4\lambda_{\min}(V_t))$. Consequently, the geometric sum has a common ratio strictly lower than 1, then it is upper bounded by:

$$\begin{aligned}
\sum_{j=0}^{k-1} \left(\frac{\|A_t\|_2^2}{(1 - \frac{3\eta h_t}{2}\lambda_{\min}(V_t))}\right)^j &\leq \sum_{j=0}^{+\infty} \left(\frac{\|A_t\|_2^2}{(1 - \frac{3\eta h_t}{2}\lambda_{\min}(V_t))}\right)^j \\
&= \frac{1 - \frac{3\eta h_t}{2}\lambda_{\min}(V_t)}{1 - \frac{3\eta h_t}{2}\lambda_{\min}(V_t) - \|A_t\|_2^2} \\
&\leq \frac{1 - \frac{3\eta h_t}{2}\lambda_{\min}(V_t)}{1/2\ \eta h_t \lambda_{\min}(V_t) - \eta^2 h_t^2 \lambda_{\min}(V_t)^2} \\
&\leq \frac{6}{\eta h_t \lambda_{\min}(V_t)} \ , \qquad \text{(A.386)}
\end{aligned}$$

where in the first inequality we have used that the ratio of the previous sum is positive. In the last inequality we have used that $\eta h_t \leq 1/(6\lambda_{\min}(V_t))$ in the denominator and we can remove the negative part of the numerator. Therefore, it holds that

$$\|\Gamma_{t,k+1}\|_2 \leq \|A_t\|_2^{2k}\|\Gamma_{t,k}\|_2 + 6\eta\kappa_t h_t D_t^{k-1}\|V_t\|_2\|\Lambda_{t,1}\|_2 \ ,$$

where the last inequality comes from (A.386) and the definition of $\kappa_t = \lambda_{\max}(V_t)/\lambda_{\min}(V_t)$. Finally, iterating over t yields to:

$$\begin{aligned}
\|\Gamma_{t,k+1}\|_2 &\leq \|A_t\|_2^{2k} \prod_{j=1}^{t-1} \|A_j\|_2^{2(K_j-1)}\|\Gamma_{1,1}\|_2 + \sum_{j=1}^{t-1} \|A_t\|_2^{2k} \prod_{i=j+1}^{t-1} \|A_i\|_2^{2(K_i-1)} \Big(6\eta\kappa_j h_j D_j^{K_j-1}\|V_j\|_2\|\Lambda_{t,1}\|_2\Big) \\
&\quad + 6\eta\kappa_t h_t D_t^{k-1}\|V_t\|_2\|\Lambda_{t,1}\|_2 \\
&\leq \sum_{j=1}^{t-1} \|A_t\|_2^{2k} \prod_{i=j+1}^{t-1} \|A_i\|_2^{2(K_i-1)} \Big(6\eta\kappa_j h_j D_j^{K_j-1}\|V_j\|_2\|\Lambda_{t,1}\|_2\Big) + 6\eta\kappa_t h_t D_t^{k-1}\|V_t\|_2\|\Lambda_{t,1}\|_2 \ ,
\end{aligned}$$

where in the last inequalities we have used that $W_{1,1}$ is initialized such that $W_{1,1} = 1/(11\eta\lambda)\mathrm{I}$ and that $J_1 V_1 = h_1(2\mathrm{I} - \eta h_1 \lambda \mathrm{I})^{-1} = 1/(11\eta\lambda)\mathrm{I}$ because $h_1 = 1/(6\eta\lambda)$. Finally, we can conclude

$$\begin{aligned}
\|\Gamma_{t,K_t}\|_2 &\leq \sum_{j=1}^{t-1} \|A_t\|_2^{2(K_t-1)} \prod_{i=j+1}^{t-1} \|A_i\|_2^{2(K_i-1)} \Big(6\eta\kappa_j h_j D_j^{K_j-1}\|V_j\|_2\|\Lambda_{t,1}\|_2\Big) + 6\eta\kappa_t h_t D_t^{K_t-1}\|V_t\|_2\|\Lambda_{t,1}\|_2 \\
&= \sum_{j=1}^{t} \prod_{i=j+1}^{t} \|A_i\|_2^{2(K_i-1)} \Big(6\eta\kappa_j h_j D_j^{K_j-1}\|V_j\|_2\|\Lambda_{t,1}\|_2\Big) \\
&\leq \sum_{j=1}^{t} \prod_{i=j}^{t} D_i^{K_i-1} \Big(6\eta\kappa_j h_j \|V_j\|_2\|\Lambda_{t,1}\|_2\Big) \ ,
\end{aligned}$$

where in the last inequality we have used that $\|A_t\|_2^2 \leq D_t$. Moreover, equation (A.385) gives us that $\|\Lambda_{j,1}\|_2 \leq 1/(\eta\lambda)\sum_{r=1}^{j}\prod_{l=r}^{j-1} D_l^{K_l-1}$. Consequently, it holds that

$$\begin{aligned}\|\Gamma_{t,K_t}\|_2 &\leq \sum_{j=1}^{t}\frac{6\kappa_j h_j\|V_j\|_2}{\lambda}\sum_{r=1}^{j}\prod_{l=r}^{j-1}D_l^{K_l-1}\prod_{i=j}^{t}D_i^{K_i-1}\\ &= \sum_{j=1}^{t}\frac{6\kappa_j h_j\|V_j\|_2}{\lambda}\sum_{r=1}^{j}\prod_{i=r}^{t}D_i^{K_i-1}\,.\end{aligned}$$

□

Lemma 39.12. *For any $t \geq 2$, given E_t^{true}, if the number of gradient descent steps is $K_t \geq 1 + 4(1+2\kappa_t^2)\log\Big(2\kappa_t d^2 T\log^2(3T^3)\Big)/3$, therefore it holds that*

$$|\phi^\top(\tilde{\mu}_{t,K_t} - \tilde{m}_{t,K_t})| \leq \left(\sqrt{\frac{3}{\eta\lambda d\log(3t^3)}} + \sqrt{4d\log(3t^3)/(11\eta)}\right)\|\phi\|_{V_t^{-1}}\,.$$

Proof. For any $\phi \in \mathbb{R}^d$,

$$|\phi^\top(\tilde{\mu}_{t,K_t} - \tilde{m}_{t,K_t})| \leq \|\phi^\top\tilde{W}_{t,K_t}^{1/2}\|_2\|\tilde{W}_{t,K_t}^{-1/2}(\tilde{\mu}_{t,K_t} - \tilde{m}_{t,K_t}))\|_2\,, \tag{A.387}$$

where $\tilde{W}_{t,K_t}^{1/2}$ is the unique symmetric square root of $\tilde{W}_{t,K_t}$. Firstly, given $\mathrm{E}_t^{\mathrm{true}}$, the term $\|\tilde{W}_{t,K_t}^{-1/2}(\tilde{\mu}_{t,K_t} - \tilde{m}_{t,K_t})\|_2 < \sqrt{4d\log(3t^3)}$.

Then, we observe that

$$\begin{aligned}\sqrt{4d\log(3t^3)}\|\tilde{W}_{t,K_t}^{1/2}\phi\|_2 &= \sqrt{4d\log(3t^3)\phi\tilde{W}_{t,K_t}\phi^\top}\\ &\leq \sqrt{4d\log(3t^3)\phi\Gamma_{t,K_t}\phi^\top} + \sqrt{4d\log(3t^3)\phi J_t V_t^{-1}\phi^\top}\,,\end{aligned} \tag{A.388}$$

where $J_t = h_t(2\mathrm{I} - \eta h_t V_t)^{-1}V_t = (2V_t^{-1}/h_t - \eta\mathrm{I})^{-1}$ and $\Gamma_{t,k} = \tilde{W}_{t,k} - J_t V_t^{-1}$.

Moreover,

$$\begin{aligned}\sqrt{\phi J_t V_t^{-1}\phi^\top} &= \|(J_t V_t^{-1})^{1/2}\phi\|_2\\ &\overset{(a)}{=} \|J_t^{1/2}V_t^{-1/2}\phi\|_2\\ &\leq \|J_t^{1/2}\|_2\|\phi\|_{V_t^{-1}}\,,\end{aligned}$$

where in inequality (a) we have used that J_t and V_t^{-1} commute.

Recall that V_t is a symmetric matrix, therefore we have $\lambda_{\min}(V_t)\mathrm{I} \preceq V_t \preceq \lambda_{\max}(V_t)\mathrm{I}$. It follows that

$$\frac{2}{h_t\lambda_{\max}(V_t)}\mathrm{I} \preceq \frac{2}{h_t}V_t^{-1} \preceq \frac{2}{h_t\lambda_{\min}(V_t)}\mathrm{I}.$$

Recall the definition of $h_t = \lambda_{\min}(V_t)/(2\eta(\lambda_{\min}(V_t)^2 + \lambda_{\max}(V_t)^2))$. Consequently, the previous relation becomes

$$\Big(\frac{4\eta(1+2\kappa_t^2)}{\kappa_t} - \eta\Big)\mathrm{I} \preceq \frac{2}{h_t}V_t^{-1} - \eta\mathrm{I} \preceq \Big(3\eta + 8\eta\kappa_t^2\Big)\mathrm{I}\,. \tag{A.389}$$

The left hand term is obviously positive, therefore it holds that

$$\begin{aligned}\|J_t^{1/2}\|_2 &= \|(\frac{2}{h_t}V_t^{-1} - \eta \mathrm{I})^{-1/2}\|_2 \\ &\leq \sqrt{\frac{\kappa_t}{\eta(4+8\kappa_t^2-\kappa_t)}} \\ &\leq \frac{1}{\sqrt{\eta(4+7\kappa_t^2)}} \\ &\leq \frac{1}{\sqrt{11\eta}}\,.\end{aligned}$$

Finally,

$$\sqrt{4d\log(3t^3)\phi J_t V_t^{-1}\phi^\top} \leq \sqrt{4d\log(3t^3)/(11\eta)}\|\phi\|_{V_t^{-1}}\,.$$

Now, we focus on the first term of equation A.388. Lemma 39.11 gives us that

$$\begin{aligned}\|\Gamma_{t,K_t}\|_2 &\leq \sum_{j=1}^{t}\frac{6\kappa_j h_j\|V_j\|_2}{\lambda}\sum_{r=1}^{j}\prod_{i=r}^{t}D_i^{K_i-1} \\ &\leq \sum_{j=1}^{t}\frac{\kappa_j}{\eta\lambda}\sum_{r=1}^{j}\prod_{i=r}^{t}D_i^{K_i-1}\,,\end{aligned}$$

where in the last inequality we have used that $h_t\|V_t\|_2 = \kappa_t/(2\eta(1+2\kappa_t)) \leq 1/(6\eta)$. For any $j \in [2,t]$, let's define $\epsilon_j = 1/(2(\kappa_j d^2 t^2 \log^2(3t^3)))$. Additionally, let's fix K_i such that $D_i^{K_i-1} \leq \epsilon_j$ (this condition will be explained later in the Lemma). Subsequently, we have

$$\begin{aligned}4d\log\Big(3t^3\Big)\|V_t\|_2/(\eta\lambda)\sum_{j=1}^{t}\kappa_j\sum_{r=1}^{j}\prod_{i=r}^{t}D_i^{K_i-1} &\leq 4d\log\Big(3t^3\Big)\|V_t\|_2/(\eta\lambda)\sum_{j=1}^{t}\kappa_j\sum_{r=1}^{j}\epsilon_j^{t-r+1} \\ &\leq \frac{2\|V_t\|_2}{t^2\eta\lambda d\log(3t^3)}\sum_{r=1}^{t}\sum_{j=r}^{t}\epsilon_j^{t-r} \\ &\overset{(a)}{\leq} \frac{2\|V_t\|_2}{t^2\eta\lambda d\log(3t^3)}\sum_{r=1}^{t}\sum_{j=r}^{t}\Big(\frac{1}{2d^2t^2\log^2(3t^3)}\Big)^{t-r} \\ &\leq \frac{2\|V_t\|_2}{t\eta\lambda d\log(3t^3)}\sum_{r=1}^{t}\frac{t-r+1}{t}\Big(\frac{1}{2d^2t^2\log^2(3t^3)}\Big)^{t-r} \\ &\overset{(b)}{\leq} \frac{2\|V_t\|_2}{t\eta\lambda d\log(3t^3)}\sum_{r=1}^{t}\Big(\frac{1}{2d^2t^2\log^2(3t^3)}\Big)^{t-r} \\ &\overset{(c)}{\leq} \frac{2\|V_t\|_2}{\eta t\lambda d\log(3t^3)}\sum_{u=0}^{t-1}\Big(\frac{1}{25}\Big)^{u} \\ &\leq \frac{3}{\eta\lambda d\log(3t^3)}\,,\end{aligned}$$

where in (a) we have used that $\epsilon_j \leq 1/(2(d^2t^2\log^2(3t^3))$. Inequality (b) is due to $t-r+1 \leq t$. The inequality (c) is obtained because $1/(2d^2t^2\log^2 3t^3) \leq 1/(4\times\log^2(8)) \leq 1/25$ and $u = t-r$. For the last inequality we have used the geometric series formula and $\|V_t\|_2 = \|\lambda\mathrm{I} + \sum_{s=1}^{t-1}\phi\phi^\top\|_2 \leq \lambda + t - 1 \leq t$, because $\lambda \leq 1$.

Consequently, as $\|\phi\|_2 \leq \|V_t^{1/2}\|_2 \|\phi\|_{V_t^{-1}}$, we obtain

$$\sqrt{4d\log(3t^3)\phi\Gamma_{t,K_t}\phi^\top} \leq \sqrt{\frac{3}{\eta\lambda d\log(3t^3)}}\|\phi\|_{V_t^{-1}} \,. \tag{A.390}$$

Moreover, the previous inequalities hold if $(1-(3/2)\eta h_i \lambda_{\min}(V_i))^{K_i-1} \leq \epsilon$, following a similar reasoning than in section 39.5.2, it follows that we need

$$K_t \geq 1 + 4(1+2\kappa_t^2)\log\Big(2\kappa_t d^2T^2\log^2(3T^3)\Big)/3$$

□

40 Concentration and anti-concentration

Lemma 40.1. ***(Concentration lemma for $\tilde{\theta}_t$)***

For any $t \in [T]$, given E_t^{true}, the following event is controlled

$$\mathbb{P}(\mathrm{E}_t^{var}|\mathcal{F}_{t-1}) \geq 1 - \frac{1}{t^2}$$

Proof. Firstly, if $t = 1$, the condition is obvious because $\mathbb{P}(\mathrm{E}_t^{\mathrm{var}}|\mathcal{F}_{t-1}) \geq 0$. For the rest of the proof, we assume that $t \geq 2$. Recall the definition of the event $\mathrm{E}_t^{\mathrm{var}}$:

$$\mathrm{E}_t^{\mathrm{var}} = \Big\{\text{for any } a \in \mathsf{A}(x_t), |\phi(x_t,a)^\top\tilde{\theta}_t - \phi(x_t,a)^\top\hat{\mu}_t| \leq g_2(t)\|\phi(x_t,a)\|_{V_t^{-1}}\Big\} \,.$$

with $g_2(t) = 10\sqrt{d\log(3t^3)/(\eta\lambda)}$.

Let $a \in \mathsf{A}(x_t)$, it holds that

$$|\phi(x_t,a)^\top(\tilde{\theta}_t - \hat{\mu}_t)| \leq |\phi(x_t,a)^\top(\tilde{\theta}_{t,K_t} - \tilde{\mu}_{t,K_t})| + |\phi(x_t,a)^\top(\tilde{\mu}_{t,K_t} - \tilde{m}_{t,K_t})| + |\phi(x_t,a)^\top(\tilde{m}_{t,K_t} - \hat{\mu})| \,,$$

where $\tilde{\theta}_t = \tilde{\theta}_{t,K_t}$ is a sample from the variational posterior distribution trained after K_t steps of Algorithm 8. $\tilde{\mu}_{t,K_t}$ and $\tilde{\Sigma}_{t,K_t}$ are, respectively, the mean and covariance matrix of the variational posterior. Moreover, $\tilde{\mu}_{t,K_t}$ is gaussian with mean $\tilde{m}_{t,K_t}$ and covariance matrix $\tilde{W}_{t,K_t}$ (see section 39.1). If the number of gradient descent steps is $K_t^{(1)} \geq 1 + 4(1+2\kappa_t^2)\log(2T)/3$, then Lemma 39.9 shows that with probability at least $1 - 1/t^2$, we have

$$\begin{aligned}|\phi(x_t,a)^\top\Big(\tilde{\theta}_{t,K_t} - \tilde{\mu}_{t,K_t}\Big)| &\leq \sqrt{4d\log(t^3)/\eta}\Big(1 + 1/\sqrt{\lambda}\Big)\|\phi(x_t,a)\|_{V_t^{-1}} \\ &\leq 4\sqrt{d\log(t^3)/(\eta\lambda)}\|\phi(x_t,a)\|_{V_t^{-1}},\end{aligned}$$

where the last inequality is due to $\lambda \leq 1$.

Similarly, Lemma 39.12 shows that for any $t \geq 2$, given $\mathrm{E}_t^{\mathrm{true}}$, if $K_t^2 \geq 1 + 4(1+2\kappa_t^2)\log\Big(2\kappa_t d^2T^2\log^2(3T^3)\Big)/3$, therefore we have

$$|\phi(x_t,a)^\top(\tilde{\mu}_{t,K_t} - \tilde{m}_{t,K_t})| \leq \left(\sqrt{\frac{3}{\eta\lambda d\log(3t^3)}} + \sqrt{4d\log(3t^3)/(11\eta)}\right)\|\phi(x_t,a)\|_{V_t^{-1}}$$

where in the last simplification we have used $\lambda \leq 1$.

Finally, Given $\mathrm{E}_t^{\text{true}}$, let's apply Lemma 39.6 with a number of gradient descent steps such $K_t^{(3)} \geq 1 + 2(1+2\kappa_t^2)\log\left(4R\sqrt{dT\log(3T^3)}\right)$, we obtain that

$$|\phi(\tilde{m}_{t,K_t} - \hat{\mu}_t)| \leq 2/\lambda \|\phi(x_t,a)\|_{V_t^{-1}}.$$

Note that $K_t = 1 + 2(1+2\kappa_t^2)\log\left(2R\kappa_t d^2T^2\log^2(3T^3)\right) \geq \max\{K_t^{(1)}, K_t^{(2)}, K_t^{(3)}\}$ (see Equation (A.368)), then with probability at least $1 - 1/t^2$ we have

$$\begin{aligned}
|\phi(x_t,a)^\top\left(\tilde{\theta}_{t,k} - \hat{\mu}_t\right)| &\leq |\phi(x_t,a)^\top\left(\tilde{\theta}_{t,k} - \tilde{\mu}_{t,k}\right)| + |\phi(x_t,a)^\top(\tilde{\mu}_{t,k} - \tilde{m}_{t,k})| + |\phi(x_t,a)^\top(\tilde{m}_{t,k} - \hat{\mu})| \\
&\leq \left(4\sqrt{d\log(t^3)/(\eta\lambda)} + \sqrt{\frac{3}{\eta\lambda d\log(3t^3)}} + \sqrt{4d\log(3t^3)/(11\eta)} + 2/\lambda\right)\|\phi(x_t,a)\|_{V_t^{-1}} \\
&\leq 10\sqrt{d\log(3t^3)/(\eta\lambda)}\|\phi(x_t,a)\|_{V_t^{-1}} \\
&\leq g_2(t)\|\phi(x_t,a)\|_{V_t^{-1}} .
\end{aligned}$$

where the last inequality holds because $t \geq 2$, $\lambda \leq 1$ and $\eta \leq 1$. □

Lemma 40.2. *(Anti-concentration lemma) Given E_t^{true}, if the number of gradient steps is $K_t = 1 + 2(1+\kappa_t^2)\log\left(2R\kappa_t d^2T^2\log^2(3T^3)\right)$ Therefore, it holds that*

$$\mathbb{P}\left(\overset{*}{\phi}_t^\top\tilde{\theta}_{t,k} > \overset{*}{\phi}_t^\top\theta^\star\right) \leq p ,$$

where $p = 1/(2\sqrt{2\pi e})$

Proof. Firstly, note that

$$\mathbb{P}\left(\overset{*}{\phi}_t^\top\tilde{\theta}_{t,K_t} > \overset{*}{\phi}_t^\top\theta^\star\right) = \mathbb{P}\left(\frac{\overset{*}{\phi}_t^\top\tilde{\theta}_{t,K_t} - \overset{*}{\phi}_t^\top\tilde{m}_{t,K_t}}{\sqrt{\overset{*}{\phi}_t^\top\tilde{\Sigma}_{t,K_t}\overset{*}{\phi}_t + \overset{*}{\phi}_t^\top\tilde{W}_{t,K_t}\overset{*}{\phi}_t}} > \frac{\overset{*}{\phi}_t^\top\theta^\star - \overset{*}{\phi}_t^\top\tilde{m}_{t,K_t}}{\sqrt{\overset{*}{\phi}_t^\top\tilde{\Sigma}_t\overset{*}{\phi}_t + \overset{*}{\phi}_t^\top\tilde{W}_{t,K_t}\overset{*}{\phi}_t}}\right) .$$

Recall that

$$\overset{*}{\phi}_t^\top\tilde{\mu}_t \sim \mathrm{N}(\overset{*}{\phi}_t^\top\tilde{m}_t,\ \overset{*}{\phi}_t\tilde{W}_{t,k}\overset{*}{\phi}_t^\top) \text{ and } \quad \overset{*}{\phi}_t^\top\tilde{\theta}_{t,K_t} \sim \mathrm{N}(\overset{*}{\phi}_t^\top\tilde{\mu}_{t,k},\ \overset{*}{\phi}_t^\top\tilde{\Sigma}_t\overset{*}{\phi}_t) .$$

Therefore, using the conditional property of Gaussian vectors, we have

$$\overset{*}{\phi}_t^\top\tilde{\theta}_t \sim \mathrm{N}(\overset{*}{\phi}_t^\top\tilde{m}_t,\ \overset{*}{\phi}_t\tilde{\Sigma}_t\overset{*}{\phi}_t^\top + \overset{*}{\phi}_t\tilde{W}_t\overset{*}{\phi}_t^\top) .$$

Consequently, we have to control the term

$$Y_t := (\overset{*}{\phi}_t^\top\theta^\star - \overset{*}{\phi}_t^\top\tilde{m}_{t,K_t})/(\sqrt{\overset{*}{\phi}_t^\top\tilde{\Sigma}_{t,K_t}\overset{*}{\phi}_{t,K_t} + \overset{*}{\phi}_t^\top\tilde{W}_{t,K_t}\overset{*}{\phi}_{t,K_t}})$$

and use the Gaussian anti-concentration lemma (Lemma 40.5). First, in this lemma, we suppose that $\mathrm{E}_t^{\text{true}}$ holds, therefore we have

$$\begin{aligned}
|\overset{*}{\phi}_t^\top(\hat{\mu}_t - \theta^\star)| &\leq g_1(t)\|\overset{*}{\phi}_t\|_{V_t^{-1}} \\
&= \left(R\sqrt{d\log(3t^3)} + \sqrt{\lambda}\right)\|\overset{*}{\phi}_t\|_{V_t^{-1}} .
\end{aligned}$$

Moreover, as the number of gradient descent, defined in section 39.2 is upper than $K_t^{(1)} = 1 + 2(1 + 2\kappa_i^2)\log\left(4R\sqrt{dT\log(3T^3)}\right)$, then Lemma 39.6 gives us that

$$|\overset{*}{\phi}_t^\top(\tilde{m}_{t,K_t} - \hat{\mu}_t)| \leq \frac{2\|\overset{*}{\phi}_t\|_{V_t^{-1}}}{\lambda}.$$

Consequently, the numerator of Y_t is upper bounded by

$$\begin{aligned}|\phi_t^{*\top}(\theta^\star - \tilde{m}_{t,K_t})| &\overset{a}{\leq} |\overset{*}{\phi}_t^\top(\theta^\star - \hat{\mu}_{t,K_t})| + |\overset{*}{\phi}_t^\top(\hat{\mu}_{t,K_t} - \tilde{m}_{t,K_t})| \\ &\overset{(b)}{\leq}\left(R\sqrt{d\log(3t^3)} + \sqrt{\lambda} + \frac{2}{\lambda}\right)\|\overset{*}{\phi}_t^\top\|_{V_t^{-1}}\end{aligned}$$

Regarding the denominator of Y_t, we need a lower bound for $\|B_{t,k}\overset{*}{\phi}_t\|_2$. Lemma 39.8 for VITS-Ior **??** for VITS-IIgives us that

$$\|B_{t,K_t}\phi\|_2 \geq 1/\sqrt{\eta}\left(1 - \sqrt{\|V_t\|_2 C_t}\right)\|\overset{*}{\phi}_t\|_{V_t^{-1}}$$

with

$$-C_t^{1/2} = -\Big(\frac{1}{\lambda}\sum_{j=1}^{t-1}\prod_{i=j+1}^{t}(1 - \frac{3h_t\eta}{2}\lambda_{\min}(V_t))^{K_i - 1}\Big)^{1/2}. \tag{A.391}$$

Finally, we find a lower bound to this quantity.

$$\begin{aligned}\|B_{t,K_t}\phi\|_2 &\geq 1/\sqrt{\eta}\left(1 - \sqrt{\|V_t\|_2 C_t}\right)\|\overset{*}{\phi}_t\|_{V_t^{-1}} \\ &\overset{(a)}{=} \frac{\|\overset{*}{\phi}_t\|_{V_t^{-1}}}{\sqrt{\eta}}\Big(1 - \sqrt{\|V_t\|_2}\Big(\frac{1}{\lambda}\sum_{j=1}^{t-1}\prod_{i=j+1}^{t}(1 - \frac{3h_t\eta}{2}\lambda_{\min}(V_t))^{K_i-1}\Big)^{1/2}\Big) \\ &\overset{(b)}{=} \frac{\|\overset{*}{\phi}_t\|_{V_t^{-1}}}{\sqrt{\eta}}\left(1 - \Big(\sum_{j=1}^{t-1}\epsilon^{t-j-1}\Big)^{1/2}\right) \\ &\overset{(c)}{=} \frac{\|\overset{*}{\phi}_t\|_{V_t^{-1}}}{\sqrt{\eta}}\left(1 - \Big(\sum_{j=0}^{t-2}\epsilon^{j}\Big)^{1/2}\right) \\ &\overset{(d)}{\geq} \frac{\|\overset{*}{\phi}_t\|_{V_t^{-1}}}{\sqrt{\eta}}\Big(1 - \frac{1}{9t^{1/4}}\Big) \\ &\overset{(e)}{\geq} \frac{\|\overset{*}{\phi}_t\|_{V_t^{-1}}}{\sqrt{\eta}}\Big(1 - \frac{1}{9}\Big) \\ &= \frac{8\|\overset{*}{\phi}_t\|_{V_t^{-1}}}{9\sqrt{\eta}}\end{aligned}$$

with (a) is A.391. Where (b) we use $\|V_t\|_2 \leq t$ and setting $\epsilon = (4t)^{-1}$, point (c) comes from a change of variable, (d) comes from the fact that for any $t \geq 1$, $\sum_{j=0}^{t-2}\epsilon^j < 1/(81\sqrt{t})$. Finally, (e) comes from that $1/t$ by can be upper bounded by 1 for any t.

Finally, regrouping the nominator and the denominator, we have the following expression for Y_t:

$$\begin{aligned} Y_t &\leq \frac{\overset{*}{\phi}_t^\top \theta^\star - \overset{*}{\phi}_t^\top \tilde{m}_{t,K_t}}{\sqrt{\overset{*}{\phi}_t^\top \tilde{\Sigma}_{t,K_t} \overset{*}{\phi}_{t,K_t} + \overset{*}{\phi}_t^\top \tilde{W}_{t,K_t} \overset{*}{\phi}_{t,K_t}}} \\ &\leq \frac{\overset{*}{\phi}_t^\top \theta^\star - \overset{*}{\phi}_t^\top \tilde{m}_{t,K_t}}{\left\| \overset{*}{\phi}_t \right\|_{\tilde{\Sigma}_{t,K_t}}} \\ &\leq \frac{R\sqrt{d\log(3t^3)} + \sqrt{\lambda} + \frac{2}{\lambda}}{8/(9\sqrt{\eta})} \\ &\leq \frac{9R\sqrt{d\log(3t^3)}\sqrt{\eta}}{2\lambda} \end{aligned}$$

Recall the definition of η in Section 39.2

$$\eta = \frac{4\lambda^2}{81R^2 d\log(3T^3)}$$

Consequently, it yields that $|Y_t| \leq 1$.

Finally, Lemma 40.5 gives us that

$$\mathbb{P}\left(\overset{*}{\phi}_t^\top \tilde{\theta}_{t,K_t} > \overset{*}{\phi}_t^\top \theta^\star\right) \geq \frac{1}{2\sqrt{2\pi e}}$$

□

40.1 Auxiliary Lemmas

Lemma 40.3. ***(Azuma-Hoeffding inequality)*** *We define $\{X_s\}_{s\in[T]}$ a super-martingale associated to the filtration $\mathcal{F}_t$. If it holds that for any $s \geq 1$, $|X_{s+1} - X_s| \leq c_{s+1}$. Then for any $\epsilon > 0$, we have*

$$\mathbb{P}(X_T - X_0 \geq \epsilon) \leq \exp\left(-\frac{\epsilon^2}{2\sum_{s=1}^T c_s^2}\right).$$

Lemma 40.4. ***(Martingale Lemma Abbasi-Yadkori et al. (2011))*** *Let $(\mathcal{F}_t)_{t\geq 0}$ be a filtration, $(m_t)_{t\geq 1}$ be an $\mathbb{R}^d$-valued stochastic process such that m_t is $(\mathcal{F}'_{t-1})$-measurable, $(\epsilon_t)_{t\geq 1}$ be a real-valued martingale difference process such that ϵ_t is $(\mathcal{F}'_t)$-measurable. For $t \geq 0$, define $\zeta_t = \sum_{\tau=1}^t m_\tau \epsilon_\tau$ and $M_t = \mathrm{I}_d + \sum_{\tau=1}^t m_\tau m_\tau^\top$, where I_d is the d-dimensional identity matrix. Assume ϵ_t is conditionally R-sub-Gaussian. Then, for any $\delta' > 0$, $t \geq 0$, with probability at least $1 - \delta'$,*

$$\|\zeta_t\|_{M_t^{-1}} \leq R\sqrt{d\log\left(\frac{t+1}{\delta'}\right)}$$

where $\|\zeta_t\|_{M_t^{-1}} = \sqrt{\zeta_t^\top M_t^{-1} \zeta_t}$

Lemma 40.5. ***(Gaussian concentration (Abramowitz and Stegun 1964))*** *Suppose Z is a Gaussian random variable $Z \sim \mathcal{N}(\mu, \sigma^2)$, where $\sigma > 0$. For $0 \leq z \leq 1$, we have*

$$\mathbb{P}(Z > \mu + z\sigma) \geq \frac{1}{\sqrt{8\pi}} e^{-\frac{z^2}{2}}, \quad \mathbb{P}(Z < \mu - z\sigma) \geq \frac{1}{\sqrt{8\pi}} e^{-\frac{z^2}{2}} \tag{A.392}$$

And for $z \geq 1$, we have

$$\frac{e^{-z^2/2}}{2z\sqrt{\pi}} \leq \mathbb{P}(|Z-\mu| > z\sigma) \leq \frac{e^{-\frac{z^2}{2}}}{z\sqrt{\pi}}$$

41 Approximation of our algorithm and complexity

In this section, the objective is to approximate the inversion of the matrix $B_{t,k}$ of Algorithm 8. Indeed, Algorithm 8, requires to compute the inversion of a $d \times d$ matrix at each step t and k, which represents a complexity of $\mathcal{O}(d^3)$. In the approximated version of Algorithm 8, we consider both the sequence of square root covariance matrix $\{B_{t,k}\}_{k=1}^{K_t}$ and the sequence of their approximations $\{C_{t,k}\}_{k=1}^{K_t}$ such that: for any $t \in [T]$ and $k \in [K_t]$

$$C_{t,k} \approx B_{t,k}^{-1} \, .$$

Recall the recursive definition of $B_{t,k}$,

$$\begin{aligned} B_{t,k+1} &= \{\mathrm{I} - h_t A_{t,k})\}\, B_{t,k} + h_t (B_{t,k}^\top)^{-1} \\ &\approx \{\mathrm{I} - h_t A_{t,k})\}\, B_{t,k} + h_t C_{t,k}^\top \, , \end{aligned} \tag{A.393}$$

where $A_{t,k} = B_{t,k}^2(\tilde{\theta}_{t,k} - \tilde{\mu}_{t,k})(\nabla U_t(\tilde{\theta}_{t,k})^\top$ if the hessian free algorithm is used or $A_{t,k} = \nabla^2 U(\tilde{\theta}_{t,k})$ otherwise. Recall that $\tilde{\theta}_{t,k} \sim \mathrm{N}(\tilde{\mu}_{t,k}, B_{t,k}B_{t,k}^\top)$. Furthermore, we can now focus on the definition of the sequence $\{C_{t,k}\}_{k=1}^{K_t}$. Firstly, we recall that

$$\begin{aligned} B_{t,k+1} &= \{\mathrm{I} - h_t A_{t,k}\}\, B_{t,k} + h_t (B_{t,k}^\top)^{-1} \\ &= \{\mathrm{I} + h_t((B_{t,k}^\top)^{-1}(B_{t,k})^{-1} - A_{t,k}\} B_{t,k}. \end{aligned}$$

Then, let's use a first order Taylor expansion of the previous equation in h_t, we obtain the approximated inverse square root covariance matrix:

$$C_{t,k+1} = C_{t,k}^{-1}\{(\mathrm{I} - h_t(C_{t,k}^\top C_{t,k} - A_{t,k})\}. \tag{A.394}$$

Note that the lower is h_t, the better is the approximation and in our case the step size h_t is decreasing with t. The approximated recursive definition of the square root covariance matrix defined in equation (A.393) and its approximated inverse defined in equation (A.394) are used to defined our the approximated version of `VITS` called `VITS - II` and is presented in Algorithm 9. Moreover, note that the updating step of Algorithm 9 uses only matrix multiplication and sampling from independent Gaussian distribution $\mathrm{N}(0, \mathrm{I})$. Therefore the global complexity of the overall algorithm is $\mathcal{O}(d^2)$.

42 Discussion on the difference between the algorithm of Urteaga and Wiggins (2018) and our algorithm VITS.

The main difference between our setting and the one of Urteaga and Wiggins (2018) is the bandit modelisation. Indeed, given a context x and an action a, in our setting, the agent receives a reward $r \sim \mathrm{R}(\cdot|x, a)$. Consequently, a parametric model R_θ is used to approximate the reward distribution and it yields to a posterior distribution $\hat{\mathrm{p}}$. In the setting of Urteaga and Wiggins (2018), the agent receives a reward $r \sim \mathrm{R}_a(\cdot|x)$. Then, it considers a set of parametric models

$\{\mathrm{R}_{\theta_a}\}_{a=1}^K$ and a set of posterior distributions: $\{\hat{\mathrm{p}}_a\}_{a=1}^K$. The setting we have used in this paper is richer as it consider the correlation between the arms distributions compared to Urteaga and Wiggins (2018) which consider that the arm distributions are independents. For example, if we consider the case of the Linear bandit. In this setting, the posterior distribution is Gaussian. With the modelisation of Urteaga and Wiggins (2018), we have for any $a \in [K]$, $\hat{\mathrm{p}}_a := \mathrm{N}(\mu_a, \Sigma_a)$, where $\mu_a \in \mathbb{R}^d$ and $\Sigma_a \in \mathcal{S}_+^d$. However, with our modelisation, $\hat{\mathrm{p}} := \mathrm{N}(\mu, \Sigma)$, where $\mu \in \mathbb{R}^{d\times K}$ and $\Sigma \in \mathcal{S}_+^{d\times K}$. We can see that the covariance matrix Σ encodes the correlations between the different arms, which is not the case of $\{\Sigma_a\}_{a=1}^K$. In addition, in our setting, we can consider any model for the mean of the reward distribution. For example we can choose $g(\theta, x, a)$ as a Neural Networks. This kind of model is unusable in the formulation of Urteaga and Wiggins (2018).

Moreover, the approximate families used in both papers are different. Indeed, we consider the family of non-degenerate Gaussian distributions, and Urteaga and Wiggins (2018) is focused on the family of mixture of mean-field Gaussian distribution. The mixture of Gaussian distribution is richer than the classic Gaussian distribution. However, the non mean-field hypothesis allow to keep the correlation between arms distributions.

Furthermore, `VTS` from Urteaga and Wiggins (2018) scales very poorly with the size of the problem. The variational parameters are very large: $\alpha \in \mathbb{R}^{K\times M}$, $\beta \in \mathbb{R}^{K\times M}$, $\gamma \in \mathbb{R}^{K\times M}$, $u \in \mathbb{R}^{K\times M\times d}$, $V \in \mathbb{R}^{K\times M\times d\times d}$ where K is the number of arms, M is the number of mixtures and d the parameter dimension. In addition, the parameter updating step is also very costly in term of memory and speed. We have re-implemented an efficient version of their algorithm in JAX in order to scale as much as possible but many memory problems occur.

Finally, our algorithm comes with theoretical guarantees in the Linear Bandit case and outperforms empirically the others approximate TS methods. `VTS` performs poorly in practice and has no theoretical guarantee, even in the Linear case.

43 Hyper-parameters tuning

This section summarizes the different grid-search used to compute all the plots in this paper for the algorithms: LinTS, LMC-TS„ `VITS − I`, `VITS − II` and `VITS − II Hessian-free`.

Parameter	Value
inverse temperature η	$10, 100, 500, 1000$
regularization λ	$0.1, 1, 10$

Table 9.31: LinTS hyperparameter grid-search

Parameter	Value
inverse temperature η	$10, 100, 500, 1000$
regularization λ	$0.1, 1, 10$
Nb gradient steps K_t	10, 50
learning rate h	$0.001, 0.01, 0.1$

Table 9.32: LMC-TS hyperparameter grid-search

Parameter	Value
inverse temperature η	$10, 100, 500, 1000$
regularization λ	$0.1, 1, 10$
Nb gradient steps K_t	10
learning rate h	$0.001/\eta, 0.01/\eta, 0.1/\eta$
Monte Carlo samples	1 (`Hessian`) and 20 (`Hessian-free`)

Table 9.33: `VITS` hyperparameter grid-search

44 Experimental comparison between Langevin Monte Carlo and VI

In this section, we conduct an experimental comparison between Langevin Monte Carlo (LMC) and two variants of Variational Inference (VI), denoted as VI-I and VI-II, in approximating a specific target distribution. Our target distribution is a straightforward Gaussian distribution, represented as $\mathrm{p}_\star = \mathrm{N}(\mu_\star, \Sigma_\star)$. We perform LMC, VI-I, and VI-II for a designated number of iterations. In each iteration, we calculate the Kullback-Leibler distance between the approximated distribution and the target distribution. In this context, all distributions generated by LMC, VI-I, and VI-II take the form of Gaussians. To compute the mean and covariance matrix for LMC, we perform parameter averaging over the results obtained after 1000 burn-in steps (which are excluded from the plotted data). Then, the training is stopped when

$$\mathrm{KL}(q_k, \mathrm{p}_\star) \leq \epsilon \,, \tag{A.395}$$

or if the number of steps exceeds 50000 steps.

Figure A44.31 illustrates the relationship between the condition number of $\Sigma_\star$ and the number of steps needed to achieve (A.395). We conducted these experiments with three different step sizes and repeated them across 100 different seeds. The red dashed line in the figure represents the maximum allowable number of iterations.

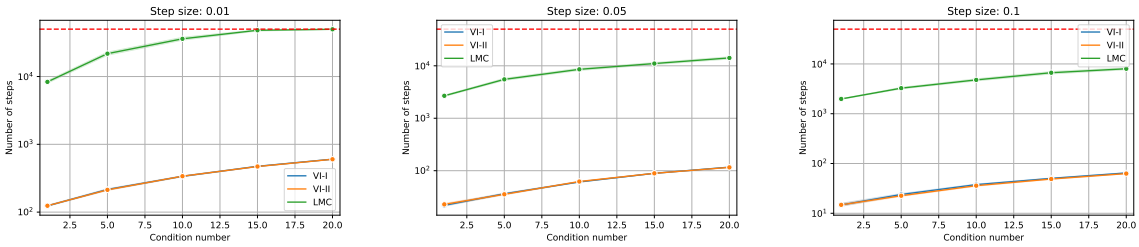


Figure A44.31: Comparison Langevin Monte Carlo and Variational inference

The first observation drawn from these figures is that VI-I and VI-II exhibit identical behavior, even when using a relatively large step size of 0.1. The second finding suggests that both LMC and VI exhibit a linear dependency on the condition number. However, we cannot definitively conclude that one algorithm is more robust in the face of varying condition numbers. Lastly, the third conclusion highlights that VI consistently requires fewer iterations to achieve (A.395).

45 Additional Results on non-contextual bandits

45.1 Linear and logistic bandit on synthetic data (non contectual)

In this subsection, we consider a contextual bandit setting with a parameter dimension $d = 10$ and a number of arms $K = 10$. The bandit environment is simulated by a random vector $\theta^\star \in \mathbb{R}^d$ sampled from a normal distribution $\mathrm{N}(0, \mathrm{I}_d)$ and subsequently scaled to unit norm. To create a complex environment that necessitates exploration, we define the set of contextual vectors as $\mathcal{S} := \{\theta^\star, \theta^\star_\epsilon, x_2, \ldots, x_K\}$. Here, $\theta^\star_\epsilon$ is defined as $(\theta^\star + \epsilon)/\|(\theta^\star + \epsilon)\|_2$, where ϵ is sampled from a normal distribution with mean 0 and standard deviation 0.1. This contextual vector corresponds to a small modification of $\theta^\star$. The other contextual vectors are sampled from a normal distribution $\mathrm{N}(0, 1)$ and then scaled to unit norm.

Linear bandit scenario. Here, the true reward $\mathcal{R}(\cdot|x_a, a)$ associated to an action $a \in \{1, \ldots, K\}$ and an arm $x_a \in \mathbb{R}^d$ corresponds to the distribution of $r_a = x_a^\top \theta^\star + \xi$, where the noise ξ is sampled from $\mathrm{N}(0, \mathrm{I})$. In this complex setting, we can calculate the expected reward for each arm as follows: $\mu_0 = \mathbb{E}[r_0] = 1$, $\mu_1 \approx 1 < \mu_0$, and for any $i > 1$, $\mu_i < \mu_1$. Intuitively, the first and second arms offered high rewards, while the remaining arms offered low rewards. On the other hand, finding the optimal arm is challenging and needs a significant amount of exploration.

Logistic bandit framework. We consider the same contextual set $\mathcal{S}$, but the true reward $\mathcal{R}(\cdot|x_a, a)$ associated to an action $a \in \{1, \ldots, K\}$ and an arm x_a now corresponds to $r_a \sim \mathrm{Ber}(\sigma(\langle x_a, \theta^\star \rangle))$, where Ber is the Bernoulli distribution, and $\sigma(x) = 1/(1 + e^{-x})$ is the logistic function. Similarly to the linear bandit, the logistic framework introduces a complex environment where a significant amount of exploration is required to accurately distinguish between the first and second arm.

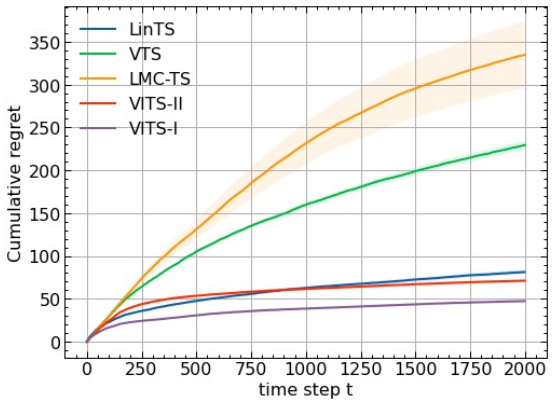


Figure A45.33: Logistic Bandits

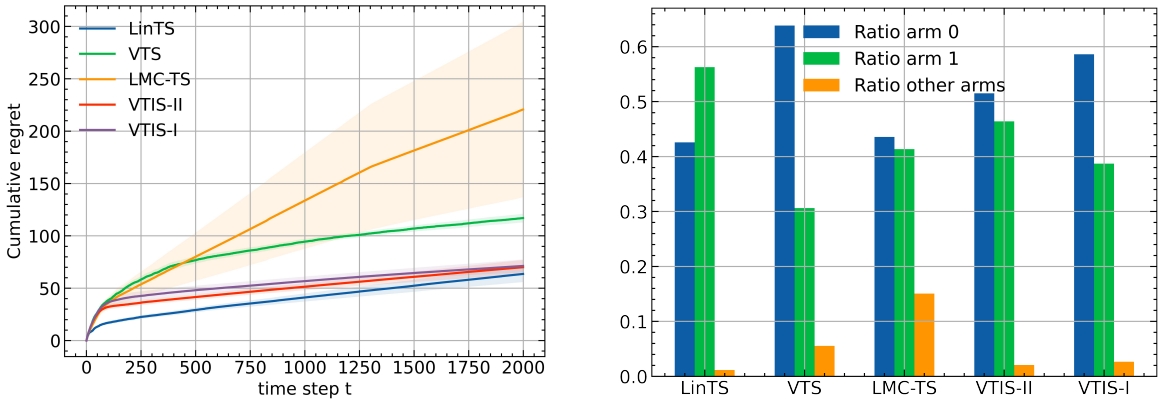


Figure A45.32: Linear Bandits

Figures A45.32 and A45.33 display the cumulative regret (8.2) obtained by various TS algorithms, namely Linear TS (LinTS), Langevin Monte Carlo TS (LMC-TS), Variational TS (VTS), `VITS-I` and `VITS-II` in the linear and logistic bandit settings. The figure shows the mean and standard error of the cumulative regret over 20 samples. As depicted in Figure A45.32, `VITS-I` outperforms the other approximate TS algorithms in the linear bandit scenario. Note that the cumulative regret of `VITS-I` and `VITS-II` is comparable to that of Lin-TS, which uses the true posterior distribution. This observation highlights the efficiency of the variational TS algorithms in approximating the true posterior distribution and achieving similar performance to the Lin-TS algorithm. Figure A45.33 shows that `VITS` outperforms all other TS algorithms in the logistic setting too. This highlights the importance of employing approximation techniques in scenarios where the true posterior distribution cannot be sampled exactly. Moreover, both figures illustrate that `VITS-II` achieves a comparable regret to `VITS-I` while significantly reducing the computational complexity of the algorithm. Finally, as emphasized earlier, the settings we have chosen require a good tradeoff between exploration and exploitation that LMC-TS cannot achieve, as illustrated by the histogram in Figure A45.32.

46 Details about experiences in synthetic contextual bandits with synthetic data

In this subsection, we provide more details about the toy example derive in this paper. Firstly, we consider a fixed pool of arms denoted as $P = [\tilde{x}_1, \dots, \tilde{x}_n]$ with $n = 50$, where each arm $\tilde{x}_i$

follows a normal distribution $\mathrm{N}(0_d, \mathrm{I}_d)$. Then, at each step $t \in [T]$, for every arm, we randomly sample a vector $\tilde{x}_i$ from the pool P, and the contextual vector associated with this arm is defined as $x = \tilde{x}_i + \zeta\epsilon$, where $\epsilon \sim \mathcal{N}(0_d, \mathrm{I}_d)$. The bandit environment is simulated using a random vector $\theta^\star$ sampled from a normal distribution $\mathrm{N}(0_d, \sigma^\star \mathrm{I}_d)$. We opted for $\sigma^\star = 1/d$ to ensure that the variance of the scalar product $x^\top \theta^\star$ remains independent of the dimension d. Indeed, both linear and quadratic settings, the reward only depends on the scalar product between the context and the true parameter. If we denote by $x[i]$ and $\theta^\star[i]$ the i^{th} coordinate of the vector x and $\theta^\star$ respectively, then the scalar product is defined by

$$x^\top \theta^\star = \sum_{i=1}^{d} x[i]\theta^\star[i],$$

and its variance is defined by

$$\begin{aligned}
\mathbb{V}[x^\top \theta^\star] &= \mathbb{V}[\sum_{i=1}^{d} x[i]\theta^\star[i]] \\
&= \sum_{i=1}^{d} \mathbb{V}[x[i]]\mathbb{V}[\theta^\star[i]] \\
&= d\sigma^\star \, \mathbb{V}[x[i]].
\end{aligned}$$

In the previous equations we have used that all coordinates are independents identically distributed and centered. Therefore, taking $\sigma^\star = 1/d$ ensure that the variance of the scalar product remains independent of d. In the linear bandit setting, the reward depends linearly on the contextual vector x, more precisely,

$$r = x^\top \theta^\star + \alpha\epsilon \ ,$$

where $\epsilon \sim \mathrm{N}(0_d, \mathrm{I}_d)$. However, to maintain problem complexity independent of ζ, we have set the signal-to-noise ratio to a fixed value of 1. This signal-to-noise ratio is the ratio between $\mathbb{E}[(x^\top \theta^\star)^2]$ and $\mathbb{E}[(\alpha\epsilon)^2]$. Firstly,

$$\begin{aligned}
\mathbb{E}[(x^\top \theta^\star)^2] &= \mathbb{V}[x^\top \theta^\star] \\
&= \mathbb{V}[x[i]] \\
&= 1 + \zeta^2 \ ,
\end{aligned}$$

where in the last equation we have used that $x = \tilde{x}_i + \zeta\epsilon$ and $\mathbb{V}[x[i]] = 1 + \zeta^2$. Moreover, the denominator of the signal-to-noise ratio is $\mathbb{E}[(\alpha\epsilon)^2] = \alpha^2$. Consequently, a signal-to-noise ratio equals to 1 implies that $\sqrt{1 + \zeta^2} = \alpha$.

In the quadratic bandit setting, the reward depends quadratically on the contextual vector x, more precisely,

$$r = (x^\top \theta^\star)^2 + \alpha\epsilon \ ,$$

where $\epsilon \sim \mathrm{N}(0, \mathrm{I})$. In this setting, the reward also depends only on the scalar product between x and $\theta^\star$, thus, we also choose $\sigma^\star = 1/d$. We also ensure a signal-to-noise equal to 1, it implies a more sophisticated condition on the noise: $\alpha = (\zeta^2 + 1)\sqrt{3 + 6/d}$. More precisely, in the quadratic setting, the signal-to-noise ratio is defined as follow

$$\frac{\mathbb{E}[(x^\top \theta^\star)^4]}{\mathbb{E}[(\alpha\epsilon)^2]} = 1.$$

Firstly,

$$
\begin{aligned}
\mathbb{E}(x^\top\theta^\star)^4] &= \mathbb{E}[(\sum_{i=1}^{d} x[i]\theta^\star[i])^4] \\
&= \mathbb{E}\Bigg[\sum_{i=1}^{d}(x[i]\theta^\star[i])^4 + 4\sum_{i=1}^{d}\sum_{j\neq i}(x[i]\theta^\star[i])^3 x[j]\theta^\star[j] + 6\sum_{i=1}^{d}\sum_{j<i}(x[i]\theta^\star[i])^2(x[j]\theta^\star[j])^2 \\
&+ 12\sum_{i=1}^{d}\sum_{j\neq i}\sum_{k\neq i,k<j}(x[i]\theta^\star[i])^2 x[j]\theta^\star[j]x[k]\theta^\star[k] + 24\sum_{i=1}^{d}\sum_{j<i}\sum_{k<j}\sum_{l<k} x[i]\theta^\star[i]x[j]\theta^\star[j]x[k]\theta^\star[k]x[l]\theta^\star[l]\Bigg] \\
&= \sum_{i=1}^{d}\mathbb{E}[x[i]^4]\mathbb{E}[\theta^\star[i]^4] + 6\sum_{i=1}^{d}\sum_{j<i}\mathbb{E}[x_i^2]\mathbb{E}[x_j^2]\mathbb{E}[\theta^\star[i]^2]\mathbb{E}[\theta^\star[j]^2] \\
&= \frac{9(\zeta^2+1)^2}{d} + 6\binom{d}{2}\frac{(\zeta^2+1)^2}{d^2} \\
&= (\zeta^2+1)^2(\frac{9}{d} + \frac{3(d-1)}{d}) \\
&= (\zeta^2+1)^2(\frac{6}{d}+3)
\end{aligned}
$$

which gives that $\alpha = (\zeta^2+1)\sqrt{3+6/d}$

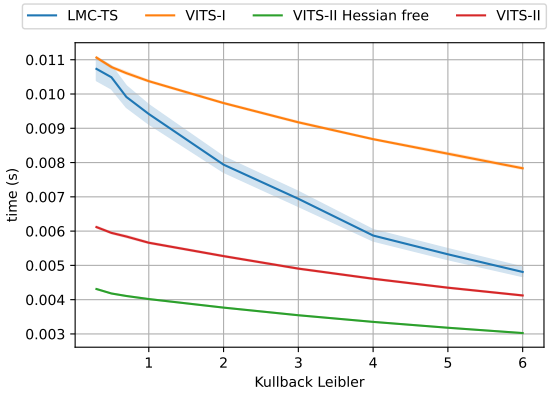
47 Computation complexity and Computational Power

We conduct an experimental comparison between Langevin Monte Carlo (LMC) and three variants of Variational Inference, denoted as $\texttt{VITS}-\texttt{I}$, $\texttt{VITS}-\texttt{II}$ and $\texttt{VITS}-\texttt{II Hessian-free}$, in approximating a specific target distribution. Our target distribution is a straightforward Gaussian distribution, represented as $p^\star = \mathrm{N}(\mu^\star, \Sigma^\star)$. At each iteration, we calculate the Kullback-Leibler distance between the approximated distribution and the target distribution. In this context, all distributions generated by LMCTS, $\texttt{VITS}-\texttt{I}$, $\texttt{VITS}-\texttt{II}$ and $\texttt{VITS}-\texttt{II Hessian-free}$ take the form of Gaussians. To compute the mean and covariance matrix for LMC, we perform parameter averaging. As both the posterior and its approximation are Gaussians, the Kullback-Leibler divergence is easily tractable. Then, the training is stopped when

$$\mathrm{KL}(q_k, p^\star) \leq \epsilon$$

or if the number of steps exceeds 10000 steps.

The following Figure illustrates the relationship between the obtained Kullback- Leibler divergence and the computational time needed to achieve 47. The computational time is the total time (in second) required to run all updating steps of the algorithm. This experiment is repeated across 1000 different seeds to compute the confidence interval. We decide not to compare with LinTS or LinUCB algorithms as they do not allow to approximate complex posteriors compared to LMCTS and VITS algorithms.



This figure shows that `VITS − II` and `VITS − II Hessian-free` are faster (in term of computational time) than LMCTS to obtain a certain Kullback-Leibler divergence. Note that `VITS − I` is the slowest algorithm, this is due to the costly inverse matrix calculation.

In this work, we use GPUs v100-16g or v100-32g for running our code with GPU Nvidia Tesla V100 SXM2 16 Go and CPUs with 192 Go per node.

ECOLE
DOCTORALE
DE MATHEMATIQUES
HADAMARD

Titre : Apprentissage par renforcement robuste : théorie et pratique

Mots clés : processus décisionnel de Markov, appentissage par renforcement, robustesse

Résumé : 'apprentissage par renforcement (RL) est un paradigme d'apprentissage automatique qui aborde la question de la prise de décision séquentielle. Dans ce paradigme, l'algorithme, désigné comme un agent, réagit à des interactions avec un environnement. À chaque interaction, l'agent effectue une action dans l'environnement, observe un nouvel état de l'environnement et reçoit une récompense en conséquence. L'objectif de l'agent est d'optimiser une récompense cumulative, qui est définie par l'utilisateur pour s'aligner sur la tâche spécifique à accomplir dans l'environnement. La théorie du processus décisionnel de Markov (MDP) est utilisée pour formaliser ce concept. Cependant, en cas de mauvaise spécification du modèle ou d'erreur dans la fonction de transition de l'environnent ou de la récompense, les performances du RL peuvent diminuer rapidement. Pour résoudre ce problème, le concept de MDP robustes a émergé, l'objectif étant d'identifier la politique optimale sous l'hypothèse que le noyau de transition appartient à un ensemble d'incertitude. Cette thèse présente une étude théorique de la complexité d'échantillonnage des MDP robustes, ou de la quantité de données nécessaires pour atteindre une erreur arbitrairement petite. Ces résultats démontrent que dans certains cas, cette complexité peut être inférieure à celle des MDP classiques, ce qui constitue une voie prometteuse pour concevoir de nouveaux algorithmes efficaces sur le plan de l'échantillonnage. La thèse se poursuit par des propositions de nouveaux algorithmes RL robustes pour renforcer les performances de RL ayant des ensembles d'action continus. Notre méthode est basée sur les MDP averses aux risques et les jeux à somme nulle, dans lesquels l'adversaire peut être considéré comme un agent qui change l'environnement dans le temps. En conclusion, la dernière section présentera des nouvelles tâches pour l'évaluation des algorithmes RL robustes, qui manquent de références pour l'évaluation des performances.

Title : Robust Reinforcement Learning : Theory and Practice

Keywords : Robust Markov Decision Process, Robust Reinforcement Learning

Abstract : einforcement learning (RL) is a machine learning paradigm that addresses the issue of sequential decision-making. In this paradigm, the algorithm, designated as an agent, responds to interactions with an environment. At each interaction, the agent performs an action within the environment, observes a new state of the environment, and receives a reward in consequence. The objective of the agent is to optimise an cumulative reward, which is defined by the user to align with the specific task at hand within the environment. The Markov Decision Process (MDP) theory is used in order to formalise these concepts. However, in the event of mispecifications or errors in the transition or reward function, the performance of RL may decline rapidly. To address this issue, the concept of robust MDPs has emerged, whereby the objective is to identify the optimal policy under the assumption that the transition kernel belongs to a bounded uncertainty set. This thesis presents a theoretical study of the sample complexity of robust MDPs, or the amount of data required to achieve an arbitrary small convergence error. It demonstrates that in certain cases, the sample complexity of robust MDPs can be lower than for classical MDPs, which is a promising avenue for the derivation of sample-efficient algorithms. The thesis then goes on to derive new robust RL algorithms to strengthen the performance of RL in continuous control. Our method is based on risk-averse MDPs and zero-sum games, in which the adversary can be seen as an agent that change the environment in the time. In conclusion, the final section present a benchmark for the evaluation of robust RL algorithms, which currently lack a reproducible benchmarks for performance assessment.

Institut Polytechnique de Paris
91120 Palaiseau, France