Experimental and Theoretical Analysis of Reinforcement Learning Algorithms

David Brellmann

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2024IPPAE008

Thèse de doctorat

Experimental and Theoretical Analysis of Reinforcement Learning Algorithms

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à École nationale supérieure de techniques avancées

École doctorale n°626 École doctorale de l'Institut Polytechnique de Paris (EDIPP)
Spécialité de doctorat: Mathématiques et Informatique

Thèse présentée et soutenue à Palaiseau, le 01/07/2024, par

David Brellmann

Composition du Jury :

Rémi Munos Directeur de Recherche, DeepMind/INRIA	Président
Odalric-Ambrym Maillard Chargé de recherche, INRIA Lille (Scool)	Rapporteur
Marcello Restelli Associate Professor, Politecnico di Milano	Rapporteur
Yaqi Duhan Assistant Professor, New York University	Examinateur
Goran Frehse Enseignant-chercheur, ENSTA Paris (U2IS)	Directeur de thèse
David Filliat Enseignant-chercheur, ENSTA Paris (U2IS)	Co-directeur de thèse

À mes parents,

Remerciements

J'aimerais tout d'abord exprimer toute ma reconnaissance à mon directeur de thèse, Goran Frehse, pour sa confiance et son soutien tout au long de cette thèse. Je souhaite le remercier d'avoir toujours été bienveillant et pédagogue et de m'avoir fourni un encadrement sans faille, aussi bien scientifique qu'humain. J'ai beaucoup appris auprès de toi et je continuerai à appliquer tes précieux conseils. Je remercie mon co-directeur de thèse, David Filliat, pour sa constante disponibilité, son écoute, sa bienveillance et ses conseils avisés. Je remercie aussi Eloïse Berthier avec qui j'ai eu la chance de collaborer. Tu t'es toujours montrée disponible, généreuse et tu m'as fait découvrir de nouveaux horizons mathématiques. Ce travail n'aurait pas également été possible sans l'École Polytechnique, sans Armines et sans l'École Nationale Supérieure des Techniques avancées de Paris qui m'ont permis grâce à leurs soutiens financiers de me consacrer sereinement à mes travaux de recherches.

J'adresse mes remerciements à Odalric-Ambrym Maillard et Marcello Restelli pour avoir accepté de rapporter ce manuscrit malgré leurs emplois du temps très chargés ainsi qu'à Rémi Munos et Yaqi Duan qui me font l'honneur de l'examiner. Je remercie également mon comité de suivi de thèse, composé de Sylvain Lamprier et Olivier Pietquin, pour leurs précieux retours et conseils.

Je tiens aussi à remercier toutes mes collègues de l'U2IS pour leur bienveillance et conseils, notamment Abdelmouaiz Tebjou, Mohamed Fnadi, Gaël Parfait et Alexandre Chapoutot.

Sans oublier un grand merci à Gianni, Pavan, Raphaël et Gwendal pour leur amitié, leur générosité, leur soutien et pour tous ces bons moments vécus. Merci aussi à mes amis Aurélien, Christian, Mathieu, Vincent et Tessa qui ont été particulièrement présents ces dernières années et qui ont su apporter un peu de légèreté par leur bonne humeur. Je remercie toutes les personnes, amis et professeurs, qui m'ont soutenu au cours de mes études.

Un grand merci à ma compagne, Fiorella, d'avoir toujours cru en moi dans les bons comme dans les mauvais moments. Sans toi, mon quotidien aurait été beaucoup moins heureux et tout cela n'aurait pas été possible.

Enfin, il me serait impossible de terminer sans remercier ma famille pour leur soutien indéfectible depuis toujours. Je tiens à remercier mes grands-parents pour leurs encouragements et leurs sollicitudes. Merci à mes parents d'avoir toujours été présents, aimants et de m'avoir poussé à donner le meilleur de moi-même. Merci à mon papa de m'avoir encouragé dans la voie scientifique et de m'avoir appris le sens de la rigueur. Merci à ma maman qui avec patience et persevérance m'a donné le goût d'apprendre. Merci à ma sœur Léa, ma confidente, de me supporter et de m'avoir accompagné depuis toutes ces années.

Abstract

In Reinforcement Learning (RL), an agent learns how to act in an unknown environment in order to maximize its reward in the long run. In recent years, the use of neural networks has led to breakthroughs, e.g., in scalability. However, there are still gaps in our understanding of how to best employ neural networks in RL. In this thesis, we improve the usability of neural networks in RL in two ways, presented in two separate parts. First, we present a theoretical analysis of the influence of the number of parameters on learning performance. Second, we propose a simple feature preprocessing based on the Fourier series, which empirically improves performance in several ways.

In the first part of this thesis, we study how the number of parameters influences performance. While in supervised learning, the regime of over-parameterization and its benefits are well understood, the situation in RL is much less clear. We present a theoretical analysis of the influence of number of parameters and L2 regularization on performance. We identify the ratio between the number of parameters and the number of visited states as a crucial factor and define over-parameterization as the regime when this ratio is larger than one. We observe a double descent phenomenon, i.e., a sudden drop in performance around the parameter/state ratio of one. Our analysis is based on the regularized Least-Squared Temporal Difference (LSTD) algorithm with random features in an asymptotic regime, as both the number of parameters and states go to infinity while maintaining a constant ratio. We derive deterministic limits of the empirical, the true Mean-Squared Bellman Error (MSBE), and the true Mean-Squared Value Error (MSVE) that feature correction due to the constant ratio between the number of parameters and distinct visited states. We experimentally associate those correction terms with the double descent phenomenon and an implict regularization of the model. We demonstrate that the correction terms vanish as either the L2 regularization increases, the number of parameters increases, or the number of unvisited states decreases.

In the second part of this thesis, we study the preprocessing of features through a Fourier series. In addition to the number of parameters, the amount of optimization that can be achieved in practice remains limited. Neural networks behave thus as under-parameterized models that are also regularized through early stopping. This regularization induces a spectral bias since fitting high-frequency components of the value function requires exponentially more gradient update steps than the low-frequency ones. We propose a simple Fourier mapping for preprocessing, which improves the learning of high-frequency components and thus helps to overcome the spectral bias in RL. We present experiments indicating that this can lead to significant performance gains in terms of rewards and sample efficiency. Furthermore, we observe that this preprocessing increases the robustness with respect to hyperparameters, leads to smoother policies, and benefits the training process by reducing learning interference, encouraging sparsity, and increasing the expressiveness of the learned features.

Résumé

En apprentissage par renforcement (RL), un agent apprend comment agir dans un environnement inconnu de façon à maximiser sa récompense sur le long terme. Ces dernières années, l'utilisation de réseaux de neurones artificiels a conduit à de nombreuses avancées, notamment en termes de scalabilité. Cependant, de nombreuses lacunes subsistent dans notre compréhension de la meilleure manière d'employer les réseaux de neurones en RL. Dans cette thèse, nous proposons d'améliorer l'utilisation des réseaux de neurones en RL de deux manières, présentées dans deux parties distinctes. La première partie présente une analyse théorique de l'impact du nombre de paramètres sur la performance d'apprentissage. La seconde partie propose un prétraitement simple des données, basé sur la série de Fourier, qui améliore empiriquement les performances des réseaux de neurones de plusieurs façons.

Dans la première partie de cette thèse, nous étudions l'influence du nombre de paramètres sur la performance. Alors que dans l'apprentissage supervisé, le régime de surparamétrisation et ses avantages sont assez bien compris, la situation en RL est beaucoup moins claire. Nous présentons donc une analyse théorique sur l'influence du nombre de paramètres et sur l'impact d'un terme de régularisation L2 sur la performance. Nous identifions le rapport entre le nombre de paramètres et le nombre d'états visités comme un facteur crucial et définissons la surparamétrisation comme le régime où ce rapport est supérieur à un. Nous observons un phénomène de double descente, caractérisé par une chute soudaine de performance au-delà d'un rapport paramètres/états visités de un. Notre analyse est basée sur l'algorithme de Least-Squares Temporal Difference learning (LSTD) doté de caractéristiques aléatoires et d'un terme de regularisation L2 dans un régime asymptotique, où le nombre de paramètres et d'états visités tendent vers l'infini tout en maintenant un rapport constant. Nous dérivons les limites déterministes de l'erreur quadratique moyenne de Bellman (MSBE) basée sur les échantillons collectés durant l'entraînement, de la vraie MSBE, et de l'erreur quadratique moyenne de la fonction de valeur (MSVE) qui comportent notamment des termes correctifs induits par le rapport fini nombre de paramètres/états visités. Nous associons expérimentallement ces termes correctifs au phénomène de double descente et à une régularisation implicite du modèle. Nous démontrons que ces termes correctifs diminuent soit lorsque le terme de pénalité associé à la régularisation L2 augmente, soit lorsque le nombre de paramètres augmente, soit lorsque le nombre d'états non visités diminue.

Dans la seconde partie de cette thèse, nous proposons l'étude d'un prétraitement des données basé sur la série de Fourier. En effet, outre le nombre de paramètres, le nombre d'optimisations réalisé en pratique reste souvent limité. Par conséquent, les réseaux de neurones tendent souvent à se comporter comme des modèles sous-paramétrisés régularisés par un arrêt prématuré. Cette forme de régularisation induit notamment un biais spectral, puisque l'apprentissage des composantes à haute fréquence de la fonction cible requiert exponentiellement plus d'itérations dans la descente de gradient stochastique que pour les composantes à basse fréquence. Pour pallier à ce problème, nous proposons un prétraitement des données basé sur la série de Fourier afin d'améliorer l'apprentissage des composantes à haute fréquence et surmonter le biais spectral en RL. Nous présentons des expériences indiquant que ce prétraitement peut conduire à des améliorations significatives des performances, en termes de récompenses obtenues et de données utilisées. De plus, nous observons que ce prétraitement favorise une plus grande robustesse face aux hyperparamètres, conduit à l'élaboration de politiques plus régulières, et bénéficie au processus d'entraînement en réduisant l'interférence d'apprentissage, en encourageant l'apprentissage de caractéristiques distinctes et *sparses* (ou creuses), et en augmentant l'expressivité des caractéristiques apprises.

Contents

List of Figures

List of Tables

List of Symbols

$\mathbb{R}$	Set of real numbers.
$\mathbb{C}$	Set of complex numbers.
$[n]$	Set of integers between 1 and n.
$\mathfrak{I}(\cdot)$	Imaginary part of a complex number.
$\text{diag}(\cdot)$	Diagonal operator, for $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\text{diag}(\boldsymbol{A}) \in \mathbb{R}^n$ is the vector with entries $\{\boldsymbol{A}_{ii}\}_{i=1}^n$; for $\boldsymbol{a} \in \mathbb{R}^n$, $\text{diag}(\boldsymbol{a}) \in \mathbb{R}^{n\times n}$ is the diagonal matrix taking $\boldsymbol{a}$ as its diagonal.
$(\cdot)^T$	Transpose operator.
$\text{Tr}(\cdot)$	Trace.
$\lvert\cdot\rvert$	Cardinality operator of a set, module of a complex number.
$\lVert\cdot\rVert$	Operator norm of a matrix and Euclidean norm of a vector.
$\lVert\cdot\rVert_{\boldsymbol{A}}$	Norm induced by a matrix $\boldsymbol{A}$, $\lVert\boldsymbol{v}\rVert_{\boldsymbol{A}} = \boldsymbol{v}^T\boldsymbol{A}\boldsymbol{v}$.
$\lVert\cdot\rVert_\infty$	Infinity norm of a matrix, $\lVert\boldsymbol{A}\rVert_\infty = \max_{i,j} \boldsymbol{A}_{ij}$.
$\lVert\cdot\rVert_F$	Frobenius norm of a matrix, $\lVert\boldsymbol{A}\rVert_F = \sqrt{\text{Tr}(\boldsymbol{A}\boldsymbol{A}^T)}$.
$\nu(\cdot)$	Spectrum of a matrix.
$\nu_{\max}(\cdot)$	Largest eigenvalue of a matrix.
$\nu_{\min}(\cdot)$	Smallest eigenvalue of a matrix.
$\mathbb{E}[\cdot]$	Expectation operator.
$\text{Var}[\cdot]$	Variance operator.
$\xrightarrow{a.s}$	Almost surely convergence. We say a sequence $x_n \xrightarrow{a.s} x$ if $\Pr(\lim_{n\to\infty} x_n = x) = 1$.
$H(\cdot)$	Symmetric part, $H(\boldsymbol{A}) = \frac{\boldsymbol{A}+\boldsymbol{A}^T}{2}$.
$\mathcal{O}(1), o(1)$	A sequence x_n is bounded or converges to zero as $n \to \infty$, respectively.

Chapter 1

Introduction

Machine Learning is a branch of Artificial Intelligence (AI) that enables machines to "learn" automatically from raw data and past experiences in order to identify patterns and make predictions without requiring explicit programming of rules or behaviors (Bishop et al., 1995; Sutton and Barto, 2018; Géron, 2022). Machine Learning algorithms learn directly from a large volume of data instead of relying on any predetermined equation serving as a model. With the advancement of computing technology and the onset of the "big data" era, characterized by an increasing collection of data, machine learning algorithms have seen widespread use. They are applied to many fields including language processing (Vaswani et al., 2017; Brown et al., 2020; Touvron et al., 2023; Team et al., 2023), computer vision (He et al., 2016; Redmon et al., 2016; Krizhevsky et al., 2017), robotics (Trautman and Krause, 2010; Berkenkamp et al., 2016; Kumar et al., 2021), agriculture (Meshram et al., 2021), medicine (Deo, 2015; Popova et al., 2018), finance (Hambly et al., 2023), video games (Mnih et al., 2015; Vinyals et al., 2019; Gillberg et al., 2023), and recommendation systems (Hu et al., 2008; Covington et al., 2016; He et al., 2017).

Reinforcement Learning (RL) is a branch of Machine Learning, where an agent learns how to solve a task in an unknown environment (Sutton, 1988). The goal of the agent is to maximize a numerical reward signal over time through trial and error. In other words, RL enables a computer or robot to learn how to perform a task by trying different strategies and identifying the most effective ones. More precisely, the learner receives rewards for its actions and adjusts its behavior accordingly to maximize the reward signal in the future. In RL, the objective for the learner is to optimize its behavior by making decisions that result in favorable rewards based on its past experiences and the feedback it receives from the environment. RL is particularly convenient as it enables learners to solve tasks through observed data, without the need for a specific model or the explicit programming of rules or behaviors. This characteristic is particularly useful in complex stochastic environments, where it is impractical to define the equations of a model or to predefine a fixed set of rules or behaviors. Another benefit of RL algorithms is their capacity to learn and adapt their behavior in real time. This can be beneficial across a wide range of applications, and RL has been applied in areas such as natural language processing (He et al., 2015; Luketina et al., 2019), robotics (Trautman and Krause, 2010; Berkenkamp et al., 2016; Kumar et al., 2021), autonomous driving (Likmeta et al., 2020; Kiran et al., 2021), video games (Mnih et al., 2015; Vinyals et al., 2019; Gillberg et al., 2023), and recommendation systems (Chen et al., 2019; Afsar et al., 2022).

The performance of machine learning and RL algorithms mainly depends on the representation of

the data they need to handle (Bishop et al., 1995; Sutton, 1988). The representation that includes the crucial information for performing a task is known as a set of *features*, and depends on the problem being solved. For instance, the glucose level is more relevant as a feature when predicting the risk of diabetes than it is for cardiovascular disease. For the latter, the cholesterol level is a more suitable feature. In conventional approaches, features are typically hand-designed according to the specific task and then passed to classical machine learning algorithms to make decisions. Choosing appropriate features for a task is a critical way of adding prior domain knowledge. However, determining which features to use can be a complex challenge. In domains such as computer vision or language processing, determining the significance of a pixel in an image or a word in a sentence may be particularly challenging (Bengio et al., 2013).

In recent years, the use of artificial *neural networks* has led to breakthroughs due to their ability to learn features from raw data without prior knowledge (Schmidhuber, 2015). Interest in RL exploded in the wake of the results from Mnih et al. (2015), who demonstrated that neural networks could learn to play a collection of Atari games, using screen images as input and applying a variant of Q-learning. Since then, RL algorithms using neural networks, i.e., *deep RL* algorithms have shown impressive performance in many domains, including robotics and natural language (Schulman et al., 2017; Haarnoja et al., 2018; Espeholt et al., 2018). Neural networks are particularly promising due to their scalability, enabling their application to a wide range of high-dimensional sequential decision-making problems. They excel in problems where other classes of techniques currently fail to provide solutions. For example, neural networks perform better than humans in complex games like Go or Starcraft (Silver et al., 2018; Vinyals et al., 2019; Perolat et al., 2022). While they perform well on challenging tasks, their theoretical understanding remains limited. Many questions arise: how do neural networks generalize? What do they learn? How many parameters do we need to achieve good performance? How many samples do we need? What are their limitations? The difficulty is further exacerbated in RL by a myriad of new challenges that limit the scope of these works, such as the absence of true targets or the non-i.i.d nature of the collected samples (Kumar et al., 2020; Luo et al., 2020; Lyle et al., 2021; Dong et al., 2020). There are still gaps in our understanding of how to best employ neural networks in RL. In this thesis, we contribute to the domain of neural networks in deep RL in two ways, presented in two separate parts. First, we present a theoretical analysis of the influence of the number of parameters and the level of regularization on learning performance. Second, we propose a simple preprocessing based on the Fourier series, which empirically improves performance in several ways. The outline and primary contributions of this thesis are summarized below.

1.1 Outline

We begin the thesis with Part I, which covers the basics of RL and provides the preliminaries necessary to follow the rest of the thesis. If the reader is already familiar with RL, we suggest skipping these chapters and going directly to Part II. Chapter 2 provides a brief introduction to RL. We first recall the concept of a Markov Decision Process and the definition of value functions and policies. After introducing dynamic programming algorithms, we then provide a brief overview of traditional tabular value-based algorithms, such as TD(0) and Q-learning. In Chapter 3, we discuss function approximation within the framework of value-based algorithms. We introduce the concept of the Markov Reward Process, which is used to mathematically describe the value evaluation. We then present value-based algorithms using linear models and neural networks with their respective

objective functions. In Chapter 4, we review the Least-Squares Temporal Difference Learning (LSTD) algorithm, its connection with stochastic gradient-based approaches, and its derivation from least-squares methods. After introducing LSTD, we discuss several regularization methods that can be applied to LSTD and Temporal Difference (TD) learning algorithms to avoid overfitting, with a particular emphasis on l_2 regularization. Following the introductory part, we proceed into two Parts that contain the main body of our work.

Part II, **Double Descent in Least-Squares Temporal Difference Learning**, investigates how the number of parameters and the level of regularization influence performance. Temporal Difference (TD) learning algorithms are widely used in deep RL as they are simple and efficient. Their performance is heavily influenced by the size of the neural network. While the regime of over-parameterization and its benefits are well understood in supervised learning, the situation in RL is much less clear. In this part, we present a theoretical analysis of the influence of network size and l_2-regularization on the performance of TD learning algorithms. This part is mainly based on our work *On Double Descent in Reinforcement Learning with LSTD and Random Features*, with Eloïse Berthier, David Filliat, and Goran Frehse, accepted for publication in the International Conference on Learning Representations (ICLR), 2024. Part II is organized as follows:

- In Chapter 5, we start by presenting the classical bias-variance tradeoff theory, which guided the selection of models and the choice of the number of parameters in traditional machine learning. This theory has been used to select models rich enough to express underlying structure in data and simple enough to avoid fitting of noise. Yet, as shown in this chapter, practitioners usually prefer using a large amount of parameters and interpolating the training data. We then briefly define and review the double descent theory introduced in supervised learning to explain the good performance of over-parameterized models. This chapter can be skipped by readers familiar with the phenomenon of double descent.

- In Chapter 6, we propose a novel theoretical framework for studying neural value function approximation in high-dimensional problems. Indeed, theoretical studies of TD learning algorithms often explore high-dimensional problems in asymptotic regimes, where the number of samples tends to infinity while the number of parameters remains constant (Tsitsiklis and Van Roy, 1996; Bradtke and Barto, 1996; Nedić and Bertsekas, 2003; Sutton, 1988). When TD learning algorithms are applied to neural networks, it is commonly assumed that the number of parameters tends to infinity with either a fixed or infinite number of samples without providing details on the relative magnitudes of those dimensions (Cai et al., 2019; Agazzi and Lu, 2022; Berthier et al., 2022; Xiao et al., 2021). In this chapter, we propose studying TD learning algorithms using neural networks in a novel double asymptotic regime, where both the number of parameters and states visited go to infinity while maintaining a constant ratio called model complexity. In this double asymptotic regime, we approximate TD learning algorithms using two-layer neural networks with the regularized Least-Squared Temporal Difference (LSTD) algorithm on random features by leveraging the lazy training regime.

- In Chapter 7, we first introduce the mathematical framework of Random Matrix Theory and concentrations results used to study the performance of regularized LSTD in the double asymptotic regime, and then we present our main theoretical results. In particular, we identify the resolvent of a non-symmetric positive-definite matrix that emerges as a crucial factor in the performance analysis of regularized LSTD. We provide a deterministic equivalent of this resolvent in the double asymptotic regime. Using the deterministic equivalent of the resolvent

and concentration results, we analyze the performance of regularized LSTD in the double asymptotic regime with the derivation of deterministic equations for the asymptotic empirical Mean-Squared Bellman Error on the collected transitions, the asymptotic Mean-Squared Bellman Error (MSBE), and the asymptotic Mean-Squared Value Error (MSVE). The deterministic forms expose correction terms that arise from the double asymptotic regime. We show that the correction terms vanish as the l_2-regularization increases or the model complexity (i.e., the ratio between the number of parameters and number of states visited) goes to infinity. We also show that the influence of the l_2-regularization parameter decreases as the model complexity increases.

- In Chapter 8, after reviewing kernel methods and their Mercer feature spaces, we revisit the results of Chapter 7 in the Mercer feature space approximated by the random features. This reformulation enables us to rewrite all the results using a similar expression and highlights the connections that exist between the asymptotic error functions of random feature models and the corresponding errors of a regularized kernel LSTD predicator. In particular, this reformulation provides a better understanding of correction terms that arise from the double asymptotic regime and highlights an implicit regularization induced by the model complexity.

- In Chapter 9, we present our experimental results and show our theory closely matches empirical results for regularized LSTD on a range of both toy and small real-world environments; where both the number of states visited m and the number of parameters N are fixed, but for which our asymptotic predictions still gives accurate predictions. From our experiments, we identify two distinct regimes: an *under-parameterized regime* where $N/m < 1$ and an *over-parameterized regime* where $N/m > 1$. Each regime exhibits different behaviors in the empirical MSBE, the true MSBE, and the MSVE. Notably, in the phase transition around $N/m = 1$, we observe a double descent phenomenon similar to what has been reported in supervised learning, with a peak in the true MSBE and MSVE around $N/m = 1$. For the empirical MSBE and MSVE on the collected transitions, the phase transition is characterized by an almost zero training error and a perfect fit with the training data. We experimentally associate correction terms found in Chapter 7 and 8 with the double descent phenomenon. We also show that correction terms, and therefore the double descent phenomenon, empirically vanish when the number of unvisited states goes to zero or the level of regularization increases. Finally, we show that the discount factor has no influence on the double descent phenomenon.

Part III, **Features Encoding in Deep Reinforcement Learning**, studies the preprocessing of neural networks through a Fourier series to enhance the performance and sample efficiency of deep RL algorithms. This part is mainly based on our work *Fourier Features in Reinforcement Learning with Neural Networks*, with David Filliat and Goran Frehse, accepted for publication in the *Transactions on Machine Learning Research (TMLR)*, 2024. Part III is organized as follows:

- In Chapter 10, we start by presenting features encoding in linear function approximation and the use of neural networks in deep RL to automatically learn features from raw data without prior knowledge. As highlighted in this chapter, although neural networks are universal approximators in theory, they suffer from some limitations in practice. These limitations include not only the number of parameters, as discussed in the last part, but also the amount of optimization that can be achieved in practice. We present experiments that indicate neural networks behave as under-parameterized models regularized through early stopping.

In particular, we observe this form of regularization induces a spectral bias, in which the fitting high-frequency components of the value function requires exponentially more gradient update steps than the low-frequency ones.

- In Chapter 11, to overcome the spectral bias and improve the learning of high-frequency components in RL, we suggest the use of two preprocessings based on the Fourier series for neural networks. The first preprocessing suggested is the Fourier Feature (FF) mapping, based on the Fourier series and introduced by Konidaris et al. (2011) for linear value function approximation. However, the major bottleneck of this Fourier preprocessing is that the dimension of the feature space grows exponentially with the dimension of the state space, which limits its use in high-dimensional problems. We propose a lighter, scalable version of the FF preprocessing called Fourier Light Features (FLF) to remedy this issue. In the following of this chapter, we present experiments indicating that the use of FF/FLF can lead to significant performance gains in terms of rewards and sample efficiency, and outperform other traditional preprocessings. We observe that both FLF and FF achieve similar performance, while FLF has fewer features than FF. Furthermore, we observe that such preprocessings increase the robustness with respect to hyperparameters.

- In Chapter 12, we empirically investigate the effects of the Fourier encodings on the learning process. In particular, we show that the proposed preprocessings lead to smoother neural networks, mitigate learning interference, promote sparsity, and increase the expressivity of learned features.

We conclude the thesis with a concise summary of our contributions and a discussion of future works.

1.2 Contributions

This thesis is divided into two distinct parts, each contributing a different aspect to the use of neural networks in Reinforcement Learning.

In Part II, we take a step towards a better theoretical understanding of the influence of the number of parameters and the l_2-regularization on the performance of Temporal Difference algorithms. Our main contributions can be summarized as follows:

- We propose a novel double asymptotic regime to study regularized LSTD with random features, where the number of features N and distinct visited states m go to infinity while maintaining a constant ratio. This leads to a precise assessment of the performance in both *over-parameterized* ($N/m > 1$) and *under-parameterized* regimes ($N/m < 1$).

- We identify the resolvent of a non-symmetric positive-definite matrix that emerges as a crucial factor in the performance analysis of TD learning algorithms in terms of the error functions and we provide its deterministic equivalent form in the double asymptotic regime.

- We derive analytical equations for the asymptotic empirical MSBE on the collected transitions, the asymptotic true MSBE, and the asymptotic MSVE and expose correction terms due to the constant ratio N/m. We show that the correction terms vanish as the l_2-regularization increases or N/m goes to infinity. We also show that the influence of the l_2-regularization parameter decreases as N/m increases.

- We show that the asymptotic errors studied can be expressed as the sum of the corresponding error terms of a regularized kernel LSTD predicator, with implicit l_2-regularization parameter $\tilde{\lambda}$ induced by the ratio N/m, and a second-order correction factor. Furthermore, we show that the second-order correction factors can be interpreted and linked to classical notions from non-parametric statistics, e.g., with the effective dimension.
- Our theory closely matches empirical results on a range of toy and small real-world Markov Reward Processes for any ratio N/m. In the phase transition around $N/m = 1$, we experimentally observe a peak in the Mean-Squared Bellman Error (MSBE) and the Mean-Squared Value Error (MSVE), i.e, a *double descent phenomenon* similar to what has been reported in supervised learning. We experimentally associate the correction terms found in our theoretical predictions with the double descent phenomenon. Correction terms, and therefore the difference between true and empirical MSBE, empirically vanish when the number of unvisited states reaches zero.

In Part III, we propose the use of a feature encoding based on the Fourier series as preprocessing for neural networks to improve performance. Our main contributions can be summarized as follows:

- While Fourier Features are standard in classic Reinforcement Learning, we suggest that Fourier Features are beneficial in kinematic observation-based RL problems with neural networks. We observe significant performance gains in both rewards and sample efficiency and extend the range of usable hyperparameters. In our experiments, Fourier Features outperform other common types of input preprocessing.
- We empirically investigate the effects of Fourier features on the learning process and show that Fourier features lead to smoother neural networks, mitigate learning interference, promote sparsity, and increase the expressivity of learned features.
- We propose a light, scalable version of Fourier Features to avoid the exponential explosion of traditional Fourier Features while maintaining much of their benefits.

Part I

Reinforcement Learning & Function Approximation

Chapter 2

Reinforcement Learning

Reinforcement Learning (RL) is a branch of Machine Learning in which an *agent* learns how to solve a task within an unknown environment by making sequential decisions based on its interactions with the *environment*. At each iteration t, the agent performs an *action* a_t based on its current situation described by the *state* s_t. One iteration later, at $t+1$, as a consequence of the action a_t, the agent receives a numerical reward r_{t+1} and transitions to a new state s_{t+1}. Agent/environment interactions are described in Figure 2.1. The reward r_{t+1} is a numeric feedback of the agent's performances after taking the action a_t in the state s_t. The objective of the agent is to maximize the numerical reward signal over time through trial and error, based on its past experiences and the feedback it receives from the environment. RL is particularly convenient as it enables learners to solve tasks through observed data without the need for a specific model or the explicit programming of rules or behaviors.

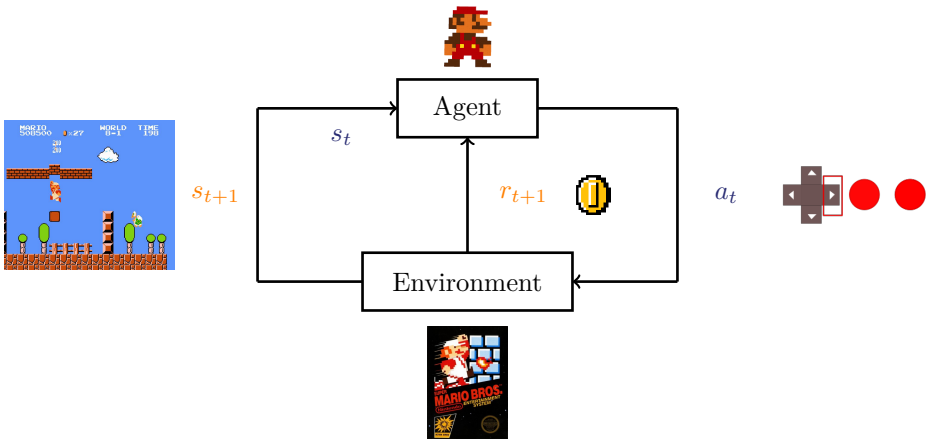


Figure 2.1: Description of the interaction with the *environment*: at time t, the *agent* is in **state** s_t and chooses the **action** a_t. The environment sends back a **reward** $r_{t+1}= R(s_t, a_t, s_{t+1})$ and a new state s_{t+1}, which will be used by the agent at time $t+1$.

In this chapter, we first recall the concept of a Markov Decision Process and the definitions of policies and value functions to mathematically describe the agent/environment interactions in Section 2.1. After introducing dynamic programming algorithms in Section 2.2, we then provide a brief overview of traditional tabular value-based algorithms, such as TD(0) and Q-learning in

Section 2.3. This chapter can be skipped by readers familiar with the RL terminology.

2.1 Mathematical Framework

2.1.1 Markov Decision Processes & Policy

Environments in RL are mathematically described by *Markov Decision Processeses* (MDPs) (Puterman, 2014; Bellman, 1957; Sutton and Barto, 2018). In the following, we denote by $\mathcal{P}(\mathcal{S})$ the space of probability distributions over the state space $\mathcal{S}$.

Definition 2.1.1 (Markov Decision Process)**.** *A MDP (Puterman, 2014; Bellman, 1957; Sutton and Barto, 2018) is defined by a tuple $(\mathcal{S}, \mathcal{A}, P, R, \mu_0)$ in which:*

- *$\mathcal{S}$ is the state space, which is measurable and may be finite or infinite;*
- *$\mathcal{A}$ is the action space, which is measurable and may be finite or infinite;*
- *$P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is the transition function (stochastic kernel) that captures the dynamics of the environment. With $P(s'|s, a)$ we indicate the probability of moving to state $s' \in \mathcal{S}$ from state $s \in \mathcal{S}$ after performing the action $a \in \mathcal{A}$ where*

$$P(s'|s, a) = Pr\big[s_{t+1} = s'|s_t = s, a_t = a\big].$$

- *$R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the bounded reward function. $R(s, a, s')$ depicts the immediate reward obtained when the agent in state s chooses an action a and moves to the state s'.*
- *$\mu_0 \in \mathcal{P}(\mathcal{S})$ is the initial state distribution. $\mu_0(s) = Pr\big[s_0 = s\big]$ depicts the probability of starting at state s.*

Remark 1. *While episodic Markov Decision Processes are also considered in the literature (Sutton, 1988) and without loss of generality, we focus here on non-terminating MDPs, where the agent interacts with its environment indefinitely.*

Remark 2. *The transition function $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ and the reward function $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ depend only on the current state and action, not on past states or actions. In this context, we say that we satisfy the Markov Assumption (Puterman, 2014).*

Remark 3. *In RL problems, the parameters of an MDP are usually assumed to be unknown. RL algorithms only learn from data collected from interactions with the MDP*

The behavior of the agent, or the action-selection strategy, is mathematically described by *policies*.

Definition 2.1.2 (Policy)**.** *A policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ maps each state in $\mathcal{S}$ to a probability distribution over actions $\mathcal{P}(\mathcal{A})$. In particular, $\pi(a|s) = Pr\big[a|s\big]$ denotes the probability of taking the action $a \in \mathcal{A}$ in the state $s \in \mathcal{S}$.*

Remark 4. *If for every state $s \in \mathcal{S}$ the associated distribution $\pi(\cdot|s)$ is deterministic, then the policy is said to be deterministic. In that case, we write the deterministic policy as a function $\pi : \mathcal{S} \to \mathcal{A}$.*

The reward function $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ formalizes the agent's objective, as the agent aims to maximize the cumulative sum of the received rewards.

Objective. RL algorithms aim at finding a policy that maximizes the total amount of rewards it receives and is mathematically described with the *expected return* defined as follows

$$J(\pi) = \mathbb{E}_{\mu_0,P,\pi}\left[\sum_{t=0}^{\infty}\gamma^t r_{t+1}\right] = \mathbb{E}_{\mu_0,P,\pi}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t,a_t,s_{t+1})\right], \tag{2.1}$$

where $r_{t+1} = R(s_t,a_t,s_{t+1})$ and the expectation is taken under $a_t \sim \pi(\cdot|s_t)$, $s_{t+1} \sim P(\cdot|s_t,a_t)$ and $s_0 \sim \mu_0$. $\gamma \in [0,1)$ is the *discount factor* and quantifies the preference for immediate rewards compared to delayed rewards. Complete indifference to the future corresponds to $\gamma = 0$. If $\gamma = 0$, the agent is "myopic" and only maximizes immediate rewards, i.e., its objective is to learn how to choose a_t to maximize only r_{t+1}. However, acting to maximize immediate reward can generally reduce access to future rewards, and leads to lower returns and poorer performance. $\gamma < 1$ guarantees $J(\pi)$ in equation 2.1 is finite.

2.1.2 Value Functions

The performance of an agent under a policy π can be described by its value function $V^\pi : \mathcal{S} \to \mathbb{R}$.

Definition 2.1.3 (Value Function)**.** *Given a state $s \in \mathcal{S}$ and policy π, we define the value function $V^\pi(s)$ at s as the expected total discounted amount of reward that the agent receives if it follows the policy π starting from the state s*

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_{P,\pi}\left[\sum_{t=0}^{\infty}\gamma^t r_{t+1} \;\middle|\; s_0 = s\right] \\ &= \mathbb{E}_{P,\pi}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t,a_t,s_{t+1}) \;\middle|\; s_0 = s\right], \end{aligned} \tag{2.2}$$

where $r_{t+1} = R(s_t,a_t,s_{t+1})$ and the expectation is taken under $a_t \sim \pi(\cdot|s_t)$ and $s_{t+1} \sim P(\cdot|s_t,a_t)$.

Remark 5. *We observe that the expected return $J(\pi)$ (equation 2.1) can be expressed with the value function $V^\pi(\cdot)$ as*

$$J(\pi) = \mathbb{E}_{\mu_0}\left[V^\pi(\boldsymbol{s})\right].$$

Remark 6. *From the Markov property, the value function is invariant to the starting time from which the cumulative rewards are considered. Indeed, for any $t_0 \geq 0$ and $s \in \mathcal{S}$:*

$$V^\pi(s) = \mathbb{E}_{P,\pi}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t,a_t,s_{t+1}) \;\middle|\; s_0 = s\right] = \mathbb{E}_{P,\pi}\left[\sum_{t=0}^{\infty}\gamma^t R(s_{t+t_0},a_{t+t_0},s_{t+t_0+1}) \;\middle|\; s_{t_0} = s\right].$$

For control purposes, instead of considering the value function for each state, it is more practical to consider a value function for each state-action pair using the *action-value function* $Q^\pi : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$.

Definition 2.1.4 (Action-Value Function)**.** *Given a state $s \in \mathcal{S}$, an action $a \in \mathcal{A}$ and a policy π, we define the action-value function $Q^\pi(s,a)$ for the state-action pair (s,a) as:*

$$\begin{aligned} Q^\pi(s,a) &= \mathbb{E}_{P,\pi}\left[\sum_{t=0}^{\infty}\gamma^t r_{t+1} \;\middle|\; s_0 = s, a_0 = a\right] \\ &= \mathbb{E}_{P,\pi}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t,a_t,s_{t+1}) \;\middle|\; s_0 = s, a_0 = a\right], \end{aligned} \tag{2.3}$$

where $r_{t+1} = R(s_t, a_t, s_{t+1})$ *and the expectation is taken under* $a_t \sim \pi(\cdot|s_t)$ *and* $s_{t+1} \sim P(\cdot|s_t, a_t)$.

2.1.3 Optimal Policy

Since the objective is to find a policy that maximizes the expected return, value functions can be used to define a partial ordering over policies. Indeed, a policy π' is said to be *better* than or equal to a policy π if its expected return is greater than or equal to that of π for all states, i.e.,

$$\pi' \geq \pi \quad \iff \quad V_{\pi'}(s) \geq V_\pi(s), \qquad \forall s \in \mathcal{S}. \tag{2.4}$$

If a policy is greater or equal to all the other policies, it is called an *optimal policy.*

Theorem 2.1.1 (Optimal Policy)**.** *For any MDP, there exists a deterministic optimal policy* $\pi^* : \mathcal{S} \to \mathcal{A}$ *(Bellman, 1952; Bertsekas et al., 2011; Feinberg, 2011; Agarwal et al., 2019).*

Although there may be more than one optimal policy, we denote all the optimal policies by π^*. All optimal policies share the same *optimal value function* $V^* : \mathcal{S} \to \mathbb{R}$ and *optimal action-value function* $Q^* : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and defined as follows:

$$V^*(s) = V^{\pi^*}(s) = \max_\pi V^\pi(s),$$
$$Q^*(s, a) = Q^{\pi^*}(s, a) = \max_\pi Q^\pi(s, a),$$

for all states $s \in \mathcal{S}$ and actions $a \in \mathcal{A}$. An example of deterministic optimal policy $\pi^* : \mathcal{S} \to \mathcal{A}$ can be found by maximizing over Q^* as

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} Q^*(s, a).$$

2.1.4 Bellman Operators

It is known that the value function V^π and the action-value function Q^π satisfy the following Bellman equations (Bellman, 1957):

$$\begin{aligned}
Q^\pi(s,a) &= \mathbb{E}_{P,\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) \;\middle|\; s_0 = s, a_0 = a\right] \\
&= \mathbb{E}_{P,\pi}\left[R(s_0, a_0, s_1) + \gamma \sum_{t=0}^{\infty} \gamma^t R(s_{t+1}, a_{t+1}, s_{t+2}) \;\middle|\; s_0 = s, a_0 = a\right] \\
&= \mathbb{E}_{s' \sim P(\cdot|s,a)}\Big[R(s, a, s')\Big] + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a), P, \pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_{t+1}, a_{t+1}, s_{t+2}) \mid s_1 = s'\right] \\
&= \bar{R}(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}\Big[V^\pi(s')\Big] \qquad \text{(Markov property)}
\end{aligned} \tag{2.5}$$

with $\bar{R}(s,a) = \mathbb{E}_{s' \sim P(\cdot|s,a)}\Big[R(s,a,s')\Big]$, and

$$\begin{aligned} V^{\pi}(s) &= \mathbb{E}_{P,\pi}\Big[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) \mid s_0 = s\Big] \\ &= \mathbb{E}_{a\sim\pi(\cdot|s)}\Big[\mathbb{E}_{P,\pi}\Big[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) \mid s_0 = s, a_0 = a\Big]\Big] \\ &= \mathbb{E}_{a\sim\pi(\cdot|s)}\Big[Q^{\pi}(s,a)\Big]. \end{aligned} \tag{2.6}$$

From these equations, we define the *Bellman operators*.

Definition 2.1.5 (Value Function Bellman Operator)**.** *Let π be a policy. The value function Bellman operator $\mathcal{T}_V^{\pi}$ of the policy π is defined for any function $V : \mathcal{S} \to \mathbb{R}$ as*

$$\big(\mathcal{T}_V^{\pi} V\big)(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}\Big[\bar{R}(s,a) + \gamma \mathbb{E}_{s'\sim P(\cdot|s,a)}\big[V(s')\big]\Big], \qquad \forall s \in \mathcal{S}. \tag{2.7}$$

Definition 2.1.6 (Action-Value Function Bellman Operator)**.** *Let π be a policy. The action-value function Bellman operator $\mathcal{T}_Q^{\pi}$ of the policy π is defined for any function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ as*

$$\big(\mathcal{T}_Q^{\pi} Q\big)(s,a) = \bar{R}(s,a) + \gamma \mathbb{E}_{s'\sim P(\cdot|s,a)}\Big[\mathbb{E}_{a'\sim\pi(\cdot|s')}\big[Q^{\pi}(s',a')\big]\Big], \qquad \forall s,a \in \mathcal{S}\times\mathcal{A}. \tag{2.8}$$

Bellman operators are γ-contracting with respect to the infinity norm l_∞ (Puterman, 2014; Agarwal et al., 2019), i.e.,

$$\begin{aligned} \|\mathcal{T}_V^{\pi} V - \mathcal{T}_V^{\pi} V'\|_\infty &\le \gamma \|V - V'\|_\infty, \\ \|\mathcal{T}_Q^{\pi} Q - \mathcal{T}_Q^{\pi} Q'\|_\infty &\le \gamma \|Q - Q'\|_\infty. \end{aligned}$$

According to the Banach fixed-point theorem, since $\mathcal{T}_V^{\pi}$ and $\mathcal{T}_Q^{\pi}$ are contractions, there are unique fixed-points V and Q such that $\mathcal{T}_V^{\pi} V = V$ and $\mathcal{T}_Q^{\pi} Q = Q$. From equation 2.5 and equation 2.6, V^{π} and Q^{π} satisfy the fixed-point equations of Bellman operators as

$$\begin{aligned} \mathcal{T}_V^{\pi} V^{\pi} &= V^{\pi}, \\ \mathcal{T}_Q^{\pi} Q^{\pi} &= Q^{\pi}. \end{aligned}$$

From Theorem 2.1.1, there always exists a deterministic optimal policy π^*. Therefore, using equation 2.5 and equation 2.6, we can derive Bellman equations for the optimal value function V^* and the optimal action-value function Q^*. In particular, for all states $s \in \mathcal{S}$ and actions $a \in \mathcal{A}$, we have

$$\begin{aligned} V^*(s) &= \max_{a\in\mathcal{A}}\Big\{\bar{R}(s,a) + \gamma \mathbb{E}_{s'\sim P(\cdot|s,a)}\big[V^*(s')\big]\Big\}, \\ Q^*(s,a) &= \bar{R}(s,a) + \gamma \mathbb{E}_{s'\sim P(\cdot|s,a)}\big[V^*(s')\big]. \end{aligned}$$

From the equations above, we can derive optimal Bellman operators that can also be shown as γ-contractions under the infinity norm l_∞. These operators ensure that V^* and Q^* satisfy the fixed-point equations.

Definition 2.1.7 (Optimal Value Function Bellman Operators)**.** *The optimal value function Bell-*

man operator $\mathcal{T}_V^*$ *is defined for any function* $V : \mathcal{S} \to \mathbb{R}$ *as*

$$\big(\mathcal{T}_V^* V\big)(s) = \max_{a \in \mathcal{A}} \Big\{ \bar{R}(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \big[V(s') \big] \Big\}, \qquad \forall s \in \mathcal{S}. \tag{2.9}$$

Definition 2.1.8 (Optimal Action-Value Function Bellman Operator)**.** *The optimal action-value function Bellman operator* $\mathcal{T}_Q^*$ *is defined for any function* $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ *as*

$$\big(\mathcal{T}_Q^* Q\big)(s,a) = \bar{R}(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)} \Big[\max_{a' \in \mathcal{A}} \big\{ Q^\pi(s',a') \big\} \Big], \qquad \forall s,a \in \mathcal{S} \times \mathcal{A}. \tag{2.10}$$

2.2 Dynamic Programming

In this section, we describe the dynamic programming (DP) approach to RL, in the case where the transition model $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is known to the agent (Bertsekas, 2012). Furthermore, we assume that the state and action spaces are finite. A common way to apply DP algorithms in continuous state and action spaces is to discretize them before applying them. The use of DP algorithms in RL is strongly limited because the transition model $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is often unknown, and because of the great computational expense with iterations overall the state space $\mathcal{S}$. Although not commonly used in practice, DP algorithms are still theoretically important and provide an essential foundation for understanding the methods presented in the following sections.

2.2.1 Policy Evaluation

In *policy evaluation* or *policy prediction*, the objective is to evaluate the performance of a policy π by estimating its value function $V^\pi : \mathcal{S} \to \mathbb{R}$. Since the dynamics $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ of the environment are known, we can repetitively compute the Bellman equation to get an increasingly accurate approximation of the value function. The initial approximation V_0 is chosen arbitrarily, and each successive approximation V_k is obtained using the Bellman operator (equation 2.7) as the update rule. In particular, the update rule for all $k \geq 1$ is given by

$$V_{k+1}(s) \leftarrow \big(\mathcal{T}_V^\pi V_k\big)(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \sum_{s' \in \mathcal{S}} P(s'|s,a) \big(R(s,a,s') + \gamma V(s') \big), \quad \forall s \in \mathcal{S}. \tag{2.11}$$

Exploiting successively the contraction property of the Bellman operator and the fact that $V^\pi = \mathcal{T}_V^\pi V^\pi$, we can show that the sequence (V_k) converges to the value function V^π. Indeed, for all $k \geq 1$, we have

$$\big\| V_k - V^\pi \big\|_\infty = \big\| \mathcal{T}_V^\pi V_{k-1} - \mathcal{T}_V^\pi V^\pi \big\|_\infty \leq \gamma \big\| V_{k-1} - V^\pi \big\|_\infty \leq \gamma^k \big\| V_0 - V^\pi \big\|_\infty.$$

A full description of policy evaluation is provided by Algorithm 1

2.2.2 Policy Iteration

With policy evaluation, we can estimate how "good" is a policy π. In the policy iteration, starting from a policy π, we seek to find a policy π' that is better than π, i.e., we want to find π' such that $\pi' \geq \pi$ as in equation 2.4. To achieve this, a popular approach is to use the following policy improvement theorem (Sutton and Barto, 2018).

Algorithm 1 Policy Evaluation (Sutton and Barto, 2018)

Input: π (the policy to be evaluated), ϵ (a small threshold determining the accuracy)
Output: Value-function V

```
V(s) ← 0   ∀s ∈ S
Δ ← ε + 1
while Δ > ε do
    for s ∈ S do
        v ← V(s)
        V(s) ← Σ_{a∈A} π(a|s) Σ_{s'∈S} P(s'|s,a)[R(s,a,s') + γV(s')]
        Δ ← min(Δ, |V(s) − v|)
    end for
end while
```

Theorem 2.2.1 (Policy Improvement Theorem)**.** *Let π and π' be any pair of deterministic policies such that, for all $s \in \mathcal{S}$,*

$$Q^{\pi}(s, \pi'(s)) \geq V^{\pi}(s).$$

Then, for all $s \in \mathcal{S}$, we have

$$V^{\pi'}(s) \geq V^{\pi}(s).$$

Proof. Let $s \in \mathcal{S}$. We have

$$\begin{aligned}
V^{\pi}(s) &\leq Q^{\pi}(s, \pi'(s)) \\
&= \mathbb{E}_{P,\pi}\left[R(s_0, a_0, s_1) + \gamma \sum_{t=0}^{\infty} \gamma^t R(s_{t+1}, a_{t+1}, s_{t+2}) \;\middle|\; s_0 = s, a_0 = \pi'(s)\right] \\
&= \mathbb{E}_{P,\pi'}\left[R(s_0, a_0, s_1) + \gamma V^{\pi}(s_1) \;\middle|\; s_0 = s\right] \\
&\leq \mathbb{E}_{P,\pi'}\left[R(s_0, a_0, s_1) + \gamma Q^{\pi}(s_1, \pi'(s_1)) \;\middle|\; s_0 = s\right] \\
&= \mathbb{E}_{P,\pi'}\left[R(s_0, a_0, s_1) + R(s_1, a_1, s_2) + \gamma V^{\pi}(s_2) \;\middle|\; s_0 = s\right] \\
&\;\;\vdots \\
&\leq V^{\pi'}(s).
\end{aligned}$$

□

A common method for obtaining a better policy π' is by acting *greedily* on the current policy π, i.e., by selecting the best action for every state according to the current value of the action-value function $Q^{\pi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, as follows

$$\pi'(s) = \arg\max_{a \in \mathcal{A}} Q^{\pi}(s, a) = \arg\max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s'|s, a)\big[R(s, a, s') + \gamma V(s')\big], \quad \forall s \in \mathcal{S}.$$

For the greedy policy π', the condition of Theorem 2.2.1 is satisfied since for all states $s \in \mathcal{S}$ we have

$$Q^{\pi}(s, \pi'(s)) = \max_{a \in \mathcal{A}} Q^{\pi}(s, a) \geq Q^{\pi}(s, \pi(s)) = V^{\pi}(s).$$

Algorithm 2 Policy Iteration (Sutton and Barto, 2018)

Input: ϵ (a small threshold determining the accuracy)
Output: Policy π
$\pi(s) \in \mathcal{A}$ arbitrarily for all $s \in \mathcal{S}$
$V(s) \leftarrow 0 \quad \forall s \in \mathcal{S}$
$\Delta \leftarrow \epsilon + 1$
stable $\leftarrow$ *false*
while not *stable* **do**

1. Policy Evaluation
while $\Delta > \epsilon$ **do**
for $s \in \mathcal{S}$ **do**
$v \leftarrow V(s)$
$V(s) \leftarrow \sum_{a \in \mathcal{A}} \pi(a|s) \sum_{s' \in \mathcal{S}} P(s'|s,a)\big(R(s,a,s') + \gamma V(s')\big)$
$\Delta \leftarrow \min\big(\Delta, |V(s) - v|\big)$
end for
end while

2. Policy Improvement
stable $\leftarrow$ *true*
for $s \in \mathcal{S}$ **do**
old_action $\leftarrow \pi(s)$
$\pi(s) \leftarrow \arg\max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s'|s,a)\big[R(s,a,s') + \gamma V(s')\big]$
If *old_action* $\neq \pi(s)$, then *stable* $\leftarrow$ *false*
end for
end while

Note that generating the greedy policy π from the policy π is equivalent to applying the Bellman optimal operator to the value function V^π. The greedy operation combined with policy evaluation generates a sequence of monotonically policy improvements:

$$\pi_0 \xrightarrow{E} V^{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} V^{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \cdots \xrightarrow{I} \pi^* \xrightarrow{E} V^*,$$

where E $\xrightarrow{E}$ denotes a policy evaluation and $\xrightarrow{I}$ denotes a policy improvement. Each policy is guaranteed to be a strict improvement over the previous one (unless it is already optimal). This process is called *policy iteration*, and its complete pseudo-code is given by Algorithm 2.

2.2.3 Value Iteration

The challenge with policy iteration is that each iteration includes a policy evaluation step and requires multiple sweeps through the state space $\mathcal{S}$. This step can be expensive and unnecessary since the policy evaluation can be truncated without negatively affecting the policy iteration step (Sutton and Barto, 2018). The *Value Iteration* algorithm combines the policy iteration and policy evaluation step into one step by constructing a sequence (V_k) where the initial approximation V_0 is chosen arbitrarily, and each successive approximation is obtained by using the optimal Bellman operator (equation 2.9) for V_k as an update rule. Therefore, the update rule for all $k \geq 1$ is given by

$$V_{k+1} \leftarrow \mathcal{T}_V^* V_k = \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s'|s,a)\big(R(s,a,s') + \gamma V(s')\big), \quad \forall s \in \mathcal{S}.$$

Algorithm 3 Value Iteration (Sutton and Barto, 2018)

Input: π (the policy to be evaluated), ϵ (a small threshold determining the accuracy)
Output: Deterministic policy π such that

$$\pi(s) = \arg\max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s'|s,a)\big(R(s,a,s') + \gamma V(s')\big)$$

$V(s) \leftarrow 0 \quad \forall s \in \mathcal{S}$
$\Delta \leftarrow \epsilon + 1$
while $\Delta > \epsilon$ **do**
  for $s \in \mathcal{S}$ **do**
    $v \leftarrow V(s)$
    $V(s) \leftarrow \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} P(s'|s,a)\big(R(s,a,s') + \gamma V(s')\big)$
    $\Delta \leftarrow \min\big(\Delta, |V(s) - v|\big)$
  end for
end while

Exploiting successively the fact that $V^* = \mathcal{T}_V^* V^\pi$ and the contraction property of the optimal Bellman operator, we can show the sequence (V_k) converges to the value function V^*. Indeed, for all $k \geq 1$, we have

$$\big\|V_k - V^*\big\|_\infty = \big\|\mathcal{T}_V^* V_{k-1} - \mathcal{T}_V^* V^*\big\|_\infty \leq \gamma \big\|V_{k-1} - V^*\big\|_\infty \leq \gamma^k \big\|V_0 - V^*\big\|_\infty.$$

A pseudo-code of Value Iteration is reported in Algorithm 3.

2.3 Tabular Reinforcement Learning Algorithms

The main drawback of the DP algorithm stems from the assumption that both the transition model $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ and the reward function $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ are known. However, in most RL problems, this information is not available making the use of these algorithms often impractical as the Bellman operator cannot be computed explicitly and must instead be estimated.

Model-based vs model-free. Two strategies can be considered to solve an RL problem: the *model-based* approach and the *model-free* approach. *Model-based* algorithms first learn a model of the transition function $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$. Based on this approximation, they attempt to find the corresponding optimal value function or the optimal policy (Moerland et al., 2023), e.g., with dynamic programming algorithms. In contrast, *model-free* approaches do not require an explicit formulation or approximation of the model and just rely on learning the optimal policy and/or value functions from interactions with the environment.

On-policy vs off-policy. The optimal solution of an MDP can be learned using two paradigms: on-policy and off-policy learning. *On-policy* methods evaluate and improve the policy used to make decisions and generate data. In contrast, *off-policy* algorithms aim to evaluate and improve a target policy that differs from the behavior policy used to interact with the environment.

In this section, we present some popular value-based algorithms that rely on estimates of value functions to compute the optimal policy in MDPs with finite state and action spaces. Temporal Difference (TD) learning algorithms presented in Section 2.3.1 are popular value-based algorithms

Algorithm 4 The On-line TD(0) Learning Algorithm (Sutton and Barto, 2018)

Input: π (the policy to be evaluated), $(\alpha_n)_{n\geq 0}$ (sequence of learning rates), T (number of steps)
Output: Value function V
  $V(s) \leftarrow 0 \quad \forall s \in \mathcal{S}$
  Initialize $s \sim \mu_0$
  for each step $t \in [T]$ **do**
    Choose action $a \sim \pi(\cdot|s)$
    Take action a, observe next state s' and reward $r = R(s, a, s')$
    $V(s) \leftarrow V(s) + \alpha_t\big(r + \gamma V(s')\big)$
    $s \leftarrow s'$
  end for

for policy evaluation, i.e., to estimate the value-function V^π of a given policy π. TD learning algorithms must be combined with a policy improvement process to compute an optimal policy. They form the basis of widely used algorithms in practice, among which SARSA presented in Section 2.3.2 and Q-learning introduced in Section 2.3.3.

2.3.1 Temporal Difference Learning

Temporal Difference (TD) learning algorithms (Sutton and Barto, 2018) execute the current policy π and update the estimation of its value function $V^\pi : \mathcal{S} \to \mathbb{R}$ at every interaction (see Algorithm 4). At each timestep t, the agent, in state s_t, takes an action a_t according to its current policy π, moves to the state s_{t+1} and receives the reward $r_{t+1} = R(s_t, a_t, s_{t+1})$ from the environment. The update rule of the simplest TD learning method, known as TD(0) (Sutton and Barto, 2018), is given for all states $s \in \mathcal{S}$ by

$$V(s) \leftarrow V(s) + \mathbf{1}_{s=s_t}\alpha_t\big(r_{t+1} + \gamma V(s_{t+1}) - V(s_t)\big), \tag{2.12}$$

where $\alpha_t \in [0, 1]$ is the *stepsize* or *learning rate*[1], $r_{t+1} + \gamma V(s_{t+1})$ is the *TD target* and $\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$ is the *TD error*. This TD method is called TD(0), or one-step TD, because it is a special case of the TD(λ) and n-step TD methods (Sutton and Barto, 2018).

Remark 7. *To highlight the proximity with the update of equation 2.11, we can rewrite the update of equation 2.12 as*

$$V(s) \leftarrow \big(1 - \mathbf{1}_{s=s_t}\alpha_t\big)V(s) + \mathbf{1}_{s=s_t}\alpha_t\big(r_{t+1} + \gamma V(s_{t+1})\big), \quad \forall s \in \mathcal{S}. \tag{2.13}$$

In particular, if $\alpha_t = 1$ and the MDP is deterministic, then equation 2.13 is just the update of equation 2.11 for the state s_t. Otherwise, equation 2.13 is an exponential average, and the temporal-difference update provides an estimate of its expectation.

TD(0) is considered a *bootstrapping method* because it updates estimates using targets $r_{t+1} + \gamma V(s_{t+1})$, which are derived from current value estimates V. Indeed, from Bellman equation 2.6 and equation 2.5, we have

$$V^\pi(s) = \mathbb{E}_{P,\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid s_0 = s\right] \tag{2.14}$$

$$= \mathbb{E}_{P,\pi}\Big[r_{t+1} + \gamma V^\pi(s_{t+1}) \mid s_0 = s\Big]. \tag{2.15}$$

[1]For the sake of simplicity, we present learning rates that only depend on time. Other approaches, like dependencies on states and actions, can be found in Sutton and Barto (2018)

Algorithm 5 SARSA (Rummery and Niranjan, 1994)

Input: π (the policy to be evaluated), $(\alpha_n)_{n\geq 0}$ (sequence of learning rates), T (number of steps)
Output: Action-value function Q
  $Q(s,a) \leftarrow 0 \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}$
  Initialize $s \sim \mu_0$
  Choose action $a \sim \pi(\cdot|s)$
  for each step $t \in [T]$ **do**
    Take action a, observe next state s' and reward $r = R(s,a,s')$
    Choose action $a \sim \pi(\cdot|s')$
    $Q(s,a) \leftarrow Q(s,a) + \alpha_t\big(r + \gamma Q(s',a') - Q(s,a)\big)$
    $s \leftarrow s', a \leftarrow a'$
  end for

The TD target is an estimate for both reasons: it samples the expected values in equation 2.15 and uses the current estimate V instead of the true V^π. The TD(0) algorithm can be shown to converge to V^π if learning rates satisfy the Robbins-Monro conditions (Robbins and Monro, 1951):

$$\sum_{t=1}^{\infty} \alpha_t = \infty \quad \sum_{t=1}^{\infty} \alpha_t^2 < \infty. \tag{2.16}$$

TD learning algorithms can also be used to estimate the action-value function $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, which is more convenient for policy improvement in control tasks. In the following of this section, we review two well-known TD(0) approaches to estimate the action-value function $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ in Section 2.3.2 and the optimal action-value function $Q^* : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ in Section 2.3.3.

2.3.2 SARSA

SARSA (Rummery and Niranjan, 1994) is an on-policy TD learning algorithm that uses the TD error to update the action-value function $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. At each timestep t, the agent, in state s_t, takes an action a_t according to its current policy π, moves to s_{t+1} and receives the reward $r_{t+1} = R(s_t, a_t, s_{t+1})$ from the environment. The update rule of SARSA for all state-action pairs $(s,a) \in \mathcal{S} \times \mathcal{A}$ is given by

$$Q(s,a) \leftarrow Q(s,a) + \mathbf{1}_{\{s=s_t, a=a_t\}} \alpha_t \big(r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\big), \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}, \tag{2.17}$$

where $a_{t+1} \sim \pi(\cdot \mid s_t)$ and $\alpha_t \in [0,1]$ is the learning rate. The use of the quintuple of events $(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1})$ gives rise to the name SARSA. The convergence properties of SARSA depend on the policies used. Singh et al. (2000) prove that SARSA converges to the action-value function $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ under the assumption that the collected rewards are bounded, the agent selects actions so as to visit every (s,a) pair infinitely often, and the sequence of learning rates $(\alpha_t)_t$ satisfy the Robbins-Monro conditions (equation 2.16). The pseudocode of SARSA can be found in Algorithm 5.

2.3.3 Q-Learning

The Q-learning algorithm (Watkins and Dayan, 1992) is one of the most popular RL algorithms. It is an off-policy TD learning algorithm. At each timestep t, the agent, in state s_t, takes an action a_t according to its current policy π, moves to s_{t+1} and receives the reward $r_{t+1} = R(s_t, a_t, s_{t+1})$

Algorithm 6 Q-learning (Watkins and Dayan, 1992)

Input: $(\alpha_n)_{n\geq 0}$ (sequence of learning rates), T (number of steps)
Output: Q (estimation of the optimal action-value function Q^*)
  $Q(s,a) \leftarrow 0 \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}$
  Initialize $s \sim \mu_0$
  for each step $t \in [T]$ **do**
    With probability ϵ select a random action a otherwise select $a = \arg\max_{a' \in \mathcal{A}} Q(s,a')$
    Take action a, observe next state s' and reward $r = R(s,a,s')$
    $Q(s,a) \leftarrow Q(s,a) + \alpha_t\big(r + \gamma \max_{a' \in \mathcal{A}} Q(s',a') - Q(s,a)\big)$
    $s \leftarrow s'$
  end for

from the environment. The update rule for all state-action pairs $(s,a) \in \mathcal{S} \times \mathcal{A}$ is given by

$$Q(s,a) = Q(s,a) + \mathbf{1}_{\{s=s_t, a=a_t\}} \alpha_t \big(r_{t+1} + \gamma \max_{a' \in \mathcal{A}} Q(s_{t+1}, a') - Q(s_t, a_t)\big), \quad \forall s, a \in \mathcal{S} \times \mathcal{A},$$

where $\alpha_t \in [0,1]$ is the learning rate. Since the Q-learning update rule does not consider the policy used to collect rewards, the algorithm is off-policy. The difference with SARSA lies in the fact that the Q-learning algorithm approximates the optimal action-value function $Q^* : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ at each iteration with a sample version of the optimal Belman operator. Watkins and Dayan (1992) prove that the Q-learning algorithm converges to the optimal action-value function $Q^* : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, under the assumption that the collected rewards are bounded, the agent selects actions so as to visit every (s,a) pair infinitely often, and the learning rates $(\alpha_t)_t$ satisfy the Robbins-Monro conditions (equation 2.16). The success of this algorithm is mainly due to its simplicity. The pseudocode of Q-Learning is presented in Algorithm 6.

Chapter 3

Function Approximation in Value-Based Algorithms

In the previous chapter, we assumed that the state space $\mathcal{S}$ and action space $\mathcal{A}$ were finite. This assumption enabled the use of lookup tables to represent value functions and policies, in which a unique distinct value is assigned for each state or state-action pair. While this approach has strong theoretical foundations and is effective in MDPs with finite spaces, it encounters significant limitations in real-world scenarios that often involve large or infinite state and action spaces. Indeed, the amount of memory required to store the lookup table and the number of samples required to learn an optimal policy increase exponentially with the dimensions of the problem. This challenge is commonly called the "*curse of dimensionality*". One solution is to use function approximation methods within RL algorithms to approximate the value functions or policies. Typically, the number of parameters is significantly lower than the number of states, and the variation of a single parameter affects the estimated values of many states. Such *generalization* makes the learning potentially more powerful but challenging to manage and understand.

In this chapter, we discuss function approximation methods for value-based algorithms, which are used to approximate value functions. In Section 3.1, we introduce the concept of the Markov Reward Process, which is the mathematical framework considered in value function approximation for describing the behavior of the agent in its environment. In Section 3.2, we present different objective functions that value-based algorithms consider to approximate value functions. In Section 3.3, we present a *linear function approximation* approach considering stochastic gradient-based approaches to approximate value functions. In Section 3.4, we explore how value-based algorithms can be extended to neural networks through the example of the Deep Q-Network (DQN) algorithm.

3.1 Markov Reward Processes

In value function approximation, the behavior of a fixed policy π within an MDP is often described by a Markov Reward Process (MRP).

Definition 3.1.1 (Markov Reward Process)**.** *For a given policy π in a MDP $(\mathcal{S}, \mathcal{A}, P, R, \mu_0)$, the corresponding Markov Reward Process is defined by the tuple $(\mathcal{S}, P^\pi, R^\pi, \mu_0)$ where:*

- $\mathcal{S}$ *is the state space of the MDP, which is measurable and may be finite or infinite;*
- $P^\pi : \mathcal{S} \to \mathcal{P}(\mathcal{S})$ *is the transition function (stochastic kernel) that captures the behavior of the policy* π *in the environment. With* $P^\pi(s'|s)$*, we indicate the probability of moving to state* $s' \in \mathcal{S}$ *from state* $s \in \mathcal{S}$ *after choosing an action* a *from* $\pi(\cdot|s)$ *where*

$$P^\pi(s'|s) = \mathbb{E}_{a\sim\pi(\cdot|s)}\big[P\big(s'|s,a\big)\big];$$

- $R^\pi : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ *is a bounded reward function.* $R^\pi(s,s')$ *depicts the immediate reward obtained when the agent in state* s *moves to the state* s'*. It is defined as*

$$R^\pi(s,s') = \mathbb{E}_{a\sim\pi(\cdot|s)}\big[P\big(s'|s,a\big)R(s,a,s')\big];$$

- $\mu_0 \in \mathcal{P}(\mathcal{S})$ *is the initial state distribution of the MDP.*

The value function $V^\pi : \mathcal{S} \to \mathbb{R}$ of a given MRP $(\mathcal{S}, P^\pi, R^\pi, \mu_0)$ is the value function V^π defined in equation 2.2 as

$$V^\pi(s) = \mathbb{E}_{P^\pi}\left[\sum_{t=0}^{\infty}\gamma^t R^\pi(s_t, s_{t+1}) \;\middle|\; s_0 = s\right] \quad \forall s \in \mathcal{S}.$$

In this chapter, we focus on value-based algorithms that approximate the value function $V^\pi : \mathcal{S} \to \mathbb{R}$ of a given MRP $(\mathcal{S}, P^\pi, R^\pi, \mu_0)$ with a parameterized function $V_{\boldsymbol{\theta}} : \mathcal{S} \to \mathbb{R}$ of parameters $\boldsymbol{\theta}$. In the following section, we present different objective functions that can be considered to estimate $V^\pi : \mathcal{S} \to \mathbb{R}$. It is important to note that this approach can also be extended to approximate the action-value function $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$.

3.2 Objective Functions

In value function approximation, the objective is to find parameters $\boldsymbol{\theta}$ that yield a value function $V_{\boldsymbol{\theta}} : \mathcal{S} \to \mathbb{R}$ as close as possible to the value function $V^\pi : \mathcal{S} \to \mathbb{R}$. Since the number of parameters is typically significantly lower than the number of states, making the approximation $V_{\boldsymbol{\theta}}(s)$ more accurate for one specific state $s \in \mathcal{S}$ invariably means making the estimates for other states less accurate. Therefore, it is necessary to include a state distribution μ within the objective functions to determine which states should be prioritized by the value function approximation. Usually, we consider the *stationary distribution* μ^π of the MRP $(\mathcal{S}, P^\pi, R^\pi, \mu_0)$, which naturally weights states according to their long-term occupancy probabilities under the policy π.

Mean-Squared Value Error. Since we are interested in estimating parameters $\boldsymbol{\theta}$ that yield a value function $V_{\boldsymbol{\theta}} : \mathcal{S} \to \mathbb{R}$ as close as possible to the true value function $V^\pi : \mathcal{S} \to \mathbb{R}$, a natual objective function is the *Mean-Squared Value error* (MSVE), defined as

$$\text{MSVE}(\boldsymbol{\theta}) = \mathbb{E}_{s\sim\mu^\pi}\Big[\big(V^\pi(s) - V_{\boldsymbol{\theta}}(s)\big)^2\Big]. \tag{3.1}$$

In supervised learning problems, we typically have access to predictions of the target function, which is not the case in RL, where we only have access to rewards collected by the agent. Although

the value function $V^\pi : \mathcal{S} \to \mathbb{R}$ can be estimated using Monte-Carlo (MC) estimates, this approach requires a large number of samples and often results in high variance. The high variance of MC estimates makes their use in the MSVE objective function inefficient in practical applications.

Mean-Squared Bellman Error. The solution is to consider a new objective function, in which the value function V^π can be approximated with fewer samples and more efficiently than with MC estimates. A solution is to use bootstrapping (Sutton, 1988), where $V^\pi : \mathcal{S} \to \mathbb{R}$ is approximated with a one-step TD target using the approximated value-function $V_{\boldsymbol{\theta}}$. This new objective function called the *Mean-Squared Bellman error* (MSBE) is defined as

$$\text{MSBE}(\boldsymbol{\theta}) = \mathbb{E}_{s\sim\mu^\pi}\Big[\big(\mathcal{T}_V^\pi V_{\boldsymbol{\theta}}(s) - V_{\boldsymbol{\theta}}(s)\big)^2\Big], \tag{3.2}$$

where the Bellman operator $\mathcal{T}_V^\pi$ is defined for any function $V : \mathcal{S} \to \mathbb{R}$ as in equation 2.7 as

$$\big(\mathcal{T}_V^\pi V\big)(s) = \bar{R}^\pi(s) + \gamma \mathbb{E}_{s'\sim P^\pi(\cdot|s)}\big[V(s')\big], \tag{3.3}$$

with $\bar{R}^\pi(s) = \mathbb{E}_{s'\sim P^\pi(\cdot|s)}\big[R^\pi(s,s')\big]$ for all states $s \in \mathcal{S}$. While the MSVE directly compares the approximated value function $V_{\boldsymbol{\theta}} : \mathcal{S} \to \mathbb{R}$ with value function $V^\pi : \mathcal{S} \to \mathbb{R}$, the MSBE leverages the Bellman equation to quantify how closely $V_{\boldsymbol{\theta}}$ approaches its unique-fixed point solution V^π.

Mean-Squared Projected Bellman Error. In the MSBE, the result $\mathcal{T}_V^\pi V_{\boldsymbol{\theta}}(s)$ may not belong to the space of functions $\mathcal{V}$ represented by parameterized functions. Therefore, the minimum of the MSBE may not be solved with function approximation. To address this issue, the *Mean-Squared Projected Bellman Error* (MSPBE) propose computing the squared distance between $V_{\boldsymbol{\theta}} : \mathcal{S} \to \mathbb{R}$ and the closest function of $\mathcal{T}_V^\pi V_{\boldsymbol{\theta}}$ that does lie in $\mathcal{V}$. This is formalized as

$$\text{MSPBE}(\boldsymbol{\theta}) = \mathbb{E}_{s\sim\mu^\pi}\Big[\big(\Pi\mathcal{T}_V^\pi V_{\boldsymbol{\theta}}(s) - V_{\boldsymbol{\theta}}(s)\big)^2\Big], \tag{3.4}$$

where Π is a projection operator defined as

$$\Pi f = \min_{f_{\boldsymbol{\theta}}\in\mathcal{V}} \mathbb{E}_{s\sim\mu^\pi}\big[f_{\boldsymbol{\theta}}(s) - f(s)\big] \tag{3.5}$$

which projects arbitrary functions f onto the space of representable functions $\mathcal{V}$. Finding $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \text{MSPBE}(\boldsymbol{\theta})$ can be solved indirectly by solving the following nested optimization problem, which includes minimizing the projection error and the fixed-point error (Antos et al., 2008; Farahmand et al., 2008)

$$\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}} \mathbb{E}_{s\sim\mu^\pi}\Big[\big(\Pi\mathcal{T}_V^\pi V_{\boldsymbol{\theta}^*}(s) - V_{\boldsymbol{u}}(s)\big)^2\Big] \quad \text{(projection error)} \tag{3.6}$$

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{s\sim\mu^\pi}\Big[\big(V_{\boldsymbol{u}^*}(s) - V_{\boldsymbol{\theta}}(s)\big)^2\Big] \quad \text{(fixed-point error).} \tag{3.7}$$

In the projection error, we approximate the Bellman operator applied to the value function $V_{\boldsymbol{\theta}^*}$ with $V_{\boldsymbol{u}}$. In the fixed-point problem, we reduce the distance between both parameter estimates $\boldsymbol{u}^*$ and $\boldsymbol{\theta}^*$. Many RL algorithms solve this problem by alternating between improving the operator and fixed-point error (Dann et al., 2014). The MSPBE is easier to optimize but loses the direct connection to the original MSVE with the projection operator.

3.3 Linear Value Function Approximation using Gradient Based Approach

One of the simplest method used for function approximation is linear function approximation. In linear value function approximation, the value function $V^\pi : \mathcal{S} \to \mathbb{R}$ is approximated by a parameterized function $V_{\boldsymbol{\theta}} : \mathcal{S} \to \mathbb{R}$ defined for all states $s \in \mathcal{S}$ as

$$V_{\boldsymbol{\theta}}(s) = \boldsymbol{\sigma}(s)^T \boldsymbol{\theta} = \sum_{i=1}^{N} \boldsymbol{\theta}_i \boldsymbol{\sigma}_i(s), \tag{3.8}$$

where $\boldsymbol{\theta} \in \mathbb{R}^N$ is the *parameter vector* and $\boldsymbol{\sigma}(s)$ denotes the *features* of the state s. The feature map $\boldsymbol{\sigma} : \mathcal{S} \to \mathbb{R}^N$ reduces the number of parameters from $|\mathcal{S}|$ to N with $N \ll |\mathcal{S}|$, but comes at the price of less precision. The choice of the feature representation $\boldsymbol{\sigma} : \mathcal{S} \to \mathbb{R}^N$ is always a trade-off between compactness and expressiveness to get $V_{\boldsymbol{\theta}}(s) \approx V(s)$ for all states $s \in \mathcal{S}$. While the feature map $\boldsymbol{\sigma}(\cdot)$ is fixed during the training, the parameter vector $\boldsymbol{\theta}$ is adjusted during the learning process to get $V_{\boldsymbol{\theta}} \approx V^\pi$. Features $\{\boldsymbol{\sigma}_i(\cdot)\}_{i=1}^N$ determine the space of functions $\mathcal{V}$ that can be represented by linear function approximation, whereas the parameter vector $\boldsymbol{\theta}$ defines the function $V_{\boldsymbol{\theta}} \in \mathcal{V}$. The problem of learning a function approximator from a static training set of i.i.d. input/output samples over which multiple passes are made has been extensively studied in supervised learning. However, the problem is more complex in RL, making most of the function approximation algorithms developed in supervised learning ineffective. This complexity arises from the online learning process, wherein the agent interacts with its environment without having direct access to predictions of the target V^π. To consider the online learning process, several approaches rely on the use of *stochastic gradient descent* (SGD) to minimize their objective function.

Stochastic Gradient Descent. In function approximation considering parameterized functions, stochastic gradient descent (SGD) is commonly used on loss functions of the form $\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{x\sim p}[l(x;\boldsymbol{\theta})]$, for which the distribution p is independent of $\boldsymbol{\theta}$. In standard gradient descent, the parameter update is given by

$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \alpha_t \nabla \mathcal{L}(\boldsymbol{\theta}_t) = \boldsymbol{\theta}_t - \alpha_t \nabla \mathbb{E}_{x\sim p}[l(x;\boldsymbol{\theta}_t)]. \tag{3.9}$$

where α_k denotes the learning rate at timestep t. In standard gradient descent, the gradient is calculated with the expectative value of $l(x;\boldsymbol{\theta}_t)$, whereas stochastic gradient descent evaluates the gradient using just one sample x_t as follows

$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \alpha_t \nabla l(x_t;\boldsymbol{\theta}_t) \quad \text{with } x_t \sim p.$$

Parameters $(\boldsymbol{\theta}_t)_t$ updated with the stochastic gradient update rule are guaranteed to converge to a local minimum $\boldsymbol{\theta}^*$ of $\mathcal{L}(\boldsymbol{\theta})$ for learning rates $(\alpha_t)_t$ satisfying the Robbins-Monro conditions given by equation 2.16 (Robbins and Monro, 1951).

Linear Value Function Approximation with SGD. The SGD update rule can be used for finding parameters $\boldsymbol{\theta}$ in linear value function approximation to minimize the Mean-Squared Value Error (equation 3.1) where

$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \alpha_t \nabla \big(V^\pi(s_t) - V_{\boldsymbol{\theta}_t}(s_t)\big)^2 = \boldsymbol{\theta}_t + 2\alpha_t \big(V^\pi(s_t) - \boldsymbol{\theta}^T \boldsymbol{\sigma}(s_t)\big)\boldsymbol{\sigma}(s_t). \tag{3.10}$$

If $V^\pi(s_t)$ is replaced by an unbiased estimate T_t for which $\mathbb{E}\big[T_t|s_t = s\big] = V^\pi(s_t)$, then $\boldsymbol{\theta}_t$ is guaranteed to converge to a local optimum under the Robbins-Monro conditions (equation 2.16). An example of unbiased estimates is the Monte-Carlo estimates.

Semi-Gradient TD Learning Algorithms. In the popular linear TD(0) algorithm proposed by Sutton and Barto (2018), the target $V^\pi(s_t)$ in equation 3.10 is replaced by its TD target $\mathcal{T}_V^\pi V_{\boldsymbol{\theta}_t}(s_t)$ as

$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t + 2\alpha_t\big(\mathcal{T}_V^\pi V_{\boldsymbol{\theta}_t}(s_t) - V_{\boldsymbol{\theta}_t}(s_t)\big)\nabla V_{\boldsymbol{\theta}_t}(s_t) = \boldsymbol{\theta}_t + 2\alpha_t\big(\mathcal{T}_V^\pi \boldsymbol{\theta}^T\boldsymbol{\sigma}(s_t) - \boldsymbol{\theta}^T\boldsymbol{\sigma}(s_t)\big)\boldsymbol{\sigma}(s_t). \quad (3.11)$$

As highlighted by Barnard (1993), bootstrapping methods are not considered as true gradient descent algorithms as they only take into account the effect of changing the weight vector $\boldsymbol{\theta}_t$ on the standard estimate part, without considering its change on the target. Furthermore, the target $\mathcal{T}_V^\pi V_{\boldsymbol{\theta}_t}(\cdot)$ is not fixed and changes over time. Because bootstrapping methods only include a part of the gradient, they are classified as *semi-gradient methods*. In practice, we prefer using for the unbiased TD targets as

$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t + 2\alpha_t\delta_t\nabla V_{\boldsymbol{\theta}_t}(s_t) = \boldsymbol{\theta}_t + 2\alpha_t\big(r_{t+1} + \gamma\boldsymbol{\theta}^T\boldsymbol{\sigma}(s_{t+1}) - \boldsymbol{\theta}^T\boldsymbol{\sigma}(s_t)\big)\boldsymbol{\sigma}(s_t), \quad (3.12)$$

where $\delta_t = r_{t+1} + \gamma V_{\boldsymbol{\theta}_t}(s_{t+1}) - V_{\boldsymbol{\theta}_t}(s_t)$ is the *TD error*. This update rule can also be extended to TD(λ) or n-step TD methods (Sutton and Barto, 2018). It has been shown that semi-gradient linear TD learning algorithms are guaranteed to converge to the unique fixed-point solution of the Mean-Squared Projected Bellman error (equation 3.4) in the on-policy setting for ergodic MRPs (Tsitsiklis and Van Roy, 1996; Sutton, 1988). If the value function is estimated in the off-policy setting, the convergence towards $\boldsymbol{\theta}^*$ is not guaranteed anymore, and we can easily find examples for which TD learning algorithms diverge (Baird, 1995).

Remark 8. *We can observe that the update rule of the semi-gradient linear TD(0) given by equation 3.12 is similar to the update rule of equation 2.12 for the tabular TD(0).*

Remark 9. *In contrast to semi-gradient algorithms, the residual-gradient (RG) algorithm (Baird, 1995) takes into account the function approximator in the target when computing the gradient of the learning. In particular, RG algorithms aim to minimize the Mean-Squared Bellman using the following stochastic gradient update*

$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t + 2\alpha_t\big(r_{t+1} + \gamma\boldsymbol{\theta}^T\boldsymbol{\sigma}(s_{t+1}) - \boldsymbol{\theta}^T\boldsymbol{\sigma}(s_t)\big)\big(\boldsymbol{\sigma}(s_t) - \gamma\boldsymbol{\sigma}(s_{t+1})\big). \quad (3.13)$$

However, the RG algorithm suffers from the double-sampling problem and does not converge to the same solution than semi-gradient linear TD learning algorithms.

3.4 Deep Q-Network

In recent years, the use of artificial *neural networks* in deep learning has led to breakthroughs due to their ability to learn features from raw data without prior knowledge. Interest in RL exploded in the wake of the results from Mnih et al. (2015), who demonstrated that neural networks could learn to play a collection of Atari games using screen images as input. Since then, RL algorithms using neural networks, i.e., *deep RL* algorithms have shown impressive performance in many domains,

Algorithm 7 DQN (Mnih et al., 2015)

Input: N (capacity of the buffer), T (number of steps), U (number of steps required to update the target network), ϵ
Output: $\hat{Q}$ (estimation of the optimal action-value function Q^*)
Initialize replay buffer $\mathcal{B}$ to capacity N
Initialize weights $\boldsymbol{\Theta}$ of DQN with random weights
Initialize the weights of the target network as $\overline{\boldsymbol{\Theta}} \leftarrow \boldsymbol{\Theta}$
Initialize $s \sim \mu_0$
for each step $t \in [T]$ **do**
  With probability ϵ select a random action a otherwise select $a = \arg\max_{a' \in \mathcal{A}} \hat{Q}(s, a'; \boldsymbol{\Theta})$
  Take action a, observe next state s' and reward $r = R(s, a, s')$
  Store transition (s, a, r, s') in $\mathcal{B}$
  Sample random minibatch of transitions $\{(s_i, a_i, r_i, s'_i)\}_{i=1}^{B}$ from $\mathcal{B}$
  for each transition (s_i, a_i, r_i, s'_i) in $\mathcal{B}$ **do**
    Set $y_i = r_i + \gamma \max_{a' \in \mathcal{A}} \hat{Q}(s'_i, a'; \overline{\boldsymbol{\Theta}})$
    Perform a stochastic gradient descent step on $\frac{1}{2}\big(y_i - \hat{Q}(s_i, a_i; \boldsymbol{\Theta})\big)^2$
    Update the weights of the target network $\overline{\boldsymbol{\Theta}} \leftarrow \boldsymbol{\Theta}$ every U steps
  end for
  $s \leftarrow s'$
end for

including robotics and natural language (Schulman et al., 2017; Haarnoja et al., 2018; Lillicrap et al., 2015).

In this section, we present the use of neural networks in RL through the Deep Q-Network (DQN) algorithm (Mnih et al., 2015), studied in Part III with the use of preprocessings based on Fourier series. The DQN algorithm is the neural network version of the Q-Learning algorithm presented in Section 2.3.3. Like its tabular version, the DQN algorithm aims to approximate the optimal Q-value function $Q^* : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ of a given MDP $(\mathcal{S}, \mathcal{A}, P, R, \mu_0)$ with a neural network approximator $\hat{Q}_{\boldsymbol{\Theta}} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ of parameters $\boldsymbol{\Theta}$. At each timestep t, the agent, in state s_t, selects an ϵ-greedily action a_t with respect to the action values $\hat{Q}_{\boldsymbol{\Theta}}(s_t, \cdot)$. The environment sends it back a new state s_{t+1} and a reward $r_{t+1} = R(s_t, a_t, s_{t+1})$. The agent stores the transition $(s_t, a_t, r_{t+1}, s_{t+1})$ to a replay memory buffer, which stores the last transitions collected by the agent. The parameters $\boldsymbol{\Theta}$ of the neural network are optimized by performing stochastic semi-gradient steps on a batch of transitions $(s_t, a_t, r_{t+1}, s_{t+1})$ drawn from the replay buffer on the loss

$$l(s_t, a_t, r_{t+1}, s_{t+1}) = \frac{1}{2}\Big(r_{t+1} + \gamma \max_{a' \in \mathcal{A}} \hat{Q}_{\overline{\boldsymbol{\Theta}}}(s_{t+1}, a') - \hat{Q}_{\boldsymbol{\Theta}}(s_t, a_t)\Big)^2,$$

where $\overline{\boldsymbol{\Theta}}$ represents the parameters of a *target network*.

Parameters $\overline{\boldsymbol{\Theta}}$ of the target network are updated to $\boldsymbol{\Theta}$ every U iterations. The target network is a duplicate of the neural network but is updated less frequently to provide a stable set of fixed target values for training. In particular, its use stabilizes the learning process by preventing the learning process from becoming unstable due to constantly shifting targets.

The use of a replay buffer (Lin, 1992) is motivated by the fact that the learning is online with the collection of transitions that are not i.i.d. Updating parameters $\boldsymbol{\Theta}$ with a batch of transitions randomly drawn from the replay buffer helps in breaking correlations between transitions and leads to more stable training. Furthermore, the use of a replay buffer can prevent feedback loops and oscillations as the current parameters $\boldsymbol{\Theta}$ determine the behavior of the agent, which in turn

determines the next transitions that are visited and used for optimization. For example, in online learning, if an agent learns to prefer using certain actions, it may collect more data derived from those actions, become biased, reinforce its preference for those actions, and could get stuck in a poor local minimum or even diverge catastrophically. By using experience replay, the behavior distribution is averaged over many previous transitions and prevents such behavior. The use of a replay buffer also makes DQN more data efficient as the same transitions can be used multiple times. The optimization in DQN is performed using RMSprop. A pseudocode version of DQN is presented in Algorithm 7. As a TD learning algorithm using function approximation, DQN aims to minimize the Mean-Squared Projected Bellman error for the optimal action-value function $Q^* : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. Variants and improvements of DQN can be found in Hessel et al. (2018).

Chapter 4

Least-Squares Temporal Difference Learning

Rather than considering a gradient-based approach, the *Least-Squares Temporal Difference Learning* (LSTD) (Bradtke and Barto, 1996) algorithm is a linear TD learning method that analytically solves an empirical version of the projection error (equation 3.6) and of the fixed-point error (equation 3.7).

Section 4.1 introduces the LSTD algorithm and its analytical solution. Section 4.2 presents the historical motivation behind LSTD by viewing it as a derivation of the linear least-squares approximation on $\bar{R}^\pi : \mathcal{S} \to \mathbb{R}$, $s \mapsto \mathbb{E}_{s' \sim P^\pi(\cdot|s)}[R^\pi(s, s')]$. In Section 4.3, we study the convergence of LSTD and its connection with TD learning methods using a gradient-based approach. Section 4.4 introduces an iterative version of LSTD and a model-based interpretation. In Section 4.5, we review some popular regularized variants of LSTD, including the regularized LSTD studied in Part II.

4.1 Definition

As described in Chapter 3, we formalize the behavior of the agent in its environment using a Markov Reward Process (MRP) $(\mathcal{S}, P^\pi, R^\pi, \mu_0)$. The dynamics $P^\pi : \mathcal{S} \to \mathcal{P}(\mathcal{S})$ are typically unknown in RL problems. Instead, we assume we have access to a dataset of n transitions consisting of states, rewards, and next-states drawn from a MRP, i.e., we have $\mathcal{D}_{\text{train}} := \left\{(\boldsymbol{s}_i, r_i, \boldsymbol{s}'_i)\right\}_{i=1}^n$ where $\boldsymbol{s}'_i \sim P^\pi(\boldsymbol{s}_i)$ and $r_i = R^\pi(\boldsymbol{s}_i, \boldsymbol{s}'_i)$. From the dataset $\mathcal{D}_{\text{train}} := \left\{(\boldsymbol{s}_i, r_i, \boldsymbol{s}'_i)\right\}_{i=1}^n$, we define the sample matrices

$$\boldsymbol{X}_n = [s_1, \dots, s_n] \in \mathbb{R}^{d \times n}, \quad \boldsymbol{r} = [r_1, \dots, r_n]^T \in \mathbb{R}^n, \quad \boldsymbol{X}'_n = [s'_1, \dots, s'_n] \in \mathbb{R}^{d \times n}. \tag{4.1}$$

Considering the feature map $\boldsymbol{\sigma} : \mathcal{S} \to \mathbb{R}^N$ of a linear parameterized model of parameters $\boldsymbol{\theta}$ (as defined in equation 3.8), we define the following features matrices

$$\boldsymbol{\Sigma}_{\boldsymbol{X}_n} = [\boldsymbol{\sigma}(s_1), \dots, \boldsymbol{\sigma}(s_n)] \in \mathbb{R}^{N \times n}, \quad \boldsymbol{\Sigma}_{\boldsymbol{X}'_n} = [\boldsymbol{\sigma}(s'_1), \dots, \boldsymbol{\sigma}(s'_n)] \in \mathbb{R}^{N \times n}. \tag{4.2}$$

LSTD solves an empirical version of equation 3.6 and equation 3.7 (Hoffman et al., 2011) defined as follows:

$$\hat{\boldsymbol{u}}_n = \arg\min_{\boldsymbol{u}_n \in \mathbb{R}^N} \|\boldsymbol{r} + \gamma \boldsymbol{\Sigma}_{\boldsymbol{X}'_n}^T \hat{\boldsymbol{\theta}}_n - \boldsymbol{\Sigma}_{\boldsymbol{X}_n}^T \boldsymbol{u}_n\|^2 \quad \text{(projection error)} \tag{4.3}$$

$$\hat{\boldsymbol{\theta}}_n = \arg\min_{\boldsymbol{\theta}_n \in \mathbb{R}^N} \|\boldsymbol{\Sigma}_{\boldsymbol{X}_n}^T \hat{\boldsymbol{u}}_n - \boldsymbol{\Sigma}_{\boldsymbol{X}_n}^T \boldsymbol{\theta}_n\|^2 \quad \text{(fixed-point error)}. \tag{4.4}$$

LSTD solves the fixed-point of the linear system approximation given by equation 4.3 and equation 4.4 and gives

$$\hat{\boldsymbol{\theta}}_n = \left[\boldsymbol{\Sigma}_{\boldsymbol{X}_n}\left[\boldsymbol{\Sigma}_{\boldsymbol{X}_n} - \gamma \boldsymbol{\Sigma}_{\boldsymbol{X}'_n}\right]^T\right]^{-1} \boldsymbol{\Sigma}_{\boldsymbol{X}_n} \boldsymbol{r}, \tag{4.5}$$

with the assumption that the matrix $\boldsymbol{A}_n = \boldsymbol{\Sigma}_{\boldsymbol{X}_n}\left[\boldsymbol{\Sigma}_{\boldsymbol{X}_n} - \gamma \boldsymbol{\Sigma}_{\boldsymbol{X}'_n}\right]^T$ is non-singular.

4.2 LSTD as a Linear Least-Squares Approximation on $\bar{R}^\pi$

Before showing that LSTD can be viewed as a linear least-squares approximation on $\bar{R}^\pi : \mathcal{S} \to \mathbb{R}, s \mapsto \mathbb{E}_{s' \sim P^\pi(\cdot|s)}\left[R^\pi(s, s')\right]$ with an input noise in Section 4.2.2, we will provide first the basics of least-squares function approximations and instrumental variables in Section 4.2.1.

4.2.1 Linear Least-Square Function Approximation & Instrumental Variables.

Least-Square Approximation. The objective in linear least-squares function approximation is to linearly approximate a target function $f : \mathbb{R}^d \to \mathbb{R}$ using n samples of observed inputs $\{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^n$ and their corresponding observed predictions $\{y_i \in \mathbb{R}\}_{i=1}^n$. Assuming that the targets $\{y_i \in \mathbb{R}\}_{i=1}^n$ are generated by a linear function f and corrupted with noise, we have the following for all $i \in [N]$

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i = \boldsymbol{x}_i^T \boldsymbol{\theta}^* + \epsilon_i, \tag{4.6}$$

where $\boldsymbol{\theta}^* \in \mathbb{R}^d$ is the unknown vector of parameters defining $f : \mathbb{R}^d \to \mathbb{R}$ and ϵ_i is the output observation noise associated to the sample i. Least-squares approximations analytically minimize the quadratic objective function $J(\boldsymbol{\theta})$ defined as

$$J(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^n (y_i - \boldsymbol{x}_i^T \boldsymbol{\theta})^2.$$

Taking the partial derivative of $J(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$, setting this equal to zero and solving for $\boldsymbol{\theta}$ gives

$$\hat{\boldsymbol{\theta}}_n = \frac{1}{n}\left[\frac{1}{n} \sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i^T\right]^{-1} \sum_{i=1}^n \boldsymbol{x}_i y_i. \tag{4.7}$$

If the correlation matrix $\frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i^T$ is nonsingular and finite and the output observation noise ϵ_i is uncorrelated with the input observations $\boldsymbol{x}_i$, then $\hat{\boldsymbol{\theta}}_n$ converges with probability 1 to $\boldsymbol{\theta}^*$ as $n \to \infty$ (Young, 2012).

Least-Square Approximation with Input Noise. Instead of being able to directly observe $\boldsymbol{x}_i$ like in equation 4.6, we assume we only observe noisy samples $\hat{\boldsymbol{x}}_i = \boldsymbol{x}_i + \boldsymbol{\eta}_i$, where $\boldsymbol{\eta}_i$ is the

input observation noise vector for the sample i. In such a setting, we have

$$y_i = f(\boldsymbol{x}_i) + \epsilon = f(\hat{\boldsymbol{x}}_i - \boldsymbol{\eta}_i) + \epsilon = \boldsymbol{x}_i^T \boldsymbol{\theta}^* - \boldsymbol{\eta}_i^T \boldsymbol{\theta}^* + \epsilon_i,$$

for all $i \in [n]$. Substituting $\hat{\boldsymbol{x}}_i$ directly with $\boldsymbol{x}_i$ in equation 4.7 introduces noise and a bias, with the consequence that $\hat{\boldsymbol{\theta}}_n$ no longer converges to $\boldsymbol{\theta}^*$. One way to overcome this problem is by introducing instrumental variables (Young, 2012). An instrumental variable $\boldsymbol{z}_i$ is a vector correlated with the true input vectors $\boldsymbol{x}_i$ but uncorrelated with the observation noise $\boldsymbol{\eta}_i$. A modification of equation 4.7 that uses the instrumental variables and the noisy inputs is given by

$$\hat{\boldsymbol{\theta}}_n = \frac{1}{n}\left[\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{z}_i \hat{\boldsymbol{x}}_i^T\right]^{-1} \sum_{i=1}^{n} \boldsymbol{z}_i y_i. \tag{4.8}$$

If the correlation matrix $\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{z}_i \hat{\boldsymbol{x}}_i^T$ is nonsingular and finite and if the true input vector $\boldsymbol{x}_i$ and the output observation noise ϵ_i are uncorrelated with the instrumental variable $\boldsymbol{z}_i$, then the solution $\hat{\boldsymbol{\theta}}_n$ defined in equation 4.8 converges with probability 1 to $\boldsymbol{\theta}^*$ as $n \to \infty$ (Young, 2012).

4.2.2 LSTD as a Least-Squares Approximation on $\bar{R}^\pi$

In this section, we assume the existence of a parameter vector $\boldsymbol{\theta}^* \in \mathbb{R}^N$ such that, for all states $s \in \mathcal{S}$, we have

$$V^\pi(s) = \boldsymbol{\sigma}(\boldsymbol{s})^T \boldsymbol{\theta}^*,$$

for the feature map $\boldsymbol{\sigma} : \mathcal{S} \to \mathbb{R}^N$. We recall that V^π is the unique-fixed point of the Bellman operator $\mathcal{T}_V^\pi$ (equation 4.9). For all states $s \in \mathcal{S}$, we have

$$V^\pi(s) = \big(\mathcal{T}_V^\pi V^\pi\big)(s) = \bar{R}^\pi(s) + \gamma \mathbb{E}_{s' \sim P^\pi(\cdot|s)}\big[V^\pi(s')\big]. \tag{4.9}$$

From above, $\bar{R}^\pi : \mathcal{S} \to \mathbb{R}$ can be linearly approximated with $\boldsymbol{\theta}^*$ since

$$\bar{R}^\pi(s) = V^\pi(s) - \gamma \mathbb{E}_{s' \sim P^\pi(\cdot|s)}\big[V^\pi(s')\big] = \mathbb{E}_{s' \sim P^\pi(\cdot|s)}\big[\ [\boldsymbol{\sigma}(s) - \gamma \boldsymbol{\sigma}(s')]^T\ \big]\boldsymbol{\theta}^*.$$

The objective is to find the parameter vector $\boldsymbol{\theta}^* \in \mathbb{R}^N$ that linearly approximates both $\bar{R}^\pi : \mathcal{S} \to \mathbb{R}$ and $V^\pi : \mathcal{S} \to \mathbb{R}$, using least-squares methods presented in the previous section with the transitions collected in $\mathcal{D}_{\text{train}} := \big\{(\boldsymbol{s}_i, r_i, \boldsymbol{s}_i')\big\}_{i=1}^n$. For every transition $i \in [n]$ in $\mathcal{D}_{\text{train}}$, we have

$$\begin{aligned} r_i &= \mathbb{E}_{s' \sim P^\pi(\cdot|s_i)}\big[\ [\boldsymbol{\sigma}(s_i) - \gamma \boldsymbol{\sigma}(s')]^T\ \big]\boldsymbol{\theta}^* + \big(r_i - \bar{R}^\pi(s_i)\big) \\ &= \boldsymbol{x}_i^T \boldsymbol{\theta}^* + \epsilon_i, \end{aligned}$$

where $\epsilon_i = r_i - \bar{R}^\pi(s_i)$ is the observed output noise and $\boldsymbol{x}_i = \mathbb{E}_{s' \sim P^\pi(\cdot|s_i)}\big[\ [\boldsymbol{\sigma}(s_i) - \gamma \boldsymbol{\sigma}(s')]^T\ \big]$ is an input vector observed for the transition i. The noise term ϵ_i has a zero mean and is uncorrelated with the input vector $\boldsymbol{x}_i$ (Bradtke and Barto, 1996). If the transition function $P^\pi : \mathcal{S} \to \mathcal{P}(\mathcal{S})$ and the state space $\mathcal{S}$ are known, we obtain a similar expression than in equation 4.6. Therefore, if $\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^T$ is nonsingular and finite, then the least-square solution $\hat{\boldsymbol{\theta}}_n$ given by equation 4.7 converges with probability 1 to $\boldsymbol{\theta}^*$ as $n \to \infty$. However, $P^\pi : \mathcal{S} \to \mathcal{P}(\mathcal{S})$ and the state space $\mathcal{S}$ are typically unknown in RL. By exploiting the next states $\{\boldsymbol{s}_i'\}_{i=1}^n$ stored in $\mathcal{D}_{\text{train}}$, we can observe that for each transition $i \in [n]$ in $\mathcal{D}_{\text{train}}$, we have

$$\hat{\boldsymbol{x}}_i = \boldsymbol{\sigma}(s_i) - \gamma \boldsymbol{\sigma}(s_i')$$

$$
\begin{aligned}
&= \mathbb{E}_{s' \sim P^\pi(\cdot|s_i)}\big[\ [\boldsymbol{\sigma}(s_i) - \gamma\boldsymbol{\sigma}(s')]^T\ \big] + \gamma\mathbb{E}_{s' \sim P^\pi(\cdot|s_i)}\big[\ [\boldsymbol{\sigma}(s_i') - \boldsymbol{\sigma}(s')]^T\ \big] \\
&= \boldsymbol{x}_i + \boldsymbol{\eta}_i,
\end{aligned}
$$

for which $\boldsymbol{\eta}_i = \gamma\mathbb{E}_{s' \sim P^\pi(\cdot|s_i)}\big[\ [\boldsymbol{\sigma}(s_i') - \boldsymbol{\sigma}(s')]^T\ \big]$ is an input observed noise. We find a similar expression than in equation 4.8 since

$$
\begin{aligned}
r_i &= \mathbb{E}_{s' \sim P^\pi(\cdot|s_i)}\big[\ [\boldsymbol{\sigma}(s_i) - \gamma\boldsymbol{\sigma}(s')]^T\ \big]\boldsymbol{\theta}^* + \big(r_i - \bar{R}^\pi(s_i)\big) \\
&= \boldsymbol{x}_i^T\boldsymbol{\theta}^* + \epsilon_i \\
&= \hat{\boldsymbol{x}}_i^T\boldsymbol{\theta}^* - \boldsymbol{\eta}_i^T\boldsymbol{\theta}^* + \epsilon_i.
\end{aligned}
$$

By observing that for each transition i, the variable $\boldsymbol{z}_i = \boldsymbol{\sigma}(s_i)$ is uncorrelated with the input noise $\boldsymbol{\eta}_i$ and the output noise ϵ_i, we can use $\boldsymbol{z}_i$ as an instrumental variable and consider the least-squares approximation approach with input noise to solve the problem. In particular, from equation 4.8, if $\frac{1}{n}\sum_{i=1}^n \boldsymbol{z}_i\hat{\boldsymbol{x}}_i^T$ is nonsingular and finite, we find the solution

$$
\begin{aligned}
\hat{\boldsymbol{\theta}}_n &= \frac{1}{n}\left[\frac{1}{n}\sum_{i=1}^n \boldsymbol{z}_i\hat{\boldsymbol{x}}_i^T\right]^{-1}\sum_{i=1}^n \boldsymbol{z}_i r_i \\
&= \frac{1}{n}\left[\frac{1}{n}\sum_{i=1}^n \boldsymbol{\sigma}(s_i)\big(\boldsymbol{\sigma}(s_i) - \gamma\boldsymbol{\sigma}(s_i')\big)^T\right]^{-1}\sum_{i=1}^n \boldsymbol{\sigma}(s_i) r_i \\
&= \frac{1}{n}\left[\frac{1}{n}\boldsymbol{\Sigma}_{\boldsymbol{X}_n}\big[\boldsymbol{\Sigma}_{\boldsymbol{X}_n} - \gamma\boldsymbol{\Sigma}_{\boldsymbol{X}_n'}\big]^T\right]^{-1}\boldsymbol{\Sigma}_{\boldsymbol{X}_n}\boldsymbol{r}.
\end{aligned}
$$

This solution corresponds to the solution returned by LSTD in equation 4.5. Analytically solving equation 4.3 and equation 4.4 is thus equivalent to a least-squares problem on $\bar{R}^\pi : \mathcal{S} \to \mathbb{R}$ with noisy observed inputs $\{\hat{\boldsymbol{x}}_i\}_{i=1}^n$ and outputs $\{\hat{r}_i\}_{i=1}^n$.

4.3 Convergence of LSTD

In this section, without loss of generality, we consider finite MRPs in which the state space $\mathcal{S}$ is finite, i.e., in which $|\mathcal{S}| < \infty$. The state space $\mathcal{S}$ can be thus described by the state matrix $\boldsymbol{S} \in \mathbb{R}^{d \times |\mathcal{S}|}$, where each column of $\boldsymbol{S}$ denoted by $\boldsymbol{S}_i$ represents a state in $\mathcal{S}$. Furthermore, the transition probability matrix associated with the stochastic kernel P^π is denoted by $\boldsymbol{P}^\pi \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$.

Lemma 4.3.1 (Convergence LSTD (Bradtke and Barto, 1996; Nedić and Bertsekas, 2003))**.** *If* (1) *each state* $s \in \mathcal{S}$ *is visited infinitely often;* (2) *if each state* $s \in \mathcal{S}$ *is visited in the long run with probability* 1 *in proportion* $\mu(s)$*; and* (3) *if* $\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{D}_\mu\big[\boldsymbol{I}_{|\mathcal{S}|} - \gamma\boldsymbol{P}\big]\boldsymbol{\Sigma}_{\mathcal{S}}^T$ *is invertible, for* $\boldsymbol{\Sigma}_{\mathcal{S}} = \big[\boldsymbol{\sigma}(\boldsymbol{S}_1), \cdots, \boldsymbol{\sigma}(\boldsymbol{S}_{|\mathcal{S}|})\big]$ *the feature matrix of the state space* $\mathcal{S}$ *and* $\boldsymbol{D}_\mu = \mathrm{diag}(\mu)$ *the diagonal matrix of* $\mu \in \mathbb{R}^{|\mathcal{S}|}$*, then the weight vector* $\hat{\boldsymbol{\theta}}_n$ *returned by LSTD (equation 4.5) converges to*

$$
\hat{\boldsymbol{\theta}} = \Big[\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{D}_\mu\big[\boldsymbol{I}_{|\mathcal{S}|} - \gamma\boldsymbol{P}\big]\boldsymbol{\Sigma}_{\mathcal{S}}^T\Big]^{-1}\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{D}_\mu\bar{\boldsymbol{r}}^\pi \tag{4.10}
$$

with probability 1*, where* $\bar{\boldsymbol{r}}^{\pi T} = \big[\bar{R}^\pi(\boldsymbol{S}_1), \cdots, \bar{R}^\pi(\boldsymbol{S}_{|\mathcal{S}|})\big]$.

If the considered data distribution μ is the stationary distribution μ^π of $\boldsymbol{P}^\pi$, we are in the *on-policy* setting and we find

$$
\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^*,
$$

where $\boldsymbol{\theta}^*$ is the analytical solution of the MSPBE (equation 3.4) defined as

$$\boldsymbol{\theta}^* = \Big[\boldsymbol{\Sigma}_{\mathcal{S}} \boldsymbol{D}_{\mu^\pi} \big[\boldsymbol{I}_{|\mathcal{S}|} - \gamma \boldsymbol{P}\big] \boldsymbol{\Sigma}_{\mathcal{S}}^T\Big]^{-1} \boldsymbol{\Sigma}_{\mathcal{S}} \boldsymbol{D}_{\mu^\pi} \bar{\boldsymbol{r}}^\pi.$$

When transitions are drawn from the stationary distribution μ^π or are derived from sample paths in ergodic or absorbing MRPs, the parameter vector $\hat{\boldsymbol{\theta}}_n$ defined in equation 4.5 and returned by LSTD converges to the analytical solution $\boldsymbol{\theta}^*$ of the MSPBE (Dann et al., 2014; Ciosek, 2013; Sutton, 1988).

In the off-policy scenario, i.e, when $\mu \neq \mu^\pi$ the situation is more complex, and the convergence towards $\boldsymbol{\theta}^*$ no longer holds (Bertsekas and Yu, 2009; Geist et al., 2014; Dann et al., 2014). However, by using eligibity traces and importance sampling reweighting (Glynn and Iglehart, 1989), the convergence of $\boldsymbol{\theta}_n$ towards $\boldsymbol{\theta}^*$ can be facilitated (Bertsekas and Yu, 2009; Geist et al., 2014; Dann et al., 2014). Finally, as linear semi-gradient TD learning algorithms tend to converge to the minimum of the MSPBE under the Robbins-Monro conditions, they converge to the same solution as the one returned by LSTD (Tsitsiklis and Van Roy, 1996; Dann et al., 2014; Sutton and Barto, 2018).

4.4 Recursive LSTD

The LSTD solution of equation 4.5 explicitely estimates the matrices

$$\hat{\boldsymbol{A}}_n = \sum_{i=1}^{n} \boldsymbol{\sigma}(s_i)\big(\boldsymbol{\sigma}(s_i) - \gamma \boldsymbol{\sigma}(s_i')\big)^T \quad \text{and} \quad \hat{\boldsymbol{b}}_n = \sum_{i=1}^{n} \boldsymbol{\sigma}(s_i) r_i$$

and then computes $\hat{\boldsymbol{\theta}}_n = \hat{\boldsymbol{A}}_n^{-1} \hat{\boldsymbol{b}}_n$. Estimating $\hat{\boldsymbol{\theta}}_n$ requires thus to invert the matrix $\hat{\boldsymbol{A}}_n$ of dimension $N \times N$. LSTD is more expensive in computation and memory with a complexity $O(n^3)$ than semi-gradient TD learning algorithms with a complexity $O(n)$. A solution to reduce the complexity to $O(n)$ is to compute and update iteratively $\hat{\boldsymbol{A}}_n$ and $\hat{\boldsymbol{b}}_n$ (Bradtke and Barto, 1996). Estimates $\hat{\boldsymbol{A}}_{n+1}, \hat{\boldsymbol{b}}_{n+1}$ can be computed iteratively using the $(n+1)^{\text{th}}$ transition $(s_{n+1}, r_{n+1}, s'_{n+1})$ as

$$\begin{aligned} \hat{\boldsymbol{A}}_{n+1} &= \hat{\boldsymbol{A}}_n + \boldsymbol{\sigma}(s_{n+1})\big(\boldsymbol{\sigma}(s_{n+1}) - \gamma \boldsymbol{\sigma}(s'_{n+1})\big)^T \\ \hat{\boldsymbol{b}}_{n+1} &= \hat{\boldsymbol{b}}_n + \boldsymbol{\sigma}(s_{n+1}) r_{n+1}. \end{aligned} \tag{4.11}$$

The iterative computation of $\hat{\boldsymbol{A}}_n^{-1}$ is obtained with the Sherman-Morrison formula as

$$\begin{aligned} \hat{\boldsymbol{A}}_{n+1}^{-1} &= \big[\hat{\boldsymbol{A}}_n + \boldsymbol{\sigma}(s_{n+1})\big(\boldsymbol{\sigma}(s_{n+1}) - \gamma \boldsymbol{\sigma}(s'_{n+1})\big)^T\big]^{-1} \\ &= \hat{\boldsymbol{A}}_n^{-1} - \frac{\hat{\boldsymbol{A}}_n^{-1} \boldsymbol{\sigma}(s_{n+1})\big(\boldsymbol{\sigma}(s_{n+1}) - \gamma \boldsymbol{\sigma}(s'_{n+1})\big)^T \hat{\boldsymbol{A}}_n^{-1}}{1 + \big(\boldsymbol{\sigma}(s_{n+1}) - \gamma \boldsymbol{\sigma}(s'_{n+1})\big)^T \hat{\boldsymbol{A}}_n^{-1} \boldsymbol{\sigma}(s_{n+1})}. \end{aligned}$$

The sequence $(\boldsymbol{A}_n^{-1})_n$ needs to be initialized with a hand-designed element $\hat{\boldsymbol{A}}_0^{-1}$ that incorporates some prior knowledge. Ideally, $\hat{\boldsymbol{A}}_0^{-1}$ should be the null-matrix. In practice, a popular choice is $\hat{\boldsymbol{A}}_0^{-1} = \frac{1}{\lambda} \boldsymbol{I}_N$ for $\lambda > 0$. Recursive LSTD does not require a learning rate for gradient descent but requires a parameter λ for initialization. If λ is set too high, the sequence of inverses can vary wildly. On the other hand, if λ is set too low, then learning slows down. As discussed in the following section, the parameter λ acts as a regularizer. From Nedić and Bertsekas (2003), as

Algorithm 8 The On-line Recursive LSTD Algorithm

Input: π (the policy to be evaluated), T (number of steps), $\sigma : \mathcal{S} \to \mathbb{R}^N$ (feature representation), ϵ
Output: Parameter vector $\hat{\boldsymbol{\theta}}_n$
$\hat{\boldsymbol{A}}^{-1} \leftarrow \frac{1}{\lambda}\boldsymbol{I}_N$
$\hat{\boldsymbol{b}} \leftarrow 0$
Initialize $s \sim \mu_0$
for each step $t \in [T]$ **do**
Choose action $a \sim \pi(\cdot|s)$
Take action a, observe next state s' and reward $r = R(s, a, s')$
$\boldsymbol{v} \leftarrow \hat{\boldsymbol{A}}^{-1T}\big(\boldsymbol{\sigma}(s) - \gamma\boldsymbol{\sigma}(s')\big)$
$\hat{\boldsymbol{A}}^{-1} \leftarrow \frac{\hat{\boldsymbol{A}}^{-1} - \hat{\boldsymbol{A}}^{-1}\boldsymbol{\sigma}(s)\boldsymbol{v}^T}{1+\boldsymbol{v}^T\boldsymbol{\sigma}(s)}$
$\hat{\boldsymbol{b}} \leftarrow \hat{\boldsymbol{b}} + r\boldsymbol{\sigma}(s)$
$\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{A}}^{-1}\hat{\boldsymbol{b}}$
$s \leftarrow s'$
end for

$n \to \infty$ we have

$$\hat{\boldsymbol{A}}_n \to \boldsymbol{A} \quad \text{and} \quad \hat{\boldsymbol{b}}_n \to \boldsymbol{b},$$

where $\boldsymbol{A} = \boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{D}_\mu\big[\boldsymbol{I}_{|\mathcal{S}|} - \gamma\boldsymbol{P}\big]\boldsymbol{\Sigma}_{\mathcal{S}}^T$ and $\boldsymbol{b} = \boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{D}_\mu\bar{\boldsymbol{r}}^\pi$ define the solution $\boldsymbol{\theta}^*$ in equation 4.10. The pseudocode of online recursive LSTD is presented in Algorithm 8.

From the iterative equation 4.11, we can also provide a model-based interpretation of LSTD (Boyan, 1999), since the matrix $\hat{\boldsymbol{A}}_n$ contains an empirical model of the transition probabilities. Indeed, we can rewrite $\hat{\boldsymbol{A}}_n$ and $\hat{\boldsymbol{b}}_n$ as

$$\hat{\boldsymbol{A}}_n = \boldsymbol{\Sigma}_{\mathcal{S}}\big[\hat{\boldsymbol{C}}_n - \gamma\hat{\boldsymbol{N}}_n\big]\boldsymbol{\Sigma}_{\mathcal{S}}^T \quad \text{and} \quad \hat{\boldsymbol{b}}_n = \boldsymbol{\Sigma}_{\mathcal{S}}\hat{\boldsymbol{C}}_n\hat{\bar{\boldsymbol{r}}};$$

where $\hat{\boldsymbol{C}}_n$ is a diagonal matrix containing the state visit counts of each state state $s \in \mathcal{S}$ in the dataset $\mathcal{D}_{\text{train}} := \big\{(\boldsymbol{s}_i, r_i, \boldsymbol{s}'_i)\big\}_{i=1}^n$; elements $\big[\hat{\boldsymbol{N}}_n\big]_{ij}$ of matrix $\hat{\boldsymbol{N}}_n$ contain the number of times a transition from state $\boldsymbol{S}_i$ to state $\boldsymbol{S}_j$ has been observed in $\mathcal{D}_{\text{train}}$; and $\hat{\bar{\boldsymbol{r}}}_i$ denotes the average observed reward when being in state $\boldsymbol{S}_i$ for all in $i \in [|\mathcal{S}|]$. Using those notations, we can rewrite equation 4.5 for $\hat{\boldsymbol{\theta}}_n$ as

$$\hat{\boldsymbol{\theta}}_n = \big[\boldsymbol{\Sigma}_{\mathcal{S}}\hat{\boldsymbol{C}}_n[\boldsymbol{I}_{|\mathcal{S}|} - \gamma\hat{\boldsymbol{P}}_n]\boldsymbol{\Sigma}_{\mathcal{S}}^T\big]^{-1}\boldsymbol{\Sigma}_{\mathcal{S}}\hat{\boldsymbol{C}}_n\hat{\bar{\boldsymbol{r}}}.$$

Note that if the diagonal matrix $\hat{\boldsymbol{C}}_n$ is invertible, then we can define the matrix $\hat{\boldsymbol{P}}_n = \hat{\boldsymbol{C}}_n^{-1}\hat{\boldsymbol{N}}_n$ and show that $\hat{\boldsymbol{P}}_n$ is a transition probability matrix. As $n \to \infty$, we have $\hat{\boldsymbol{P}}_n \to \boldsymbol{P}$ and $\frac{1}{n}\hat{\boldsymbol{C}}_n \to \boldsymbol{D}_\mu$. $\hat{\boldsymbol{P}}_n$ and $\hat{\boldsymbol{C}}_n$ can be thus interpreted as approximations of $\boldsymbol{P}$ and $\boldsymbol{D}_\mu$.

4.5 Regularized LSTD

Value function approximation faces several challenges in high-dimensional feature space when considering a large number of features N. Indeed, when the number of transitions n collected in $\mathcal{D}_{\text{train}}$ is small compared to the number of features N, performance can deteriorate as explained by the bias-variance tradeoff model described in Section 5.1. This scenario often results in *overfitting*, where the model better fits the noise of outputs rather than the underlying system itself. Over-

	Regularization Penalities	Optimization Technique
LSTD with l_2	$\Omega_p(\boldsymbol{u}_n) \propto \|\boldsymbol{u}_n\|, \Omega_f(\boldsymbol{\theta}_n) = 0$	closed-form solution (Bradtke and Barto, 1996)
LSTD with l_2, l_2	$\Omega_p(\boldsymbol{u}_n) \propto \|\boldsymbol{u}_n\|, \Omega_f(\boldsymbol{\theta}_n) \propto \|\boldsymbol{\theta}_n\|$	closed-form solution (Hoffman et al., 2011)
LARS-TD	$\Omega_p(\boldsymbol{u}_n) \propto \|\boldsymbol{u}_n\|_1, \Omega_f(\boldsymbol{\theta}_n) = 0$	custom LARS–like solver (Kolter and Ng, 2009)
LC-TD	$\Omega_p(\boldsymbol{u}_n) \propto \|\boldsymbol{u}_n\|_1, \Omega_f(\boldsymbol{\theta}_n) = 0$	standard LCP solvers (Johns et al., 2010)
l_1-PBR	$\Omega_p(\boldsymbol{u}_n) = 0, \Omega_f(\boldsymbol{\theta}_n) \propto \|\boldsymbol{\theta}_n\|_1$	standard Lasso solvers (Geist and Scherrer, 2011)
LSTD with l_2, l_1	$\Omega_p(\boldsymbol{u}_n) \propto \|\boldsymbol{u}_n\|, \Omega_f(\boldsymbol{\theta}_n) \propto \|\boldsymbol{\theta}_n\|_1$	standard Lasso solvers (Hoffman et al., 2011)
Laplacian-based reg. LSTD	$\Omega_p(\boldsymbol{u}_n) \propto \|\boldsymbol{L}\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{\theta}\|, \Omega_f(\boldsymbol{\theta}_n) = 0$	closed-form solution (Geist et al., 2012)

Table 4.1: Comparison of Regularization Approaches for LSTD. $\Omega_p : \mathbb{R}^N \to \mathbb{R}$ and $\Omega_f : \mathbb{R}^N \to \mathbb{R}$ are the regularization terms in the nested problem formulation of LSTD (Equations 4.12 and 4.13) (Dann et al., 2014). (*) l_1-PBR actually assumes a small l_2 regularization on the operator problem if the estimate of $\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{D}_{\mu}\boldsymbol{\Sigma}_{\mathcal{S}}^T$ is singular.

fitting typically occurs when the number of parameters exceeds the number of samples. In this section, we will examine various regularization methods designed to prevent such overfitting.

Most regularization methods in value function approximation introduce different penalty terms into the projection and/or to the fixed-point error equations defined in equation 3.6 and equation 3.7. In particular, regularization penalities $\Omega_p : \mathbb{R}^N \to \mathbb{R}$ and $\Omega_f : \mathbb{R}^N \to \mathbb{R}$ are added as follows:

$$\hat{\boldsymbol{u}}_n = \underset{\boldsymbol{u}_n \in \mathbb{R}^N}{\arg\min} \Big(\|\boldsymbol{r} + \gamma \boldsymbol{\Sigma}_{\boldsymbol{X}'_n}^T \hat{\boldsymbol{\theta}}_n - \boldsymbol{\Sigma}_{\boldsymbol{X}_n}^T \boldsymbol{u}_n\|^2 + \Omega_p(\boldsymbol{u}_n) \Big) \quad \text{(projection error)} \tag{4.12}$$

$$\hat{\boldsymbol{\theta}}_n = \underset{\boldsymbol{\theta}_n \in \mathbb{R}^N}{\arg\min} \Big(\|\boldsymbol{\Sigma}_{\boldsymbol{X}_n}^T \hat{\boldsymbol{u}}_n - \boldsymbol{\Sigma}_{\boldsymbol{X}_n}^T \boldsymbol{\theta}_n\|^2 + \Omega_f(\boldsymbol{\theta}_n) \Big) \quad \text{(fixed-point error).} \tag{4.13}$$

Different regularization approaches are presented in Table 4.1. The simplest and most popular form of regularization involves adding an l_2 regularization penalty $\Omega_p(\boldsymbol{u}_n) = \lambda \|\boldsymbol{u}_n\|$ to the projection error (equation 4.12) (Kolter and Ng, 2009; Hoffman et al., 2011; Chen et al., 2013). The l_2-regularization parameter λ controls the strength of regularization. With the introduction of the l_2-penality $\Omega_p(\boldsymbol{u}_n) = \lambda \|\boldsymbol{u}_n\|$ into the projection error, equation 4.12 and 4.13 still have a closed-form solution given by

$$\begin{aligned} \hat{\boldsymbol{\theta}}_n^{\lambda} &= \Big[\boldsymbol{\Sigma}_{\boldsymbol{X}_n} \big[\boldsymbol{\Sigma}_{\boldsymbol{X}_n} - \gamma \boldsymbol{\Sigma}_{\boldsymbol{X}'_n} \big]^T + \lambda \boldsymbol{I}_N \Big]^{-1} \boldsymbol{\Sigma}_{\boldsymbol{X}_n} \boldsymbol{r} \\ &= \big[\hat{\boldsymbol{A}}_n + \lambda \boldsymbol{I}_N \big]^{-1} \hat{\boldsymbol{b}}_n, \end{aligned}$$

where $\hat{\boldsymbol{A}}_n = \boldsymbol{\Sigma}_{\boldsymbol{X}_n} \big[\boldsymbol{\Sigma}_{\boldsymbol{X}_n} - \gamma \boldsymbol{\Sigma}_{\boldsymbol{X}'_n} \big]^T$ and $\hat{\boldsymbol{b}}_n = \boldsymbol{\Sigma}_{\boldsymbol{X}_n} \boldsymbol{r}$. In previous sections, we have assumed that

$\hat{\boldsymbol{A}}_n = \boldsymbol{\Sigma}_{\boldsymbol{X}_n}\left[\boldsymbol{\Sigma}_{\boldsymbol{X}_n} - \gamma\boldsymbol{\Sigma}_{\boldsymbol{X}'_n}\right]^T$ is invertible to compute $\hat{\boldsymbol{\theta}}_n = \hat{\boldsymbol{A}}_n^{-1}\hat{\boldsymbol{b}}_n$. However, in practice, $\hat{\boldsymbol{A}}_n$ may be singular. To avoid this and instead of computing $\hat{\boldsymbol{\theta}}_n$, many practical implementations of LSTD opt for its l_2 variant and compute $\hat{\boldsymbol{\theta}}_n^{\lambda} = \left[\hat{\boldsymbol{A}}_n + \lambda\boldsymbol{I}_N\right]^{-1}\hat{\boldsymbol{b}}_n$, by choosing λ such that it is not equal to one of the eigenvalues of $\hat{\boldsymbol{A}}_n$, i.e., $\lambda \notin \nu(\hat{\boldsymbol{A}}_n)$. For example, by initializing $\boldsymbol{A}_0 = \lambda\boldsymbol{I}_N$, the recursive LSTD (Algorithm 8) introduces an l_2-regularization term $\Omega_p(\boldsymbol{u}_n) = \frac{\lambda}{n}\|\boldsymbol{u}_n\|$ into the prohection error and computes $\hat{\boldsymbol{\theta}}_n^{\lambda}$ instead of $\hat{\boldsymbol{\theta}}_n$. The performance of the l_2-regularized LSTD is studied in high-dimensional problems in the following Part.

Part II

Double Descent in Least-Squares Temporal Difference Learning

Temporal Difference (TD) algorithms are widely used in Deep Reinforcement Learning (RL). Their performance is heavily influenced by the size of the neural network. While in supervised learning, the regime of over-parameterization and its benefits are well understood, the situation in RL is much less clear. In this part, we present a theoretical analysis of the influence of network size and l_2-regularization on performance. We identify the ratio between the number of parameters and the number of visited states as a crucial factor and define over-parameterization as the regime when it is larger than one. Furthermore, we observe a double descent phenomenon, i.e., a sudden drop in performance around the parameter/state ratio of one. Leveraging random features and the lazy training regime, we study the regularized Least-Squared Temporal Difference (LSTD) algorithm in an asymptotic regime, as both the number of parameters and states go to infinity, maintaining a constant ratio. We derive deterministic limits of both the empirical, the true Mean-Squared Bellman Error (MSBE) and the true Mean-Squared Value Error (MSVE) that feature correction terms responsible for the double descent. Correction terms vanish when the l_2-regularization is increased or the number of unvisited states goes to zero. Numerical experiments with synthetic and small real-world environments closely match the theoretical predictions.

Part II is organized as follows:

- In Chapter 5, we start by presenting the classical bias-variance tradeoff theory, which shows that models with a large number of parameters and a near-zero training error tend to overfit and perform poorly on new data. Despite this, practitioners prefer using heavily parameterized models that interpolate the training data. In this chapter, we then briefly define and review the double descent theory introduced in supervised learning to reconcile the classical bias-variance tradeoff and modern practice. This chapter can be skipped by readers familiar with the phenomenon of double descent.
- In Chapter 6, we propose a novel theoretical framework for studying neural value function approximation in high-dimensional problems. In particular, we propose studying TD learning algorithms using neural networks in a novel double asymptotic regime, where both the number of parameters and states visited go to infinity and are comparable. Within the double asymptotic regime, we approximate TD learning algorithms using two-layer neural networks with the regularized Least-Squared Temporal Difference (LSTD) algorithm on random features by leveraging the lazy training regime.
- In Chapter 7, we first introduce the mathematical framework of Random Matrix Theory and concentrations results used to study the performance of regularized LSTD in the double asymptotic regime, and then we present our main theoretical results. In particular, we identify the resolvent of a non-symmetric positive-definite matrix that emerges as a crucial factor in the performance analysis of regularized LSTD. We provide a deterministic equivalent of this resolvent in the double asymptotic regime. Using the deterministic equivalent of the resolvent and concentration results, we analyze the performance of regularized LSTD in the double asymptotic regime with the derivation of deterministic equations for the asymptotic empirical Mean-Squared Bellman Error on the collected transitions, the asymptotic Mean-Squared Bellman Error (MSBE), and the asymptotic Mean-Squared Value Error (MSVE). The deterministic forms expose correction terms that arise from the double asymptotic regime. We show that the correction terms vanish as the l_2-regularization increases or the model complexity (i.e., the ratio between the number of parameters and number of states visited) goes to infinity. We also show that the influence of the l_2-regularization parameter decreases as the model complexity increases.

- In Chapter 8, after reviewing kernel methods and their Mercer feature spaces, we revisit the results of Chapter 7 in the Mercer feature space approximated by the random features. This reformulation enables us to rewrite all the results using a similar expression and highlights the connections that exist between the asymptotic error functions of random feature models and the corresponding errors of a regularized kernel LSTD predicator. In particular, this reformulation provides a better understanding of correction terms that arise from the double asymptotic regime and highlights an implicit regularization induced by the model complexity.

- In Chapter 9, we present our experimental results and show our theory closely matches empirical results for regularized LSTD on a range of both toy and small real-world environments; where both the number of states visited m and the number of parameters N are fixed, but for which our asymptotic predictions still gives accurate predictions. From our experiments, we identify two distinct regimes: an *under-parameterized regime* where $N/m < 1$ and an *over-parameterized regime* where $N/m > 1$. Each regime exhibits different behaviors in the empirical MSBE, the true MSBE, and the MSVE. Notably, in the phase transition around $N/m = 1$, we observe a double descent phenomenon similar to what has been reported in supervised learning, with a peak in the true MSBE and MSVE around $N/m = 1$. For the empirical MSBE and MSVE on the collected transitions, the phase transition is characterized by an almost zero training error and a perfect fit with the training data. We experimentally associate correction terms found in Chapter 7 and 8 with the double descent phenomenon. We also show that correction terms, and therefore the double descent phenomenon, empirically vanish when the number of unvisited states goes to zero or the level of regularization increases. Finally, we show that the discount factor has no influence on the double descent phenomenon.

This part is mainly based on our work *On Double Descent in Reinforcement Learning with LSTD and Random Features*, with Éloïse Berthier, David Filliat and Goran Frehse, accepted for publication in the *International Conference on Learning Representations (ICLR)*, 2024.

Chapter 5

Introduction to the Double Descent Phenomenon

In this chapter, we start in Section 5.1 by presenting the classical bias-variance tradeoff theory, which guided the selection of models and the choice of the number of parameters in traditional machine learning. This theory has been used to select models rich enough to express underlying structure in data and simple enough to avoid fitting of noise. Yet, as shown in Section 5.2, practitioners usually prefer using a large amount of parameters and interpolate the training data. We then briefly define and review the double descent theory introduced in supervised learning to bridge the gap between the traditional theory and the modern practice. Section 5.3 reviews the different asymptotic regimes considered for theoretical studies of over-parameterized models in supervised learning and reinforcement learning. Contributions and motivations of this part are summarized in Section 5.4 Note that Sections 5.1, 5.2, and 5.3 can be skipped by readers familiar with the double descent phenomenon.

5.1 Classical Bias-Variance Tradeoff

Machine Learning Problems. We assume we have a training dataset of n samples denoted by $\mathcal{D}_{\text{train}} := \{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_n, y_n)\}$, where each $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. Targets $\{y_i\}_{i=1}^n$ are generated by a function $f : \mathbb{R}^d \to \mathbb{R}$ such that, for all $i \in [n]$, we have

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i,$$

with the noise terms ϵ_i drawn from a distribution with zero mean and variance σ^2. We assume the training samples in $\mathcal{D}_{\text{train}}$ are drawn from a probability distribution P over $\mathbb{R}^d \times \mathbb{R}$.

In machine learning problems, the objective is to learn a predictor $\hat{f} : \mathbb{R}^d \to \mathbb{R}$ to approximate the function $f : \mathbb{R}^d \to \mathbb{R}$ using samples from the training dataset $\mathcal{D}_{\text{train}}$. The predictor $\hat{f} : \mathbb{R}^d \to \mathbb{R}$ can then be used to predict the output y of a new point $\boldsymbol{x}$, i.e., unseen during the training. The predictor $\hat{f}$ is commonly chosen from some function class $\mathcal{F}$, such as neural networks with certain architectures or linear models with hand-crafted features. During the learning, the objective is to

find the predictor $\hat{f} \in \mathcal{F}$ that minimizes the training error

$$\widehat{\mathcal{L}}(\hat{f}, \mathcal{D}_{\text{train}}) = \frac{1}{n} \sum_{i=1}^{n} l\big(\hat{f}(\boldsymbol{x}_i), y_i\big),$$

where $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is the *loss function*. Typically, we choose the *mean-squared loss* $l(\hat{f}(x), y) = \big(\hat{f}(x) - y\big)^2$ for *regression problems* or the *zero-one loss* $l(\hat{f}(x), y) = \mathbf{1}_{\hat{f}(x) \neq y}$ for *classification problems*. Ideally, we also want to find a predictor $\hat{f}$ that performs well on unseen data. To study the generalization, i.e., performance on unseen data, we typically study the *test* or the *generalization error* defined as

$$\mathcal{L}(\hat{f}, \mathcal{D}_{\text{train}}) = \mathbb{E}_{(\boldsymbol{x},y)\sim P}\big[l\big(\hat{f}(\boldsymbol{x}), y\big)\big].$$

Bias-Variance Decomposition. In regression problems using the Mean-Squared Error (MSE), we can decompose the test error as

$$\begin{aligned}
\mathcal{L}(\hat{f}, \mathcal{D}_{\text{train}}) &= \mathbb{E}_{(\boldsymbol{x},y)\sim P,\epsilon}\big[\big(\hat{f}(\boldsymbol{x}) - y\big)^2\big] \\
&= \mathbb{E}_{(\boldsymbol{x},y)\sim P,\epsilon}\big[\hat{f}(\boldsymbol{x})^2\big] + \mathbb{E}_{(\boldsymbol{x},y)\sim P,\epsilon}\big[y^2\big] - 2\mathbb{E}_{(\boldsymbol{x},y)\sim P,\epsilon}\big[\hat{f}(\boldsymbol{x})y\big] \\
&= \mathbb{E}_{(\boldsymbol{x},y)\sim P,\epsilon}\big[\hat{f}(\boldsymbol{x})^2\big] \pm \mathbb{E}_{(\boldsymbol{x},y)\sim P,\epsilon}\big[\hat{f}(\boldsymbol{x})\big]^2 + \mathbb{E}_{(\boldsymbol{x},y)\sim P,\epsilon}\big[y^2\big] \pm \mathbb{E}_{(\boldsymbol{x},y)\sim P,\epsilon}\big[y\big]^2 - 2\mathbb{E}_{(\boldsymbol{x},y)\sim P,\epsilon}\big[\hat{f}(\boldsymbol{x})y\big] \\
&= \mathrm{Var}\big[\hat{f}(\boldsymbol{x})\big] + \mathbb{E}_{(\boldsymbol{x},y)\sim P,\epsilon}\big[\hat{f}(\boldsymbol{x})\big]^2 + \mathrm{Var}\big[y\big] + \mathbb{E}_{(\boldsymbol{x},y)\sim P,\epsilon}\big[y\big]^2 - 2\mathbb{E}_{(\boldsymbol{x},y)\sim P,\epsilon}\big[\hat{f}(\boldsymbol{x})y\big] \\
&= \mathrm{Var}\big[\hat{f}(\boldsymbol{x})\big] + \mathbb{E}_{(\boldsymbol{x},y)\sim P,\epsilon}\big[\hat{f}(\boldsymbol{x})\big]^2 + \sigma^2 + \mathbb{E}_{(\boldsymbol{x},y)\sim P,\epsilon}\big[f(\boldsymbol{x})\big]^2 - 2\mathbb{E}_{(\boldsymbol{x},y)\sim P,\epsilon}\big[\hat{f}(\boldsymbol{x})f(\boldsymbol{x})\big] \\
&= \mathbb{E}_{(\boldsymbol{x},y)\sim P,\epsilon}\big[(\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x}))^2\big] + \mathrm{Var}\big[\hat{f}(\boldsymbol{x})^2\big] + \sigma^2,
\end{aligned} \tag{5.1}$$

where $\mathbb{E}_{(\boldsymbol{x},y)\sim P,\epsilon}\big[(\hat{f}(\boldsymbol{x}) - f(\boldsymbol{x}))^2\big]$ depicts the *bias*, i.e, the amount by which the average of our estimate differs from the true mean; $\mathrm{Var}\big[\hat{f}(\boldsymbol{x})\big]$ is the *variance*, i.e, the expected squared deviation of $\hat{f}(\boldsymbol{x})$ around its mean; and σ^2 is the *irreductible error*, i.e., the variance of the target around its true mean $f(\boldsymbol{x})$, which cannot be avoided no matter how well we estimate $f(\boldsymbol{x})$ (unless $\sigma = 0$). Note that a similar decomposition can be found for classification problems with the zero-one loss.

Bias-Variance Tradeoff. The bias and the variance in equation 5.1 can be controlled with the *capacity of the function class* or the *model complexity* $\mathcal{H}$. For parameterized models, the model complexity $\mathcal{H}$ is directly related to the number of parameters N. Typically, the variance increases and the bias decreases as the model complexity $\mathcal{H}$ increases. Indeed, complex models have more degrees of freedom, better fit the training data, and can adapt to more complicated underlying structures. The opposite behavior occurs when the model complexity $\mathcal{H}$ is decreased. To find a predictor $\hat{f} \in \mathcal{F}$ that approximates the target f, the objective is to minimize the generalization error $\mathcal{L}(\hat{f}, \mathcal{D}_{\text{train}})$ by finding the *"sweet spot"*, i.e., the model complexity $\mathcal{H}$ that balances the bias and the variance. The challenge stems from the fact that the training error $\widehat{\mathcal{L}}(\hat{f}, \mathcal{D}_{\text{train}})$ is not a good estimate of the generalization error $\mathcal{L}(\hat{f}, \mathcal{D}_{\text{train}})$, since it does not take into account properly the model complexity. Figure 5.1 highlights the typical behavior of the generalization and the training error as the model complexity $\mathcal{H}$ increases. The training error decreases as the model complexity increases, i.e., we better fit the training data. However, when the model *overfits* and fits too much the training data, the model captures the noise along with the underlying pattern in data. In such a scenario, it does not generalize well, has a large generalization error, and the predictions $\hat{f}(\boldsymbol{x})$

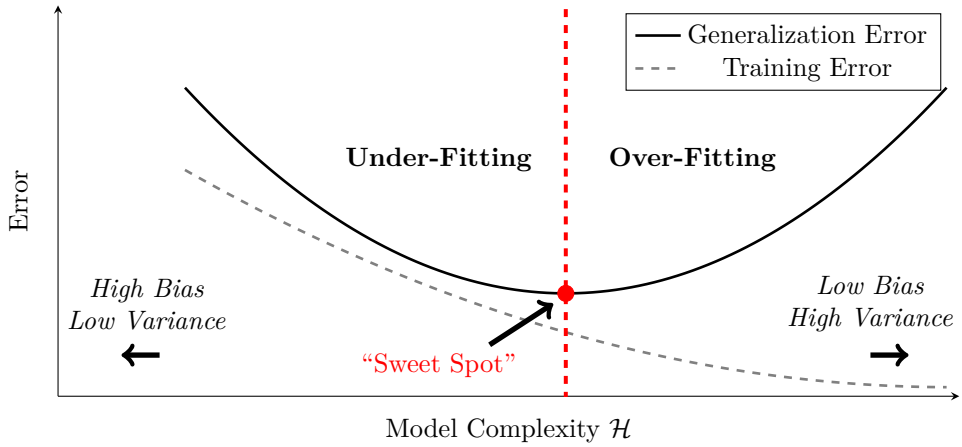


Figure 5.1: **Classical Bias-Variance Tradeoff.** As the model complexity $\mathcal{H}$ increases, the generalization error exhibits a U-shaped curve with a miminum at the sweet spot, whereas the training error is a decreasing function. The "sweet spot" is the balance between under-fitting and over-fitting.

have a large variance. On the other hand, when the models are not complex enough, they *underfit*, have a large bias, and generalize poorly. From this theory, a model with a zero training error that overfits the training data is expected to generalize poorly. Classical thinking is thus concerned with finding the "sweet spot" between under-fitting and over-fitting (Hastie et al., 2009).

5.2 The Double Descent Phenomenon

Modern Machine Learning Problems. Instead of finding the "sweet spot" of Figure 5.1, practitioners prefer using modern machine learning methods, such as huge neural network architectures or other non-linear predictors with very low or zero training error. With a high function capacity $\mathcal{H}$, those predictors perfectly fit the training data and perform well on unseen data. Those empirical observations guided the practitioners to choose huge neural network architectures to achieve zero training loss and interpolate the training data. Furthermore, empirical evidence indicates that neural networks and kernel machines trained to interpolate training data obtain near-optimal generalization results, even in the case where the training data are corrupted with high levels of noise (Zhang et al., 2021; Belkin et al., 2018b).

The Double Descent Phenomenon. To bridge the gap between theory and practice, Belkin et al. (2018a) reconcile the classical understanding and the modern practice within the "double descent" phenomenon described in Figure 5.2. For "small" model complexities $\mathcal{H}$, learned predictors are in the *under-parameterized regime* and exhibit the classical U-shaped curve depicted in Figure 5.1. When predictors fit too much the training data in the under-parameterized regime, the generalization error increases as the model capacity $\mathcal{H}$ increases. However, when the model complexity is higher than the *interpolation threshold*, i.e., when learned predictors perfectly fit the training data and are in the *over-parameterized regime*, increasing the model complexity leads to a decreasing generalization error. In the over-parameterized regime, the generalization error typically goes below the test error achieved at the sweet spot of the under-parameterized regime. A popular intuitive explanation of this phenomenon is that by considering larger function classes

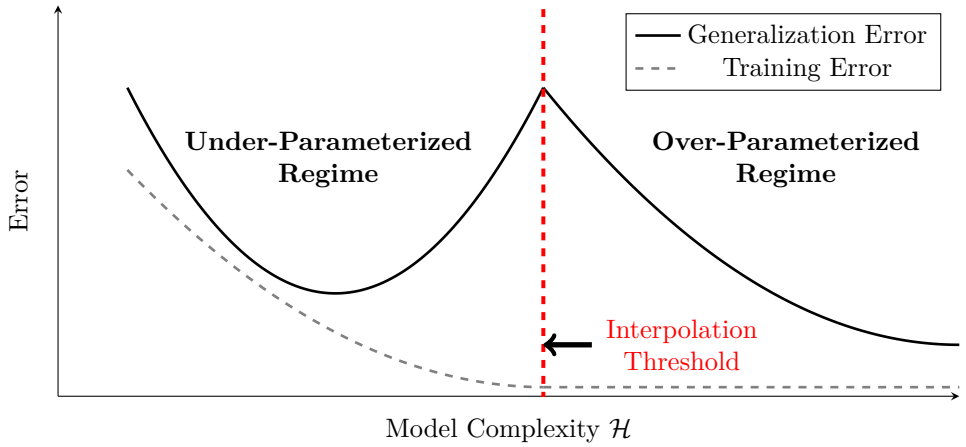


Figure 5.2: **The Double Descent Phenomenon.** As the model complexity $\mathcal{H}$ increases, the generalization error first shows the U-shaped curve depicted in Figure 5.1, peaking around the interpolation threshold. The double descent phenomenon refers to the decreasing behavior of the generalization error beyond the interpolation threshold, i.e, when predicators prefectly interpolate training data.

$\mathcal{F}$ that contain more candidate predictors compatible with the training data, we are also able to find interpolating functions that are "simpler" and are smoother to follow a form of Occam's razor (Belkin et al., 2018a). While the double descent phenomenon has been theoretically shown in asymptotic regimes, where neural networks are ideally approximated by linear models with random features (Louart et al., 2018; Mei and Montanari, 2022; Belkin et al., 2020; Liao et al., 2020; Jacot et al., 2020a;b; Canatar et al., 2021; Bach, 2024), the double descent phenomenon has also been observed in experiments with popular neural network architectures (Belkin et al., 2018a; Nakkiran et al., 2021). In addition to depending on the model complexity $\mathcal{H}$, the double descent phenomenon also depends on other dimensions such as the level of regularization (Mei and Montanari, 2022; Liao et al., 2020), the number of epochs (Nakkiran et al., 2021; Stephenson and Lee, 2021), or the data eigen-profile (Liu et al., 2021).

5.3 Asymptotic Regimes

As described in the previous section, modern machine learning problems using neural networks typically consider huge deep architectures with thousands or even billions of parameters. To bridge the gap between theory and practice and for mathematical convenience, a line of theoretical works consider asymptotic regimes where the number of parameters tends to infinity. This section reviews three related approaches to study huge neural network architectures in supervised learning and Reinforcement Learning.

Lazy Training regime. In the lazy training regime, one considers that infinitely wide neural networks, with appropriate scaling and initial conditions, behave like the linearization of the neural network around its initialization (Jacot et al., 2018; Chizat et al., 2019). However, as highlighted by Chizat et al. (2019), this behavior is not specific to neural networks and is not so much due to over-parameterization than an implicit choice of scaling. In such a scenario, neural networks can be modeled as linear models with random features (Rahimi and Recht, 2007). A recent line of

works (Jacot et al., 2018; Du et al., 2019; Bietti and Mairal, 2019; Fan and Wang, 2020; Golikov et al., 2022) has focused on the lazy training regime for studying the dynamics of neural networks with the neural tangent kernels (Jacot et al., 2020b). The lazy training regime was also considered in RL to prove the convergence of infinite-width neural TD learning algorithms towards the global optimum of the MSBE in both finite and infinite state spaces (Cai et al., 2019; Agazzi and Lu, 2022; Liu et al., 2019a).

Mean-Field regime. Under appropriate initial conditions and scaling, the mean-field analysis models the neural network and its induced feature representation with an empirical distribution, which, at the infinite-width limit, corresponds to a population distribution. The evolution of such a population distribution is characterized by a partial differential equation (PDE) known as the continuity equation and captures Stochastic Gradient Descent (SGD) dynamics as a Wasserstein gradient flow of the objective function (Chizat and Bach, 2018; Rotskoff and Vanden-Eijnden, 2018; Mei et al., 2018). Although more challenging than the NTK regime, the mean-field regime is more realistic since the weights are not restricted to staying in their initial regions (Chizat et al., 2019). The mean-field regime was studied in RL to prove the convergence of infinite-width neural TD learning algorithms towards the global optimum of the MSBE (Zhang et al., 2021; Agazzi and Lu, 2022).

Double Asymptotic regime. In the above regimes, the number of training samples n in $\mathcal{D}_{\text{train}}$ is negligible compared to the number of parameters as it grows to infinity. However, this is rarely the case in practice, particularly in modern machine learning problems, where we have to deal with a massive amount of high-dimensional data. For example, the popular ImageNet dataset (Russakovsky et al., 2015) contains typically more than $n = 500,000$ image samples of dimension $p = 256 \times 256 = 65,536$ in each class. Furthermore, we constantly face situations where those dimensions are comparable. For this reason, a line of theoretical studies in supervised learning (Louart et al., 2018; Mei and Montanari, 2022; Belkin et al., 2020; Liao et al., 2020; Jacot et al., 2020a;b; Canatar et al., 2021; Bach, 2024) considers a double asymptotic regime (Mei and Montanari, 2022; Louart et al., 2018; Liao et al., 2020; Belkin et al., 2020); where both the number of parameters, the number of samples n and their dimension go to infinity while maintaining their ratios constants. Since the number of parameters goes to infinity, the above works leverage the lazy training assumption to approximate neural networks as linear models with N random features and assume that N, n, p go to infinity while maintaining a constant ratio. In such a setting, the model complexity is defined as the ratio between the number of parameters N of the linear approximation and the number of samples n. Techniques from Random Matrix Theory or statistical physics can be used to show that the interpolation threshold is defined for $N/n = 1$ and to derive a precise description of the phase transition between under-($N/n < 1$) and over-($N/n > 1$) parameterization and the double descent phenomenon. In RL, Thomas (2022) investigated off-policy linear TD methods in the limit of large number of states and parameters on a transition matrix of rank 1 and observed a peaking behavior in the MSBE.

5.4 Motivations in Reinforcement Learning & Contributions

In recent years, neural networks have seen increased use in Reinforcement Learning (RL) (Mnih et al., 2015; Schulman et al., 2017; Haarnoja et al., 2018; Espeholt et al., 2018). While they can outperform traditional RL algorithms on challenging tasks, their theoretical understanding remains limited. Even for supervised learning, which can be considered a special case of RL with a discount factor equal to zero, deep neural networks are still far from being fully understood despite significant research efforts (Arora et al., 2019; Mei et al., 2018; Rotskoff and Vanden-Eijnden, 2018; Lee et al., 2019; Bietti and Mairal, 2019; Cao et al., 2019). The difficulty is further exacerbated in RL by a myriad of new challenges that limit the scope of these works, such as the absence of true targets or the non-i.i.d nature of the collected samples (Kumar et al., 2020; Luo et al., 2020; Lyle et al., 2021; Dong et al., 2020).

We decided to study theoretically the behavior of TD Learning methods in high-dimensional problems because they are widely used in practice as they are simple and efficient. In this part, we propose to study the l_2-regularized Least-squares Temporal Difference (LSTD) algorithm (Bradtke and Barto, 1996) presented in Section 4.5, which is easier to analyze since it doesn't use gradient descent, and because it converges to the same solution as other TD learning algorithms (Bradtke and Barto, 1996; Boyan, 1999; Berthier et al., 2022).

Theoretical studies of TD learning algorithms often explore high-dimensional problems in asymptotic regimes where the number of samples $n \to \infty$ while the number of model parameters N remains constant (Tsitsiklis and Van Roy, 1996; Bradtke and Barto, 1996; Nedić and Bertsekas, 2003; Sutton, 1988). When TD learning algorithms are applied to neural networks, it is commonly assumed that the number of parameters $N \to \infty$ with either a fixed or infinite number of samples without providing details on the relative magnitudes of those dimensions (Cai et al., 2019; Agazzi and Lu, 2022; Berthier et al., 2022; Xiao et al., 2021). Inspired by advancements in supervised learning (Louart et al., 2018; Liao et al., 2020), we apply Random Matrix tools and propose a novel *double asymptotic regime* where the number of parameters N and the number of *distinct visited* states m go to infinity, maintaining a constant ratio N/m, called *model complexity*. We use a linear model and nonlinear random features (RF) (Rahimi and Recht, 2007) to approximate an overparameterized single-hidden-layer network in the lazy training regime (Chizat et al., 2019). The results of our theoretical and empirical analyses are outlined below.

Contributions. We make the following contributions in this part, taking a step towards a better theoretical understanding of the influence of model complexity N/m and l_2-regularization on the performance of Temporal-Difference learning algorithms:

1. We propose a novel double asymptotic regime, where the number of parameters N and distinct visited states m go to infinity while maintaining a constant ratio. This leads to a precise assessment of the performance in both *over-parameterized* ($N/m > 1$) and *under-parameterized* regimes ($N/m < 1$). This is a nontrivial extension of existing work in supervised learning since several properties essential to proofs, such as the positive definiteness of key matrices, are voided by a discount factor in RL.

2. In the phase transition around $N/m = 1$, we observe a peak in the Mean-Squared Bellman Error (MSBE) and the Mean-Squared Value Error (MSVE), i.e, a *double descent phenomenon*

similar to what has been reported in supervised learning (Mei and Montanari, 2022; Liao et al., 2020).

3. We identify the resolvent of a non-symmetric positive-definite matrix that emerges as a crucial factor in the performance analysis of TD learning algorithms in terms of the error functions and we provide its deterministic equivalent form in the double asymptotic regime.

4. We derive analytical equations for the asymptotic empirical MSBE on the collected transitions, the asymptotic true MSBE, and the asymptotic MSVE. The deterministic forms expose correction terms that we experimentally associate with the double descent phenomenon. We show that the correction terms vanish as the l_2-regularization is increased or N/m goes to infinity. We also show that the influence of the l_2-regularization parameter decreases as N/m increases.

5. We show that the asymptotic errors studied can be expressed as the sum of the corresponding error terms of a regularized kernel LSTD predictor, with implicit l_2-regularization parameter $\tilde{\lambda}$ induced by the ratio N/m, and a second-order correction factor.

6. Our theory closely matches empirical results on a range of both toy and small real-world Markov Reward Processes where m and N are fixed, but for which the asymptotic regime still gives accurate predictions. Notably, we observe a peak in the true MSBE and MSVE around $N/m = 1$ that is not observed in the empirical MSBE and MSVE. Correction terms, and therefore the difference between true and empirical MSBE, empirically vanish when the number of unvisited states goes to zero.

Chapter 6

Regularized LSTD with Random Features in High-Dimensional Problems

This chapter proposes a theoretical framework for studying neural value function approximation in high-dimensional problems. In particular, we propose studying TD learning algorithms using neural networks in a double asymptotic regime, where both the number of parameters and states go to infinity while maintaining a constant ratio called model complexity. Before introducing this double asymptotic regime, we revisit and recall the framework of linear function approximation in Markov Reward Processes in Section 6.1. In Section 6.2, we present the l_2-regularized Least-Squared Temporal Difference (LSTD) algorithm with random features. Section 6.3 formally introduces the double asymptotic regime in which, by leveraging random features and the lazy training regime, we approximate TD learning algorithms using a two-layer neural networks with the regularized Least-Squared Temporal Difference (LSTD) algorithm on random features.

6.1 Linear Function Approximation in Markov Reward Processes

Markov Reward Processes. In the context of function approximation for value-based algorithms, the behavior of a fixed policy π within an MDP is characterized by a Markov Reward Process (MRP) $(\mathcal{S}, P^\pi, R^\pi, \mu_0)$ properly defined in Definition 3.1.1, where $\mathcal{S} \subseteq \mathbb{R}^d$ is the state space; $P^\pi : \mathcal{S} \times \mathcal{S} \to [0,1]$ is the transition kernel (stochastic kernel) for which $P^\pi(\boldsymbol{s}, \boldsymbol{s}')$ denotes the probability of transitioning to state $\boldsymbol{s}'$ from state $\boldsymbol{s}$; $R^\pi : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ is the reward function; and $\mu_0 \in \mathcal{P}(\mathcal{S})$ is the initial state distribution of the MDP. For notational convenience, the state space $\mathcal{S}$ is described by the state matrix $\boldsymbol{S} \in \mathbb{R}^{d \times |\mathcal{S}|}$, where each column $i \in |\mathcal{S}|$ of $\boldsymbol{S}$ is denoted by $\boldsymbol{S}_i$ and represents a state in $\mathcal{S}$. The transition probability matrix associated with the stochastic kernel P^π is denoted by $\boldsymbol{P}^\pi \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$. The objective is to learn the value function $V^\pi : \mathcal{S} \to \mathbb{R}$, which maps each state $\boldsymbol{s}$ to the expected discounted sum of rewards when starting from a state $\boldsymbol{s} \in \mathcal{S}$ and

following the dynamics of the MRP defined by $\boldsymbol{P}^\pi$ as

$$V^\pi(\boldsymbol{s}) := \mathbb{E}_{\boldsymbol{P}^\pi}\Big[\sum_{k=1}^{\infty} \gamma^{k-1} R^\pi(\boldsymbol{s}_k, \boldsymbol{s}_{k+1}) \;\Big|\; \boldsymbol{s}_1 = \boldsymbol{s}\Big],$$

where $\gamma \in [0, 1)$ is the discount factor. The value function is the unique fixed-point of the Bellman equation (equation 4.9)

$$\boldsymbol{V}^\pi = \bar{\boldsymbol{r}}^\pi + \gamma \boldsymbol{P}^\pi \boldsymbol{V}^\pi, \tag{6.1}$$

where $\boldsymbol{V}^\pi \in \mathbb{R}^{|\mathcal{S}|}$ is the vector representation of $V^\pi : \mathcal{S} \to \mathbb{R}$, for which the i-th element $\boldsymbol{V}_i^\pi$ is equal to $V^\pi(\boldsymbol{S}_i)$; and $\bar{\boldsymbol{r}}^\pi \in \mathbb{R}^{|\mathcal{S}|}$ is the vector containing the expected rewards, for which $\bar{\boldsymbol{r}}_i^\pi = \bar{R}^\pi(\boldsymbol{S}_i) = \mathbb{E}_{\boldsymbol{s}' \sim \boldsymbol{P}^\pi}[R^\pi(\boldsymbol{S}_i, \boldsymbol{s}')]$ for all $i \in [\,|\mathcal{S}|\,]$.

Linear Function Approximation. In practice, equation 6.1 cannot be solved since $\boldsymbol{P}^\pi$ is unknown and $|\mathcal{S}|$ is too large. One common solution is to use Linear Function Approximation methods (LFA) introduced in Section 3. Using a parameter vector $\boldsymbol{\theta} \in \mathbb{R}^N$ and a feature matrix $\boldsymbol{\Sigma}_{\mathcal{S}} \in \mathbb{R}^{N \times |\mathcal{S}|}$, whose columns are the feature vectors for every state, the objective of LFA methods is to approximate $\boldsymbol{V}^\pi$ as $\boldsymbol{V}^\pi \approx \boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\theta}$. For a given feature matrix, the learning process based on equation 6.1 amounts to finding a parameter vector $\boldsymbol{\theta}$ that minimizes the Mean-Squared Bellman error (Section 3.2) defined as

$$\mathrm{MSBE}(\boldsymbol{\theta}) = \|\bar{\boldsymbol{r}}^\pi + \gamma \boldsymbol{P}^\pi \boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\theta} - \boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\theta}\|^2_{\boldsymbol{D}_{\boldsymbol{\mu}^\pi}}, \tag{6.2}$$

where $\boldsymbol{\mu}^\pi \in \mathbb{R}^{|\mathcal{S}|}$ is the stationary distribution induced by the MRP $(\mathcal{S}, P^\pi, R^\pi, \mu_0)$ and $\boldsymbol{D}_{\boldsymbol{\mu}^\pi} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ is its diagonal matrix. Since $\bar{\boldsymbol{r}}^\pi + \gamma \boldsymbol{P}^\pi \boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\theta}$ may not lie in the span of the bases $\boldsymbol{\Sigma}_{\mathcal{S}}$, there may not be a parameter vector $\boldsymbol{\theta}$ that brings the MSBE to zero.

Linear Temporal-Difference Methods. Linear Temporal-Difference (TD) learning methods presented in Section 4.2.1 are LFA methods that aim to minimize the MSBE in equation 6.2 by replacing the second occurrence of $\boldsymbol{\theta}$ in equation 6.2 with an auxiliary vector $\boldsymbol{u}$, minimizing on $\boldsymbol{u}$ and then finding a $\boldsymbol{\theta}$ close to $\boldsymbol{u}$ (Dann et al., 2014):

$$\boldsymbol{u}^* = \operatorname*{arg\,min}_{\boldsymbol{u} \in \mathbb{R}^N} \|\bar{\boldsymbol{r}}^\pi + \gamma \boldsymbol{P}^\pi \boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\theta}^* - \boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{u}\|^2_{\boldsymbol{D}_{\boldsymbol{\mu}}^\pi} \qquad \text{(projection step)}, \tag{6.3}$$

$$\boldsymbol{\theta}^* = \operatorname*{arg\,min}_{\boldsymbol{\theta} \in \mathbb{R}^N} \|\boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{u}^* - \boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\theta}\|^2_{\boldsymbol{D}_{\boldsymbol{\mu}}^\pi} \qquad \text{(fixed-point step)}. \tag{6.4}$$

The projection step (equation 6.3) implies that TD learning methods actually minimize the Mean-Squared Projected Bellman error (MSPBE) rather than the MSBE (see Section 3.2). In the following section, we describe the key elements on which we base our asymptotic analysis of the MSBE in TD learning algorithms: random features, the regularized LSTD algorithm, and the double asymptotic regime.

6.2 Regularized LSTD with Random Features

Random Features. We consider value function approximation using the random feature mapping $\mathrm{RF} : \mathcal{S} \to \mathbb{R}^N$ defined for all $\boldsymbol{s} \in \mathcal{S}$ as

$$\mathrm{RF}(\boldsymbol{s}) = \sigma(\boldsymbol{W}\boldsymbol{s}), \tag{6.5}$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ is K_σ-Lipschitz continuous and applied component-wise; $\boldsymbol{W} = \varphi(\tilde{\boldsymbol{W}}) \in \mathbb{R}^{N\times d}$ is a random weight matrix fixed throughout training, for which $\tilde{\boldsymbol{W}} \in \mathbb{R}^{N\times d}$ has independent and identically distributed $\mathcal{N}(0,1)$ entries and $\varphi : \mathbb{R} \to \mathbb{R}$ is K_φ-Lipschitz continuous and applied component-wise. From the perspective of neural networks, the N random features can be interpreted as N outputs from a single-hidden-layer neural network. Indeed, in asymptotic regimes where the number of features N of the single layer grows towards infinity, this simplification becomes even more accurate as we enter into the *lazy training regime* — where weights of the hidden layer barely deviate from their random initial values (Chizat et al., 2019). In the literature, the lazy training regime is often considered to approximate large-width single-hidden-layer neural networks as linear models of random features (Louart et al., 2018; Liao et al., 2020; Mei and Montanari, 2022), including in RL (Cai et al., 2019; Agazzi and Lu, 2022; Liu et al., 2019a). The use of random features is also popular on theoretical Kernel Ridge Regression (KRR) works (Jacot et al., 2020b; Canatar et al., 2021; Bordelon et al., 2020; Simon et al., 2023a;b); where the Gaussian assumption is leveraged to interpret the KRR as a linear model of Gaussian random features. In the KRR works, the Gaussian assumption is leveraged to approximate the Mercer feature map with random features (see Section 8.1.1 for further details on the Mercer feature map). In the following, we denote the random feature matrix of any state matrix $\boldsymbol{A} \in \mathbb{R}^{d\times p}$ as $\boldsymbol{\Sigma}_{\boldsymbol{A}}$ where RF is applied column-wise, i.e.,

$$\boldsymbol{\Sigma}_{\boldsymbol{A}} = \sigma(\boldsymbol{W}\boldsymbol{A}).$$

Sample Matrices and Empirical MSBE. We assume that the transition probability matrix $\boldsymbol{P}^\pi$ is unknown during the training phase. Instead, we have a dataset of n transitions consisting of states, rewards, and next-states drawn from the MRP, i.e., we have $\mathcal{D}_{\text{train}} := \{(\boldsymbol{s}_i, r_i, \boldsymbol{s}'_i)\}_{i=1}^n$ where $\boldsymbol{s}'_i \sim P^\pi(\boldsymbol{s}_i)$ and $r_i = R^\pi(\boldsymbol{s}_i, \boldsymbol{s}'_i)$. We consider the *on-policy setting*, where $\mathcal{D}_{\text{train}}$ is derived from a sample path of the MRP or its stationary distribution $\boldsymbol{\mu}^\pi$. We collect the states and rewards in the sample matrices

$$\boldsymbol{X}_n = [\boldsymbol{s}_1, \dots, \boldsymbol{s}_n] \in \mathbb{R}^{d\times n}, \quad \boldsymbol{r} = [r_1, \dots, r_n]^T \in \mathbb{R}^n, \quad \boldsymbol{X}'_n = [\boldsymbol{s}'_1, \dots, \boldsymbol{s}'_n] \in \mathbb{R}^{d\times n}. \tag{6.6}$$

Let $\hat{\mathcal{S}} \subseteq \mathcal{S}$ be the set of distinct states in $\mathcal{D}_{\text{train}}$, which we call *visited states*, and let $m = |\hat{\mathcal{S}}|$ be the number of distinct visited states. We denote by $\hat{\boldsymbol{S}} \in \mathbb{R}^{d\times m}$ the state matrix of $\hat{\mathcal{S}}$, where each column $\hat{\boldsymbol{S}}_i$ of $\hat{\boldsymbol{S}}$ describes a state in $\hat{\mathcal{S}}$. $\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \in \mathbb{R}^{N\times m}$, $\boldsymbol{\Sigma}_{\boldsymbol{X}_n} \in \mathbb{R}^{N\times n}$, and $\boldsymbol{\Sigma}_{\boldsymbol{X}'_n} \in \mathbb{R}^{N\times n}$ depict the random feature matrices of $\hat{\mathcal{S}}$, $\boldsymbol{X}_n$, and $\boldsymbol{X}'_n$, respectively. For the proof of our results, it will be mathematically advantageous to express $\boldsymbol{\Sigma}_{\boldsymbol{X}_n}$ and $\boldsymbol{\Sigma}_{\boldsymbol{X}'_n}$ as the product of $\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}$ with auxiliary matrices $\hat{\boldsymbol{U}}_n \in \mathbb{R}^{m\times n}$ and $\hat{\boldsymbol{V}}_n \in \mathbb{R}^{m\times n}$ as follows:

$$\boldsymbol{\Sigma}_{\boldsymbol{X}_n} = \sqrt{n}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n \quad \text{and} \quad \boldsymbol{\Sigma}_{\boldsymbol{X}'_n} = \sqrt{n}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{V}}_n\,. \tag{6.7}$$

Each column i of $\sqrt{n}\hat{\boldsymbol{U}}_n$ is a one-hot vector, where the j-th element equals 1 if the i-th state $\boldsymbol{s}_i$ of $\boldsymbol{X}_n$ is $\hat{\boldsymbol{S}}_j$, and similarly for $\sqrt{n}\hat{\boldsymbol{V}}_n$ and $\boldsymbol{X}'_n$. Since $\boldsymbol{P}^\pi$ is unknown, we want to find a parameter vector $\boldsymbol{\theta} \in \mathbb{R}^N$ that minimizes the empirical version of the MSBE (equation 6.2) obtained with transitions collected in $\mathcal{D}_{\text{train}}$:

$$\widehat{\text{MSBE}}(\boldsymbol{\theta}) = \tfrac{1}{n}\|\boldsymbol{r} + \gamma\boldsymbol{\Sigma}_{\boldsymbol{X}'_n}^T\boldsymbol{\theta} - \boldsymbol{\Sigma}_{\boldsymbol{X}_n}^T\boldsymbol{\theta}\|^2, \tag{6.8}$$

which uses the Euclidean norm since the distribution is reflected by the samples. Assuming globally stable MRP, a fixed number of features, and all states being visited, $\widehat{\text{MSBE}}(\boldsymbol{\theta})$ converges to

MSBE($\boldsymbol{\theta}$) with probability 1 as the number of collected transitions $n \to \infty$ (Nedić and Bertsekas, 2003). This follows from the law of large numbers (Stachurski, 2009). In our analysis, we will also consider the case where $n \to \infty$ without visiting all states, i.e., with $m < |\mathcal{S}|$ such that there can be a significant difference between $\widehat{\text{MSBE}}(\boldsymbol{\theta})$ and MSBE($\boldsymbol{\theta}$).

L_2-Regularized Least-Square Temporal-Difference Methods. The l_2-Regularized Least-Square Temporal-Difference (LSTD) algorithm introduced in Section 4.5 is a linear TD learning method that solves an empirical regularized version of equation 6.3 and 6.4 with the n transitions collected in $\mathcal{D}_{\text{train}}$ as

$$\boldsymbol{u}_n^\lambda = \underset{\boldsymbol{u}_n \in \mathbb{R}^N}{\arg\min} \left(\|\boldsymbol{r} + \gamma \boldsymbol{\Sigma}_{\boldsymbol{X}'_n}^T \hat{\boldsymbol{\theta}}_n^\lambda - \boldsymbol{\Sigma}_{\boldsymbol{X}_n}^T \boldsymbol{u}_n\|^2 + \lambda mn \|\boldsymbol{u}_n\|^2 \right), \tag{6.9}$$

$$\hat{\boldsymbol{\theta}}_n^\lambda = \underset{\boldsymbol{\theta}_n \in \mathbb{R}^N}{\arg\min} \ \|\boldsymbol{\Sigma}_{\boldsymbol{X}_n}^T \boldsymbol{u}_n^\lambda - \boldsymbol{\Sigma}_{\boldsymbol{X}_n}^T \boldsymbol{\theta}_n\|^2, \tag{6.10}$$

where $\lambda > 0$ is the *l_2-regularization parameter* introduced to mitigate overfitting (Hoffman et al., 2011; Chen et al., 2013). It is well known that for $\lambda = 0$, the fixed point $\hat{\boldsymbol{\theta}}_n^\lambda$ of the approximation equation 6.9 and 6.10 converges to the fixed point $\boldsymbol{\theta}^*$ of equation 6.3 and 6.4 with probability one as the number of samples $n \to \infty$ (Nedić and Bertsekas, 2003). As shown in Section 4.5, solving the fixed-point of the linear system approximation given by equation 6.9 and 6.10 gives

$$\hat{\boldsymbol{\theta}}_n^\lambda = \left[\boldsymbol{\Sigma}_{\boldsymbol{X}_n} \left[\boldsymbol{\Sigma}_{\boldsymbol{X}_n} - \gamma \boldsymbol{\Sigma}_{\boldsymbol{X}'_n} \right]^T + \lambda mn \boldsymbol{I}_N \right]^{-1} \boldsymbol{\Sigma}_{\boldsymbol{X}_n} \boldsymbol{r}. \tag{6.11}$$

Under appropriate learning rates, linear TD learning methods based on gradient-descent converge towards the same fixed-point $\hat{\boldsymbol{\theta}}_n^\lambda$ (Robbins and Monro, 1951; Dann et al., 2014; Sutton and Barto, 2018). Besides reducing overfitting, l_2-regularized LSTD with an appropriate λ ensures in practice that $\boldsymbol{\Sigma}_{\boldsymbol{X}_n}[\boldsymbol{\Sigma}_{\boldsymbol{X}_n} - \gamma \boldsymbol{\Sigma}_{\boldsymbol{X}'_n}]^T + \lambda mn \boldsymbol{I}_N$ is invertible. Note that LSTD with random features has also been considered in Ghavamzadeh et al. (2010) for high-dimensional spaces, where the number of features is bigger than the number of samples collected. Authors show that random features act as regularizer and prevent overfitting.

6.3 Double Asymptotic Regime & Resolvent in LSTD

We study the l_2-regularized LSTD in the following double asymptotic regime:

Assumption 1 (Double Asymptotic Regime)**.** *As $N, m, d \to \infty$, we have:*

1. $0 < \liminf \left\{ \frac{N}{m}, \frac{d}{m} \right\} < \limsup \left\{ \frac{N}{m}, \frac{d}{m} \right\} < \infty$.
2. *There exists $K_{\mathcal{S}}, K_r > 0$ such that $\limsup_{|\mathcal{S}|} \|\boldsymbol{S}\| < K_{\mathcal{S}}$ and $R^\pi(\cdot, \cdot)$ is bounded by K_r.*

Remark 10. *As mentioned in the previous section, we leverage the lazy training regime to approximate TD learning algorithms using two-layer neural networks by a linear value approximation on random features with the regularized LSTD algorithm. Indeed, linear models with random features approximate single-hidden-layer neural networks in the lazy training regime, and the solution returned by regularized LSTD is the solution on which converge TD learning with a stochastic gradient based approach under appropriate learning rates.*

In order to use Random Matrix tools, we rewrite equation 6.11 as (see proof in Lemma A.8.8)

$$\hat{\boldsymbol{\theta}}_n^\lambda = \tfrac{1}{mn}\boldsymbol{\Sigma}_{\boldsymbol{X}_n}\left[\tfrac{1}{mn}\left[\boldsymbol{\Sigma}_{\boldsymbol{X}_n} - \gamma\boldsymbol{\Sigma}_{\boldsymbol{X}'_n}\right]^T\boldsymbol{\Sigma}_{\boldsymbol{X}_n} + \lambda\boldsymbol{I}_n\right]^{-1}\boldsymbol{r}. \tag{6.12}$$

We observe that $\hat{\boldsymbol{\theta}}_n^\lambda = \frac{1}{mn}\boldsymbol{\Sigma}_{\boldsymbol{X}_n}\boldsymbol{Q}_m(\lambda)\boldsymbol{r}$ depends on the random *resolvent*

$$\boldsymbol{Q}_m(\lambda) = \left[\tfrac{1}{mn}\left[\boldsymbol{\Sigma}_{\boldsymbol{X}_n} - \gamma\boldsymbol{\Sigma}_{\boldsymbol{X}'_n}\right]^T\boldsymbol{\Sigma}_{\boldsymbol{X}_n} + \lambda\boldsymbol{I}_n\right]^{-1} = \left[\tfrac{1}{m}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n + \lambda\boldsymbol{I}_n\right]^{-1}, \tag{6.13}$$

when $\frac{1}{m}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n + \lambda\boldsymbol{I}_n$ is invertible, which in general may not be the case. We can guarantee invertibility if the *empirical transition model matrix* $\hat{\boldsymbol{A}}_m \in \mathbb{R}^{m\times m}$

$$\hat{\boldsymbol{A}}_m = \hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \tag{6.14}$$

has a symmetric positive-definite part (see Appendix A.5 for a formal proof). For the remainder of the paper, we therefore make the following assumption on $\hat{\boldsymbol{A}}_m$:

Assumption 2 (Bounded Eigenspectrum)**.** *There exist* $0 < \xi_{\min} < \xi_{\max}$ *such that for every* m*, all the eigenvalues of* $H(\hat{\boldsymbol{A}}_m)$ *are in* $[\xi_{\min}, \xi_{\max}]$*.*

Assumption 2 is satisfied for the l_2-regularized pathwise LSTD (Lazaric et al., 2012), and may also be valid for sufficiently large n (see Appendix A.5).

Remark 11. *The empirical transition model matrix* $\hat{\boldsymbol{A}}_m$ *depicts a model-based interpretation similar to that presented in Section 4.4. Indeed,* $\hat{\boldsymbol{A}}_m$ *contains an empirical model of the transition probabilities as*

$$\hat{\boldsymbol{A}}_m = \tfrac{1}{n}\left[\hat{\boldsymbol{C}}_n - \gamma\hat{\boldsymbol{N}}_n\right], \tag{6.15}$$

where $\hat{\boldsymbol{C}}_n = n\hat{\boldsymbol{U}}_n\hat{\boldsymbol{U}}_n^T \in \mathbb{R}^{m\times m}$ *is a diagonal matrix containing the state visit counts of each state* $\boldsymbol{s} \in \hat{\mathcal{S}}$ *in the dataset* $\mathcal{D}_{train} := \{(\boldsymbol{s}_i, r_i, \boldsymbol{s}'_i)\}_{i=1}^n$*; the elements* $\left[\hat{\boldsymbol{N}}_n\right]_{ij}$ *of the matrix* $\hat{\boldsymbol{N}}_n = n\hat{\boldsymbol{U}}_n\hat{\boldsymbol{V}}_n^T \in \mathbb{R}^{m\times m}$ *contain the number of times a transition from state* $\hat{\boldsymbol{S}}_i$ *to state* $\hat{\boldsymbol{S}}_j$ *has been observed in* $\mathcal{D}_{train}$*. Under Assumption 2, the diagonal matrix* $\hat{\boldsymbol{C}}_n$ *is invertible and we can define the matrix* $\hat{\boldsymbol{P}}_n = \hat{\boldsymbol{C}}_n^{-1}\hat{\boldsymbol{N}}_n \in \mathbb{R}^{m\times m}$*. It can be shown that* $\hat{\boldsymbol{P}}_n$ *is a transition probability matrix. From the sample collected in* $\mathcal{D}_{train}$ *and from Assumption 2, we can construct the empirical MRP* $(\hat{\mathcal{S}}, \hat{\boldsymbol{P}}_n, R^\pi, \mu_0)$ *and write* $\hat{\boldsymbol{A}}_m$ *as*

$$\hat{\boldsymbol{A}}_m = \boldsymbol{D}_{\hat{\boldsymbol{\mu}}_n}\left[\boldsymbol{I}_m - \gamma\hat{\boldsymbol{P}}_n\right], \tag{6.16}$$

where $\boldsymbol{D}_{\hat{\boldsymbol{\mu}}_n} = \frac{1}{n}\hat{\boldsymbol{C}}_n = \hat{\boldsymbol{U}}_n\hat{\boldsymbol{U}}_n^T$*. Note that diagonal elements of* $\boldsymbol{D}_{\hat{\mu}_n}$ *define the stationary distribution* $\hat{\boldsymbol{\mu}}_n \in \mathbb{R}^m$ *of the transition probability matrix* $\hat{\boldsymbol{P}}_n$*. The regularized LSTD solution* $\hat{\boldsymbol{\theta}}_n^\lambda$ *(equation 6.12) on* $\mathcal{D}_{train}$ *can be rewritten as*

$$\begin{aligned}\hat{\boldsymbol{\theta}}_n^\lambda &= \tfrac{1}{\sqrt{n}}\left[\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\boldsymbol{D}_{\hat{\boldsymbol{\mu}}_n}\left[\boldsymbol{I}_m - \gamma\hat{\boldsymbol{P}}_n\right]\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T + \lambda\boldsymbol{I}_N\right]^{-1}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{r} \\ &= \left[\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\boldsymbol{D}_{\hat{\boldsymbol{\mu}}_n}\left[\boldsymbol{I}_m - \gamma\hat{\boldsymbol{P}}_n\right]\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T + \lambda\boldsymbol{I}_N\right]^{-1}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\boldsymbol{D}_{\hat{\mu}_n}\hat{\bar{\boldsymbol{r}}}_n;\end{aligned}$$

where $\left[\hat{\bar{\boldsymbol{r}}}_n\right]_i = \sum_{j=1}^m [\hat{\boldsymbol{P}}_n]_{ij}R^\pi(\hat{\boldsymbol{S}}_i, \hat{\boldsymbol{S}}_j)$*, for all* $i \in [m]$*. From equation 4.10,* $\hat{\boldsymbol{\theta}}_n^\lambda$ *can be thus interpreted as the asymptotic solution on the empirical MRP of regularized LSTD when each state* $\boldsymbol{s} \in \hat{\mathcal{S}}$ *is visited infinitely often and is visited in the long run with probability* 1 *in proportion* $\hat{\mu}_n(\boldsymbol{s})$*.*

Chapter 7

Main Results in High-Dimensional Problems

In this chapter, we first introduce the mathematical framework of Random Matrix Theory and concentration results used to study the performance of regularized LSTD in the double asymptotic regime of Assumption 2, and then we present our main theoretical results. In Section 7.1, we start by highlighting challenges encountered in studying random matrices in high-dimensional problems and introduce the Random Matrix Theory framework. In Section 7.2, we identify the resolvent $\boldsymbol{Q}_m(\lambda)$ of a non-symmetric positive-definite matrix that emerges as a crucial factor in the performance analysis of regularized LSTD, and we provide a deterministic equivalent of this resolvent in the double asymptotic regime. In the following, using the deterministic equivalent of the resolvent $\boldsymbol{Q}_m(\lambda)$ and concentration results, we analyze the performance of regularized LSTD in the double asymptotic regime of Assumption 2 with the derivation of deterministic equations for the asymptotic empirical Mean-Squared Bellman Error on the collected transitions in Section 7.3, the asymptotic Mean-Squared Bellman Error (MSBE) in Section 7.4, and the asymptotic Mean-Squared Value Error (MSVE) in Section 7.5. In particular, we expose correction terms in those deterministic forms that arise from the double asymptotic regime. We show that the correction terms vanish as the l_2-regularization increases or the model complexity N/m (i.e., the ratio between the number of parameters N and the number of states visited m) goes to infinity. We also show that the influence of the l_2-regularization parameter decreases as the model complexity N/m increases.

7.1 Pitfalls of High-Dimensional Problems & Deterministic Equivalent

7.1.1 Counterintuitive Phenomenon in High-Dimensional Problems

As the number of parameters $N \to \infty$, it can be shown that entry-wise of the Gram matrix $\frac{1}{N}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \in \mathbb{R}^{m\times m}$ converges almost surely to the kernel matrix

$$\boldsymbol{\Phi}_{\hat{\mathcal{S}}} = \mathbb{E}_{\boldsymbol{w}}\left[\sigma\big(\boldsymbol{w}^T\hat{\boldsymbol{S}}\big)^T\sigma\big(\boldsymbol{w}^T\hat{\boldsymbol{S}}\big)\right] = \mathbb{E}_{\tilde{\boldsymbol{w}}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}\left[\sigma\big(\varphi(\tilde{\boldsymbol{w}})^T\hat{\boldsymbol{S}}\big)^T\sigma\big(\varphi(\tilde{\boldsymbol{w}})^T\hat{\boldsymbol{S}}\big)\right]. \tag{7.1}$$

This follows from the strong law of large numbers, which particularly states that as $N \to \infty$, for any $i, j \in [m]$,

$$\left[\tfrac{1}{N}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\right]_{ij} = \tfrac{1}{N}\sigma\big(\boldsymbol{W}^T\hat{\boldsymbol{S}}_i\big)^T\sigma\big(\boldsymbol{W}^T\hat{\boldsymbol{S}}_j\big) \xrightarrow{a.s} \mathbb{E}_{\boldsymbol{w}}\left[\sigma\big(\boldsymbol{w}^T\hat{\boldsymbol{S}}_i\big)\sigma\big(\boldsymbol{w}^T\hat{\boldsymbol{S}}_j\big)\right] = \left[\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\right]_{ij}. \tag{7.2}$$

While $\frac{1}{N}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \to \boldsymbol{\Phi}_{\hat{\mathcal{S}}}$ holds in the asymptotic $N \to \infty$ limit, the situation becomes more subtle when $N, m \to \infty$ and N and m are comparable. The entry-wise convergence of equation 7.2 remains valid, but the convergence of $\left\|\frac{1}{N}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} - \boldsymbol{\Phi}_{\hat{\mathcal{S}}}\right\|$ no longer holds in the operator norm, due to the large factor m in the following norm inequality

$$\underbrace{\left\|\tfrac{1}{N}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} - \boldsymbol{\Phi}_{\hat{\mathcal{S}}}\right\|_\infty}_{\to\, 0} \le \left\|\tfrac{1}{N}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} - \boldsymbol{\Phi}_{\hat{\mathcal{S}}}\right\| \le \underbrace{m\left\|\tfrac{1}{N}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} - \boldsymbol{\Phi}_{\hat{\mathcal{S}}}\right\|_\infty}_{\to\, ?}.$$

From the inequality above, we can not upper bound anymore $\left\|\frac{1}{N}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} - \boldsymbol{\Phi}_{\hat{\mathcal{S}}}\right\|$ when $N, m \to \infty$, and

$$\left\|\tfrac{1}{N}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} - \boldsymbol{\Phi}_{\hat{\mathcal{S}}}\right\| \not\to 0,$$

as $N, m \to \infty$ for $N/m \to c < \infty$ with $c > 0$. We can not guarantee anymore the convergence of $\left\|\frac{1}{N}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} - \boldsymbol{\Phi}_{\hat{\mathcal{S}}}\right\|$ under the operator norm.

7.1.2 The Empirical Covariance Matrix Example

To illustrate the non-convergence of random matrices for the operator norm in high-dimensional problems, a popular example found in the literature of Random Matrix Theory is the *empirical covariance matrix example* (Marchenko and Pastur, 1967; Couillet and Liao, 2022). In this example, we consider the empirical covariance matrix $\hat{\boldsymbol{C}}_N \in \mathbb{R}^{m\times m}$ defined as

$$\hat{\boldsymbol{C}}_N = \tfrac{1}{N}\boldsymbol{X}\boldsymbol{X}^T = \tfrac{1}{N}\sum_{i=1}^N \boldsymbol{x}_i\boldsymbol{x}_i^T; \tag{7.3}$$

where $\boldsymbol{X} = \big[\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\big] \in \mathbb{R}^{m\times N}$ is composed of N independent and identically distributed observations from a m-dimensional Gaussian distribution, i.e., $\boldsymbol{x}_i \sim \mathcal{N}(0, \boldsymbol{I}_m)$ for all $i \in [N]$. When $N \to \infty$ and for a fixed m, by the strong law of large numbers, the empirical covariance matrix $\hat{\boldsymbol{C}}_N$ converges almost surely to the covariance matrix $\boldsymbol{I}_M$, i.e.,

$$\hat{\boldsymbol{C}}_N \xrightarrow{a.s} \boldsymbol{I}_m.$$

However, when $N, m \to \infty$ with $m/N \to c < \infty$ for $c > 0$, we still have the entry-wise convergence by the strong law of large numbers, but the eigenvalue distributions of $\hat{\boldsymbol{C}}_N$ and $\boldsymbol{I}_m$ mismatch. When we consider the special case for which N and m are both large, but with $m > N$, the rank of $\hat{\boldsymbol{C}}_N$ is at most equal to N since $\hat{\boldsymbol{C}}_N$ is the sum of N rank one matrices. Because $\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T$ is a $m \times m$ matrix with $m > N$, the matrix $\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T$ is singular and has at least $m - N > 0$ zero eigenvalues. As a consequence, as $N, m \to \infty$ with $m/N \to c > 1$, we have

$$\hat{\boldsymbol{C}}_N \not\to \boldsymbol{I}_N.$$

The above claim also holds when $N, m \to \infty$ with $m/N \to c < 1$ as depicted by Figure 7.1. Even if $\hat{\boldsymbol{C}}_N$ is a poor estimate of $\boldsymbol{I}_m$ and $\hat{\boldsymbol{C}}_N$ does not converge in any useful way as $N, m \to \infty$, the

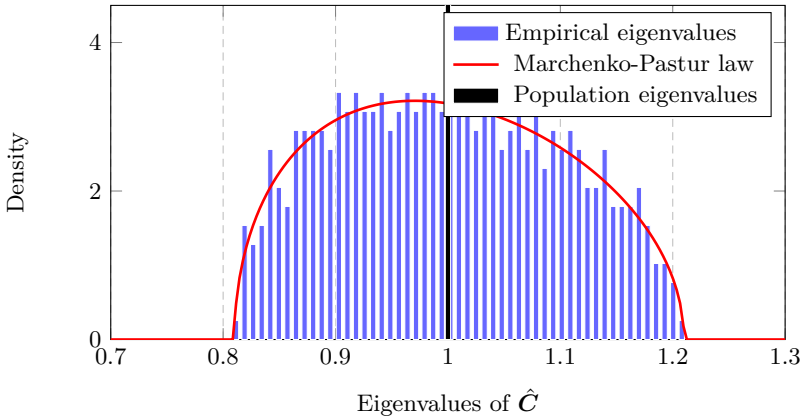


Figure 7.1: **Eigenvalue distributions of the empirical covariance matrix $\hat{\boldsymbol{C}}_N$ (equation 7.3) and the covariance matrix $\boldsymbol{I}_m$ mismatch for $N = 100m$. The eigenvalue distribution of the empirical covariance matrix $\hat{\boldsymbol{C}}_N$ converges to the Marchenko-Pastur distribution.** Eigenvalue histogram of $\hat{\boldsymbol{C}}_N$ versus the Marchenko-Pastur distribution for $m = 512$ and $N = 100m$ (Liao et al., 2020).

limiting eigenvalue distribution of $\hat{\boldsymbol{C}}_N$ as $N, m \to \infty$ is known in the Random Matrix Theory to be the popular *Marchenko-Pastur law* (Marchenko and Pastur, 1967) given by

$$\mu^{MP}(dx) = (1 - c^{-1}) \cdot \mathbf{1}_0(x) + \frac{1}{2\pi c x} \sqrt{\left(x - (1 - \sqrt{c})^2\right)^+ \left((1 + \sqrt{c})^2 - x\right)^+} dx, \tag{7.4}$$

with $\mathbf{1}_0(x)$ the Dirac mass at zero, $c = \lim m/N$ and $(x)^+ = \max(x, 0)$.

Remark 12. *The Marchenko-Pastur distribution reveals that the eigenvalues of $\hat{\boldsymbol{C}}_N$ do not concentrate around 1 like in the asymptotic N limit, but concentrate between $(1 - \sqrt{c})^2$ to $(1 + \sqrt{c})^2$. Therefore, the eigenvalues span on the range*

$$(1 - \sqrt{c})^2 + (1 + \sqrt{c})^2 = 4\sqrt{c} = 4\sqrt{\tfrac{m}{N}}.$$

In particular, for $N = 100m$, where one would expect a sufficiently large N for $\hat{\boldsymbol{C}}_N$ to estimate $\boldsymbol{I}_m$ properly, one has a spread around the true eigenvalue 1 equal to $4\sqrt{0.01} = 0.4$ as observed in Figure 7.1.

Remark 13. *Although introduced here in the context of Gaussian distributions for $\boldsymbol{x}_i \sim \mathcal{N}(0, \boldsymbol{I}_m)$. The Marcenko-Pastur law applies to much more general cases than in the context of Gaussian distributions. Indeed, the result remains valid when $\boldsymbol{x}_i$ has i.i.d. normalized entries of zero mean and unit variance.*

7.1.3 Empirical Eigenvalue Distribution & Resolvent

As in the example found in Section 7.1.2, the symmetric random matrix $\frac{1}{N}\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}$ does not converge in any useful way as $N, m \to \infty$. However, the empirical eigenvalue distribution of its limit can be studied in the Random Matrix Theory through its resolvent with the *Stieltjes transform*.

Definition 7.1.1 (Stieltjes Transform)**.** *For a real probability measure μ with support* supp(μ),

the Stieltjes transform $m_\mu : \mathbb{C} \setminus \operatorname{supp}(\mu) \to \mathbb{C}$ *is defined for all* $\lambda \in \mathbb{C} \setminus \operatorname{supp}(\mu)$ *as*

$$m_\mu(\lambda) = \int_{\mathbb{R}} \tfrac{1}{t-\lambda} d\mu(t).$$

Let $\boldsymbol{M} \in \mathbb{R}^{m \times m}$ be a symmetric matrix. Since $\boldsymbol{M}$ is symmetric, the matrix has real eigenvalues. We can define its *empirical eigenvalue measure* $\hat{\mu}_{\boldsymbol{M}}$ as

$$\hat{\mu}_{\boldsymbol{M}}(x) = \tfrac{1}{m} \sum_{i=1}^{m} \mathbf{1}_{\nu_i(\boldsymbol{M})}(x),$$

for all $x \in \mathbb{R}$. The empirical eigenvalue measure $\hat{\mu}_{\boldsymbol{M}}$ is a probability measure since $\hat{\mu}_{\boldsymbol{M}}(x) \geq 0$ for all $x \in \mathbb{R}$, and $\int_{\mathbb{R}} \hat{\mu}_{\boldsymbol{M}}(x)dx = 1$. For all $\lambda \in \mathbb{C} \setminus \operatorname{supp}(\hat{\mu}_{\boldsymbol{M}})$, the Stieltjes transform of $\hat{\mu}_{\boldsymbol{M}}$ can be expressed as the trace of the resolvent $[\boldsymbol{M} - \lambda \boldsymbol{I}_m]^{-1}$ since

$$\begin{aligned} m_{\hat{\mu}_{\boldsymbol{M}}}(\lambda) = \int_{\mathbb{R}} \tfrac{1}{t-\lambda} d\hat{\mu}_{\boldsymbol{M}}(t) &= \tfrac{1}{m} \sum_{i=1}^{m} \int_{\mathbb{R}} \tfrac{\mathbf{1}_{\nu_i(\boldsymbol{M})}(t)}{t-\lambda} dt \\ &= \tfrac{1}{m} \sum_{i=1}^{m} \tfrac{1}{\nu_i(\boldsymbol{M})-\lambda} \\ &= \tfrac{1}{m} \operatorname{Tr}\big([\boldsymbol{M} - \lambda \boldsymbol{I}_m]^{-1}\big). \end{aligned}$$

From Lemma 7.1.1, we deduce that the empirical eigenvalue distribution $\hat{\mu}_{\boldsymbol{M}}$ of the symmetric random matrix $\boldsymbol{M}$ can be studied with the resolvent $[\boldsymbol{M} - \lambda \boldsymbol{I}_m]^{-1}$ through the Stieltjes transform.

Lemma 7.1.1 (Inverse Stieltjes Transform (Couillet and Liao, 2022))**.** *For* a, b *continuity points of a probability measure* μ*, we have*

$$\mu\big([a,b]\big) = \tfrac{1}{\pi} \lim_{y \to 0} \int_a^b \Im\big(m_\mu(x+iy)\big) dx$$

If μ *admits a density* f *at* x*, i.e.,* $\mu(x)$ *is differentiable in a neighborhood of* x *and*

$$\lim_{\epsilon \to 0} \tfrac{1}{2\epsilon} \mu\big(x-\epsilon, x+\epsilon\big) = f(x),$$

then

$$f(x) = \tfrac{1}{\pi} \lim_{y \to 0} \Im\big(m_\mu(x+iy)\big).$$

Finally, if μ *has an isolated mass at* x*, then*

$$\mu\big(\{x\}\big) = -\tfrac{1}{\pi} \lim_{y \to 0} iy m_\mu(x+iy).$$

7.1.4 Deterministic Equivalent

Since the limit of high-dimensional symmetric random matrices does not converge in any useful sense as highlighted in Sections 7.1.1 and 7.1.2, a first line of works in the Random Matrix Theory has instead investigated the asymptotic characterization of their spectral measures using the Stieltjes transform and Lemma 7.1.1. In particular, the objective has been to determine the limit of the spectral measure $\hat{\mu}_{\boldsymbol{M}}$ of a symmetric random matrix $\boldsymbol{M} \in \mathbb{R}^{m \times m}$ when the dimension m

tends to infinity. For this purpose, by using Lemma 7.1.1, a natural approach is to study the associated random Stieltjes transform $m_{\hat{\mu}_{\boldsymbol{M}}}(\lambda) = \frac{1}{m}\operatorname{Tr}\big([\boldsymbol{M} - \lambda \boldsymbol{I}_m]^{-1}\big)$ of $\hat{\mu}_{\boldsymbol{M}}$ and to show that it admits a limit $m(\lambda)$ in probability or almost surely as $m \to \infty$.

Example 1. *In the example of the empirical covariance matrix presented in Section 7.1.2, Marchenko and Pastur (1967) demonstrated that as $N, m \to \infty$ with $m/N \to c$ with $c > 0$, we have*

$$m_{\hat{\mu}_{\hat{\boldsymbol{C}}_N}}(\lambda) = \frac{1}{m}\operatorname{Tr}\big(\big[\hat{\boldsymbol{C}}_N - \lambda \boldsymbol{I}_m\big]^{-1}\big) \xrightarrow{a.s} m(\lambda),$$

where $m_{\hat{\mu}_{\hat{\boldsymbol{C}}_N}}(\lambda)$ is the Stieltjes transform at λ of the empirical eigenvalue distribution $\hat{\mu}_{\hat{\boldsymbol{C}}_n}$ (equation 7.3) of $\hat{\boldsymbol{C}}_N$, and $m(\lambda)$ is defined as the unique positive solution of the Marchenko-Pastur equation

$$c\lambda m^2(\lambda) - (1 - \lambda - c)m(\lambda) + 1 = 0. \tag{7.5}$$

From the Stieltjes transform $m(\lambda)$, we can derive the Marchenko-Pastur law μ^{MP} given in equation 7.4.

Nevertheless, a study of the limit $m(\lambda)$ of the random Stieltjes transform $\lambda \mapsto m_{\hat{\mu}_{\boldsymbol{M}}}(\lambda) = \frac{1}{m}\operatorname{Tr}\big(\big[\boldsymbol{M} - \lambda \boldsymbol{I}_m\big]^{-1}\big)$ of $\hat{\mu}_{\boldsymbol{M}}$ assumes such a limit $m(\lambda)$ exists. Furthermore, it only quantifies the Stieltjes transform $\lambda \mapsto m_{\hat{\mu}_{\boldsymbol{M}}}(\lambda) = \frac{1}{m}\operatorname{Tr}\big(\big[\boldsymbol{M} - \lambda \boldsymbol{I}_m\big]^{-1}\big)$ and does not take into account other subspace information of the symmetric random matrix $\boldsymbol{M}$ carried in the resolvent matrix $\big[\boldsymbol{M} - \lambda \boldsymbol{I}_m\big]^{-1}$, e.g, the eigenvector space of $\boldsymbol{M}$.

To overcome those limitations, we focus instead on finding a *"deterministic equivalent"* of the resolvent of the symmetric random matrix $\boldsymbol{M}$, which is a non-asymptotic deterministic matrix having in probability or almost surely asymptotically the same scalar observations as the random one (Hachem et al., 2007; Couillet and Debbah, 2011; Couillet and Liao, 2022). The notion of "deterministic equivalent" has not been formally defined in the literature. However, in the following, we propose a definition we will adopt in this part.

Definition 7.1.2 (Deterministic Equivalent)**.** *Let $\boldsymbol{M} \in \mathbb{R}^{N \times m}$ be a random matrix and $f : \mathbb{R}^{N \times m} \to \mathbb{R}$ be a mapping from matrices of size $N \times m$ to scalars. A matrix $\overline{\boldsymbol{M}} \in \mathbb{R}^{N \times m}$ is said to be a deterministic equivalent of $\boldsymbol{M}$ if, as $N, m \to \infty$, we have*

$$f(\boldsymbol{M}) - f(\overline{\boldsymbol{M}}) \to 0,$$

where the convergence is either in probability or almost surely.

Remark 14. *This definition extends the definition proposed by Couillet and Liao (2022) to include non-symmetric and rectangular random matrices and any mapping $f : \mathbb{R}^{N \times m} \to \mathbb{R}$. Indeed, Couillet and Liao (2022) assume in their definition that $\boldsymbol{M}$ is symmetric for the mappings $f : \boldsymbol{M} \mapsto \frac{1}{m}\operatorname{Tr}(\boldsymbol{AM})$ and $f : \boldsymbol{M} \mapsto \boldsymbol{a}^T\boldsymbol{Mb}$, for any deterministic matrix $\boldsymbol{A} \in \mathbb{R}^{m \times m}$ and vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^m$ of unit norms, with respect to the operator and the Euclidean norm.*

Remark 15. *Let $\boldsymbol{M} \in \mathbb{R}^{m \times m}$ be a symmetric random matrix. If $\overline{\boldsymbol{Q}} \in \mathbb{R}^{m \times m}$ is a deterministic equivalent of the resolvent $\big[\boldsymbol{M} - \lambda \boldsymbol{I}_m\big]^{-1}$ for the application $f : \boldsymbol{M} \mapsto \frac{1}{m}\operatorname{Tr}\big(\boldsymbol{M}\big)$, then the Stieljes transform $m_{\hat{\mu}_{\boldsymbol{M}}}$ of the empirical eigenvalue distribution $\hat{\mu}_{\boldsymbol{M}}$ of $\boldsymbol{M}$ converges towards $\frac{1}{m}\operatorname{Tr}\big(\bar{\boldsymbol{Q}}\big)$. In the empirical covariance matrix example presented in Section 7.1.2, the matrix $m(\lambda)\boldsymbol{I}_m$ can be thus interpreted as a deterministic equivalent of the resolvent $\big[\hat{\boldsymbol{C}}_N - \lambda \boldsymbol{I}_m\big]^{-1}$ for the application $f : \boldsymbol{M} \mapsto \frac{1}{m}\operatorname{Tr}\big(\boldsymbol{M}\big)$, where $m(\lambda)$ is the solution of the Marchenko-Pastur equation (equation 7.5).*

Finding deterministic equivalents of resolvents of random matrices is particularly useful in theoretical works on supervised learning (Louart et al., 2018; Liao et al., 2020; Jacot et al., 2020a; Mei and Montanari, 2022; Couillet and Liao, 2022). In particular, the aforementioned works investigate the performance of ridge regression with random features, where the solution to the ridge regression depends on the resolvent of a symmetric semi-positive-definite random matrix. These studies assess the asymptotic training error and the test error by identifying a deterministic equivalent of the resolvent in a double asymptotic regime; where the number of samples $n \to \infty$ and the number of features $N \to \infty$, while maintaining a constant ratio N/n. Their proofs mainly rely on the symmetric property of the resolvent and on Random Matrix tools designed for random symmetric matrices, e.g, the Stieltjes transform. In the following sections of this part, we aim to extend these results to the regularized LSTD algorithm and to certain resolvents of non-symmetric random matrices with specific structures.

7.2 A Deterministic Equivalent Resolvent for Regularized LSTD

From equation 6.12, the weight vector $\hat{\boldsymbol{\theta}}_n^\lambda$ returned by regularized LSTD depends on the random resolvent $\boldsymbol{Q}_m(\lambda)$ defined in equation 6.13 since

$$\hat{\boldsymbol{\theta}}_n^\lambda = \tfrac{1}{\sqrt{n}} \tfrac{1}{m} \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m(\lambda) \boldsymbol{r}.$$

The random resolvent $\boldsymbol{Q}_m(\lambda)$ thus plays a significant role in the performance of regularized LSTD and in error functions studied in this chapter; such as the empirical Mean-Squared Bellman Error (in Section 7.3), the Mean-Squared Bellman Error (in Section 7.4), and the Mean-Squared Value Error (in Section 7.5). In order to assess the asymptotic performance of regularized LSTD in the double asymptotic regime of Assumption 2, we need to find a *deterministic equivalent* for the random matrix $\boldsymbol{Q}_m(\lambda)$ and for other random matrices that are functions of random features. In our theoretical analysis, the identification of deterministic equivalents relies on the following concentration measure for Lipschitz applications of a Gaussian vector.

Lemma 7.2.1 (Normal Concentration)**.** *((Ledoux, 2001, Corollary 2.6, Propositions 1.3, 1.8) or (Tao, 2012, Theorem 2.1.12)) For $d \in \mathbb{N}$, consider μ the canonical Gaussian probability on $\mathbb{R}^d$ defined through its density $d\mu(\boldsymbol{w}) = (2\pi)^{-\frac{d}{2}} e^{-\frac{1}{2}\|\boldsymbol{w}\|^2}$ and $f : \mathbb{R}^d \to \mathbb{R}$ a L_f-Lipschitz function. Then*

$$\mu\left(\left\{\left|f - \int f d\mu\right| \geq t\right\}\right) \leq C e^{-c\frac{t^2}{L_f^2}}, \tag{7.6}$$

where $C, c > 0$ are independent of d and L_f.

This concentration result is particularly interesting in our analysis since it can be extended to Lipschitz functions of sub-Gaussian matrices as stated by the following Lemma.

Lemma 7.2.2. *Let $f : \mathbb{R}^{N\times d} \to \mathbb{R}$, $\boldsymbol{W} \mapsto f(\boldsymbol{W})$ be a K_f-Lipschitz function with respect to the Frobenius norm, for which $\boldsymbol{W} = \varphi(\tilde{\boldsymbol{W}})$ is the matrix defined in equation 6.5. Then, we have*

$$\Pr\left(\left|f(\boldsymbol{W}) - \mathbb{E}_{\boldsymbol{W}}\left[f(\boldsymbol{W})\right]\right| > t\right) \leq C e^{-\frac{ct^2}{K_f^2 K_\varphi^2}},$$

for some $C, c > 0$.

Proof. The vectorization of $\tilde{\boldsymbol{W}}$, $\text{vec}(\tilde{\boldsymbol{W}}) = \left[\tilde{\boldsymbol{W}}_{11}, \cdots, \tilde{\boldsymbol{W}}_{nd}\right] \in \mathbb{R}^{N\times d}$ is a Gaussian vector. A K_f-Lipschitz function f of $\boldsymbol{W}$ with respect to the Frobenius norm is also a K_f-Lipschitz function of $\text{vec}(\boldsymbol{W})$ with respect to the Euclidean norm. Applying Lemma 7.2.1 gives

$$\Pr\left(\left|f(\boldsymbol{W}) - \mathbb{E}_{\boldsymbol{W}}\left[f(\boldsymbol{W})\right]\right| > t\right) = \Pr\left(\left|f\big(\varphi(\tilde{\boldsymbol{W}})\big) - \mathbb{E}_{\boldsymbol{W}}\left[f\big(\varphi(\tilde{\boldsymbol{W}})\big)\right]\right| > t\right) \leq Ce^{-\frac{ct^2}{K_\varphi^2 K_f^2}},$$

for some $C, c > 0$. □

In the following sections of this chapter, by leveraging Lemma 7.2.2 and the Lipschitz continuity of the error functions with respect to the random weight matrix $\boldsymbol{W}$ under the frobenius norm, we show that the error functions asymptotically concentrate around their deterministic expected values in the double asymptotic regime of Assumption 2. A natural deterministic equivalent for the random resolvent $\boldsymbol{Q}_m(\lambda)$ would be thus $\mathbb{E}_{\boldsymbol{W}}[\boldsymbol{Q}_m(\lambda)]$. However, $\mathbb{E}_{\boldsymbol{W}}[\boldsymbol{Q}_m(\lambda)]$ involves integration without having a closed form expression due to the matrix inverse, and is inconvenient for practical computation. Leveraging the Random Matrix Theory, the following Theorem 7.2.3 proposes an asymptotic form that is *i.* close to $\mathbb{E}_{\boldsymbol{W}}[\boldsymbol{Q}_m(\lambda)]$ under Assumptions 1 and 2, and *ii.* numerically more accessible.

Theorem 7.2.3 (Asymptotic Deterministic Resolvent)**.** *Under Assumptions 1 (double asymptotic regime) and 2 (bounded spectrum), let* $\lambda > 0$ *and let* $\bar{\boldsymbol{Q}}_m(\lambda) \in \mathbb{R}^{n\times n}$ *be the deterministic resolvent defined as*

$$\bar{\boldsymbol{Q}}_m(\lambda) = \left[\tfrac{N}{m}\tfrac{1}{1+\delta}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n + \lambda\boldsymbol{I}_n\right]^{-1}, \tag{7.7}$$

where the deterministic Gram feature matrix $\boldsymbol{\Phi}_{\hat{\mathcal{S}}} \in \mathbb{R}^{m\times m}$ *is defined as*

$$\boldsymbol{\Phi}_{\hat{\mathcal{S}}} = \mathbb{E}_{\boldsymbol{w}}\left[\sigma\big(\boldsymbol{w}^T\hat{\boldsymbol{S}}\big)^T\sigma\big(\boldsymbol{w}^T\hat{\boldsymbol{S}}\big)\right] = \mathbb{E}_{\tilde{\boldsymbol{w}}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}\left[\sigma\big(\varphi(\tilde{\boldsymbol{w}})^T\hat{\boldsymbol{S}}\big)^T\sigma\big(\varphi(\tilde{\boldsymbol{w}})^T\hat{\boldsymbol{S}}\big)\right]$$

and the correction factor δ *is the unique, positive, solution to*

$$\delta = \tfrac{1}{m}\,\text{Tr}\left((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\left[\tfrac{N}{m}\tfrac{1}{1+\delta}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n + \lambda\boldsymbol{I}_n\right]^{-1}\right). \tag{7.8}$$

Then

$$\lim_{m\to\infty}\left\|\mathbb{E}_{\boldsymbol{W}}\left[\boldsymbol{Q}_m(\lambda)\right] - \bar{\boldsymbol{Q}}_m(\lambda)\right\| = 0.$$

Proof. Details of the proof can be found in Appendix A.1. □

Remark 16. *Since* $\delta \to 0$ *when* $N/m \to \infty$*, the correction factor* $\frac{1}{1+\delta}$ *arises from the double asymptotic regime, which keeps the ratio* N/m *asymptotically constant. Similar correction factors arise in related Random Matrix literature, which, however, deals with semi-positive-definite matrices (Couillet and Debbah, 2011; Liu et al., 2019a; Liao et al., 2020; Jacot et al., 2020a;b). Our problem exceeds this frame, so we prove the result, including existence and uniqueness, with a somewhat more involved analysis based on the eigenspectrum of the products of matrices with semi-positive-definite symmetric part and skew-symmetric matrices (see Appendix A.6).*

Table 7.1: Values of $\boldsymbol{\Phi}_{\boldsymbol{ab}}$ for $\boldsymbol{w} \sim \mathcal{N}(0, \boldsymbol{I}_d)$, $\angle(\boldsymbol{a}, \boldsymbol{b}) \equiv \frac{\boldsymbol{a}^T\boldsymbol{b}}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|}$ (Louart et al., 2018).

$\sigma(t)$	$\boldsymbol{\Phi}_{\boldsymbol{ab}}$
t	$\boldsymbol{a}^T\boldsymbol{b}$
$\max(t, 0)$	$\frac{1}{2\pi}\|\boldsymbol{a}\|\|\boldsymbol{b}\| \left(\angle(\boldsymbol{a}, \boldsymbol{b}) \arccos(-\angle(\boldsymbol{a}, \boldsymbol{b})) + \sqrt{1 - \angle(\boldsymbol{a}, \boldsymbol{b})^2}\right)$
$\|t\|$	$\frac{2}{\pi}\|\boldsymbol{a}\|\|\boldsymbol{b}\| \left(\angle(\boldsymbol{a}, \boldsymbol{b}) \arcsin(\angle(\boldsymbol{a}, \boldsymbol{b})) + \sqrt{1 - \angle(\boldsymbol{a}, \boldsymbol{b})^2}\right)$
$\mathrm{erf}(t)$	$\frac{2}{\pi} \arcsin\left(\frac{2\boldsymbol{a}^T\boldsymbol{b}}{\sqrt{(1+2\|\boldsymbol{a}\|^2)(1+2\|\boldsymbol{b}\|^2)}}\right)$
$1_{\{t>0\}}$	$\frac{1}{2} - \frac{1}{2\pi} \arccos(\angle(\boldsymbol{a}, \boldsymbol{b}))$
$\mathrm{sign}(t)$	$\frac{2}{\pi} \arcsin(\angle(\boldsymbol{a}, \boldsymbol{b}))$
$\cos(t)$	$\exp(-\frac{1}{2}(\|\boldsymbol{a}\|^2 + \|\boldsymbol{b}\|^2)) \cosh(\boldsymbol{a}^T\boldsymbol{b})$
$\sin(t)$	$\exp(-\frac{1}{2}(\|\boldsymbol{a}\|^2 + \|\boldsymbol{b}\|^2)) \sinh(\boldsymbol{a}^T\boldsymbol{b})$.

Remark 17. *It can be shown that δ is a decreasing function with respect to the number of parameters N (Lemma A.6.3) and with respect to the l_2-regularization parameter λ (Lemma A.6.4).*

Remark 18. *In supervised learning, a comparable proposition is presented by Louart et al. (2018, Theorem 1) for the resolvent of random symmetric semi-positive-definite matrices. It constitutes a special case of Theorem 7.2.3 with $\gamma = 0$, which corresponds to the case where we learn the reward function $R^\pi : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ using samples of $\mathcal{D}_{train}$. Our analysis extends this result to the resolvent $\boldsymbol{Q}_m(\lambda)$, which is the resolvent of a non-symmetric random matrix. The loss of the symmetric property significantly complicates our analysis, e.g., with the proof of the existence of the correction factor δ or the uniform boundess of $\boldsymbol{Q}_m(\lambda)$ with respect to the operator norm.*

Remark 19. *Note that the matrix $\boldsymbol{Q}_m(\lambda)$ is the resolvent of the random matrix $\frac{1}{m}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n$, which is non-symmetric when $\gamma > 0$. Therefore, many tools from the related Random Matrix literature used to study the spectrum of $\frac{1}{m}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n$ with $\boldsymbol{Q}_m(\lambda)$ are not applicable, e.g., the Stieljes transform. Indeed, with complex eigenvalues, the empirical eigenvalue measure of $\frac{1}{m}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n$ is no longer a real probability measure, does not admit a Stieltjes transform, and $\frac{1}{m}\mathrm{Tr}(\boldsymbol{Q}_m(\lambda))$ can no longer be interpreted as the Stieltjes transform of this empirical eigenvalue measure as it was the case in Section 7.1.3. As a consequence, we can not use the deterministic equivalent $\bar{\boldsymbol{Q}}_m(\lambda)$ of the resolvent $\boldsymbol{Q}_m(\lambda)$ for the application $f : \boldsymbol{M} \mapsto \frac{1}{m}\mathrm{Tr}(\boldsymbol{M})$ to provide spectral information about $\frac{1}{m}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n$.*

Remark 20. *The evaluation of $\boldsymbol{\Phi}_{\hat{\mathcal{S}}} = \mathbb{E}_{\boldsymbol{w}}\left[\sigma(\boldsymbol{w}^T\hat{\boldsymbol{S}})^T\sigma(\boldsymbol{w}^T\hat{\boldsymbol{S}})\right]$ is obtained through the evaluation of its individual entries and thus to the calculus, for arbitrary vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d$ of*

$$\boldsymbol{\Phi}_{ab} = \mathbb{E}_{\boldsymbol{w}}\left[\sigma(\boldsymbol{w}^T\boldsymbol{a})\sigma(\boldsymbol{w}^T\boldsymbol{b})\right] = (2\pi)^{-\frac{p}{2}} \int \sigma(\varphi(\tilde{\boldsymbol{w}})^T\boldsymbol{a})\sigma(\varphi(\tilde{\boldsymbol{w}})^T\boldsymbol{b})e^{-\frac{1}{2}\|\tilde{\boldsymbol{w}}\|^2} d\tilde{\boldsymbol{w}}. \tag{7.9}$$

The evaluation of equation 7.9 can be obtained through various integration tricks for a wide family of mappings $\varphi(\cdot)$ and activation functions $\sigma(\cdot)$. We provide in Table 7.1 (found in Louart et al. (2018)) the values of $\boldsymbol{\Phi}_{\boldsymbol{ab}}$ when $\boldsymbol{w} \sim \mathcal{N}(0, \boldsymbol{I}_d)$ (i.e., for $\varphi(t) = t$) and for a set of activation functions $\sigma : \mathbb{R} \to \mathbb{R}$ not necessarily satisfying the Lipschitz continuity. In experiments in Section 9, we focus only on the ReLU function, i.e., $\sigma(t) = \max(t, 0)$.

7.3 Asymptotic Empirical Mean-Squared Bellman Error

Value-based algorithms that use bootstrapping, especially TD learning methods, aim to minimize the empirical mean squared Bellman error (MSBE), denoted by $\widehat{\text{MSBE}}$. As mentioned in Section 4.3, under appropriate learning rates (Robbins and Monro, 1951), linear TD learning methods using a gradient-based approach converge towards the solution $\boldsymbol{\theta}_n^\lambda = \frac{1}{\sqrt{n}}\frac{1}{m}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m(\lambda)\boldsymbol{r}$ of the regularized LSTD. It is straightforward to show that $\widehat{\text{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda)$ depends on a quadratic form of the random resolvent $\boldsymbol{Q}_m(\lambda)$ as

$$\begin{aligned}
\widehat{\text{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda) &= \tfrac{1}{n}\|\boldsymbol{r} + \gamma\boldsymbol{\Sigma}_{\boldsymbol{X}_n'}^T\hat{\boldsymbol{\theta}}_n^\lambda - \boldsymbol{\Sigma}_{\boldsymbol{X}_n}^T\hat{\boldsymbol{\theta}}_n^\lambda\|^2 \\
&= \tfrac{1}{n}\left\|\boldsymbol{r} - \tfrac{1}{mn}\left(\boldsymbol{\Sigma}_{\boldsymbol{X}_n} - \gamma\boldsymbol{\Sigma}_{\boldsymbol{X}_n'}\right)^T\boldsymbol{\Sigma}_{\boldsymbol{X}_n}\boldsymbol{Q}_m(\lambda)\boldsymbol{r}\right\|^2 \\
&= \tfrac{1}{n}\left\|\left[\tfrac{1}{mn}\left(\boldsymbol{\Sigma}_{\boldsymbol{X}_n} - \gamma\boldsymbol{\Sigma}_{\boldsymbol{X}_n'}\right)^T\boldsymbol{\Sigma}_{\boldsymbol{X}_n} + \lambda\boldsymbol{I}_n - \tfrac{1}{mn}\left(\boldsymbol{\Sigma}_{\boldsymbol{X}_n} - \gamma\boldsymbol{\Sigma}_{\boldsymbol{X}_n'}\right)^T\boldsymbol{\Sigma}_{\boldsymbol{X}_n}\right]\boldsymbol{Q}_m(\lambda)\boldsymbol{r}\right\|^2 \\
&= \tfrac{\lambda^2}{n}\boldsymbol{r}^T\boldsymbol{Q}_m(\lambda)^T\boldsymbol{Q}_m(\lambda)\boldsymbol{r}.
\end{aligned}$$

Using Lemma 7.2.2, we show in the following Lemma that $\widehat{\text{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda)$ concentrates around $\frac{\lambda^2}{n}\boldsymbol{r}^T\mathbb{E}_{\boldsymbol{W}}\left[\boldsymbol{Q}_m(\lambda)^T\boldsymbol{Q}_m(\lambda)\right]\boldsymbol{r}$ under the double asymptotic regime of Assumption 2.

Lemma 7.3.1. *Under Assumptions 1 and 2, we have*

$$\Pr\left(\left|\widehat{\text{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda) - \tfrac{\lambda^2}{n}\boldsymbol{r}^T\mathbb{E}_{\boldsymbol{W}}[\boldsymbol{Q}_m(\lambda)^T\boldsymbol{Q}_m(\lambda)]\boldsymbol{r}\right| > t\right) \leq Ce^{-cmt^2},$$

for some $C, c > 0$ independent of m and N.

Proof. Let the mapping $f : \boldsymbol{R} \mapsto \frac{\lambda^2}{n}\boldsymbol{r}^T\boldsymbol{Q}_m(\boldsymbol{R})^T\boldsymbol{Q}_m(\boldsymbol{R})\boldsymbol{r}$, for the resolvent mapping

$$\boldsymbol{Q}_m : \boldsymbol{R} \mapsto \left[\frac{1}{m}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\sigma(\boldsymbol{R}\hat{\boldsymbol{S}})^T\sigma(\boldsymbol{R}\hat{\boldsymbol{S}})\hat{\boldsymbol{U}}_n + \lambda\boldsymbol{I}_n\right]^{-1}$$

defined as in equation 6.13. We want to show f is Lipschitz in order to apply Lemma 7.2.2. From Lemma A.4.6, we know there exists a real $K > 0$ independent of N and m such that, for all $\boldsymbol{R}, \boldsymbol{H}$, we have

$$\|\boldsymbol{Q}_m(\boldsymbol{R}+\boldsymbol{H})^T\boldsymbol{Q}_m(\boldsymbol{R}+\boldsymbol{H}) - \boldsymbol{Q}_m(\boldsymbol{R})^T\boldsymbol{Q}_m(\boldsymbol{R})\| \leq \tfrac{K}{\sqrt{m}}\|\boldsymbol{H}\|_F.$$

Let $\boldsymbol{H} \in \mathbb{R}^{N\times d}$, we have thus

$$\begin{aligned}
|f(\boldsymbol{R}+\boldsymbol{H}) - f(\boldsymbol{R})| &= \left|\tfrac{\lambda^2}{n}\boldsymbol{r}^T\left[\boldsymbol{Q}_m(\boldsymbol{R}+\boldsymbol{H})^T\boldsymbol{Q}_m(\boldsymbol{R}+\boldsymbol{H}) - \boldsymbol{Q}_m(\boldsymbol{R})^T\boldsymbol{Q}_m(\boldsymbol{R})\right]\boldsymbol{r}\right| \\
&\leq \tfrac{\lambda^2}{n}\|\boldsymbol{r}\|^2\|\boldsymbol{Q}_m(\boldsymbol{R}+\boldsymbol{H})^T\boldsymbol{Q}_m(\boldsymbol{R}+\boldsymbol{H}) - \boldsymbol{Q}_m(\boldsymbol{R})^T\boldsymbol{Q}_m(\boldsymbol{R})\| \\
&\leq \tfrac{\lambda^2 K_r^2 K}{\sqrt{m}}\|\boldsymbol{H}\|_F \qquad \left(\tfrac{1}{n}\|\boldsymbol{r}\|^2 \leq \tfrac{1}{n}n\|\boldsymbol{r}\|_\infty^2 = K_r^2\right).
\end{aligned}$$

We deduce f is $\frac{\lambda^2 K_r^2 K}{\sqrt{m}}$-Lipschitz under the operator norm. From Lemma 7.2.1, we have for the random matrix $\boldsymbol{W}$ defined in equation 6.5

$$\begin{aligned}
&\Pr\left(\left|f(\boldsymbol{W}) - \mathbb{E}_{\boldsymbol{W}}\left[f(\boldsymbol{W})\right]\right| > t\right) \\
&= \Pr\left(\left|\tfrac{\lambda^2}{n}\boldsymbol{r}^T\boldsymbol{Q}_m(\lambda)^T\boldsymbol{Q}_m(\lambda)\boldsymbol{r} - \tfrac{\lambda^2}{n}\boldsymbol{r}^T\mathbb{E}_{\boldsymbol{W}}[\boldsymbol{Q}_m(\lambda)^T\boldsymbol{Q}_m(\lambda)]\boldsymbol{r}\right| > t\right) \\
&= \Pr\left(\left|\widehat{\text{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda) - \tfrac{\lambda^2}{n}\boldsymbol{r}^T\mathbb{E}_{\boldsymbol{W}}[\boldsymbol{Q}_m(\lambda)^T\boldsymbol{Q}_m(\lambda)]\boldsymbol{r}\right| > t\right)
\end{aligned}$$

$$\leq Ce^{-\frac{cmt^2}{K^2K_{\boldsymbol{r}}^4\lambda^4}},$$

for some $C, c > 0$ independent of other parameters. □

Remark 21. *The Lipschitz nature of the mapping $f : \boldsymbol{R} \mapsto \frac{\lambda^2}{n}\boldsymbol{r}^T\boldsymbol{Q}_m(\boldsymbol{R})^T\boldsymbol{Q}_m(\boldsymbol{R})\boldsymbol{r}$ depends on Lemma A.4.6, which is guaranteed by the uniform boundedness of the resolvent $\boldsymbol{Q}_m(\lambda)$ for any random matrix $\boldsymbol{W}$. The uniform boundedness of $\boldsymbol{Q}_m(\lambda)$ in the double asymptotic regime of Assumption 2 is non-trivial since the dimension n of $\boldsymbol{Q}_m(\lambda) \in \mathbb{R}^{n\times n}$ tends toward infinity ($n > m \to \infty$). The result plays a key role in our proofs and is proven in Appendix A.4.*

From Lemma 7.3.1, to assess the asymptotic $\widehat{\mathrm{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda)$ in the double asymptotic regime, we need to find a deterministic equivalent of $\boldsymbol{Q}_m(\lambda)^T\boldsymbol{Q}_m(\lambda)$ close to $\mathbb{E}\big[\boldsymbol{Q}_m(\lambda)^T\boldsymbol{Q}_m(\lambda)\big]$. Using the deterministic equivalent $\bar{\boldsymbol{Q}}_m(\lambda)$ of $\boldsymbol{Q}_m(\lambda)$ (found in Theorem 7.2.3), we identify an asymptotic form close to $\mathbb{E}\big[\boldsymbol{Q}_m(\lambda)^T\boldsymbol{Q}_m(\lambda)\big]$ in Lemma A.2.2 to derive the following theorem.

Theorem 7.3.2 (Asymptotic Empirical MSBE)**.** *Under the conditions of Theorem 7.2.3, the deterministic asymptotic empirical MSBE is*

$$\overline{\widehat{\mathrm{MSBE}}}(\hat{\boldsymbol{\theta}}_n^\lambda) = \tfrac{\lambda^2}{n}\|\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{r}\|^2 + \hat{\Delta}, \tag{7.10}$$

with second-order correction factor

$$\hat{\Delta} = \tfrac{\lambda^2}{n}\frac{\frac{1}{N}\operatorname{Tr}\big(\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m(\lambda)^T\big)}{1-\frac{1}{N}\operatorname{Tr}\big(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m(\lambda)^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m(\lambda)\big)}\|\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2, \tag{7.11}$$

where

$$\boldsymbol{\Psi}_1 = \tfrac{N}{m}\tfrac{1}{1+\delta}\hat{\boldsymbol{U}}_n^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n, \quad \textit{and} \quad \boldsymbol{\Psi}_2 = \tfrac{N}{m}\tfrac{1}{1+\delta}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n). \tag{7.12}$$

As $N, m \to \infty$ with asymptotic constant ratio N/m,

$$\widehat{\mathrm{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda) - \overline{\widehat{\mathrm{MSBE}}}(\hat{\boldsymbol{\theta}}_n^\lambda) \xrightarrow{a.s} 0.$$

Proof. Details of the proof can be found in Appendix A.2. □

Remark 22. *Similarly to the first order-correction factor δ (equation 7.8), the second order-correction factor $\hat{\Delta}$ arises from the double asymptotic regime since $\hat{\Delta} \to 0$ as $N/m \to \infty$.*

Remark 23. *A comparable proposition is presented by Louart et al. (2018, Theorem 3) for the training error in supervised learning. It is a special case of Theorem 7.3.2 with $\gamma = 0$, where we learn the reward function $R^\pi : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$.*

An interpretation of terms in $\overline{\widehat{\mathrm{MSBE}}}(\hat{\boldsymbol{\theta}}_n^\lambda)$ is provided in Chapter 8.

7.4 Asymptotic Mean-Squared Bellman Error

While the empirical MSBE only takes transitions from the dataset $\mathcal{D}_{\text{train}}$ into account, the true or population MSBE (equation 6.2) involves all states in $\mathcal{S}$. To extend the convergence results from the previous section to this case, we require some further notations. Using a decomposition similar

to equation 6.7, we express $\boldsymbol{\Sigma}_{\boldsymbol{X}_n}$ and $\boldsymbol{\Sigma}_{\boldsymbol{X}'_n}$ as a product of the random feature matrix of the entire state space $\boldsymbol{\Sigma}_{\mathcal{S}} \in \mathbb{R}^{N \times |\mathcal{S}|}$ with $\boldsymbol{U}_n \in \mathbb{R}^{|\mathcal{S}| \times n}$ and $\boldsymbol{V}_n \in \mathbb{R}^{|\mathcal{S}| \times n}$ as

$$\boldsymbol{\Sigma}_{\boldsymbol{X}_n} = \sqrt{n}\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{U}_n \quad \text{and} \quad \boldsymbol{\Sigma}_{\boldsymbol{X}'_n} = \sqrt{n}\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{V}_n.$$

Each column i of $\sqrt{n}\boldsymbol{U}_n$ is a one-hot vector, where the j-th element equals 1 if the i-th state $\boldsymbol{s}_i$ of $\boldsymbol{X}_n$ is $\boldsymbol{S}_j$, and similarly for $\sqrt{n}\boldsymbol{V}_n$ and $\boldsymbol{X}'_n$. We obtain a decomposition of the transition model matrix

$$\boldsymbol{A}_n = \boldsymbol{U}_n(\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T.$$

Using those notations, we can decompose $\mathrm{MSBE}(\hat{\boldsymbol{\theta}}_n^\lambda)$ as

$$\begin{aligned}
\mathrm{MSBE}(\hat{\boldsymbol{\theta}}_n^\lambda) &= \|\bar{\boldsymbol{r}} + \gamma \boldsymbol{P}^\pi \boldsymbol{\Sigma}_{\mathcal{S}}^T \hat{\boldsymbol{\theta}}_n^\lambda - \boldsymbol{\Sigma}_{\mathcal{S}}^T \hat{\boldsymbol{\theta}}_n^\lambda\|^2_{\boldsymbol{D}_{\boldsymbol{\mu}^\pi}} \\
&= \|\bar{\boldsymbol{r}} + \left[\gamma \boldsymbol{P}^\pi - \boldsymbol{I}_{|\mathcal{S}|}\right]\boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\theta}_n^\lambda\|^2_{\boldsymbol{D}_{\boldsymbol{\mu}^\pi}} \\
&= \left\|\bar{\boldsymbol{r}} - \tfrac{1}{m\sqrt{n}}\left[\boldsymbol{I}_{|\mathcal{S}|} - \gamma \boldsymbol{P}^\pi\right]\boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\Sigma}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_m(\lambda)\boldsymbol{r}\right\|^2_{\boldsymbol{D}_{\boldsymbol{\mu}^\pi}} \\
&= \|\bar{\boldsymbol{r}}\|^2_{\boldsymbol{D}_{\boldsymbol{\mu}^\pi}} \\
&\quad \underbrace{- \tfrac{2}{m\sqrt{n}} \bar{\boldsymbol{r}}^T \boldsymbol{D}_{\boldsymbol{\mu}^\pi}\left[\boldsymbol{I}_{|\mathcal{S}|} - \gamma \boldsymbol{P}^\pi\right]\boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\Sigma}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_m(\lambda)\boldsymbol{r}}_{=Z_2} \\
&\quad \underbrace{+ \left\|\tfrac{1}{m\sqrt{n}}\left[\boldsymbol{I}_{|\mathcal{S}|} - \gamma \boldsymbol{P}^\pi\right]\boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\Sigma}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_m(\lambda)\boldsymbol{r}\right\|^2_{\boldsymbol{D}_{\boldsymbol{\mu}^\pi}}}_{=Z_3}.
\end{aligned}$$

Similar to the empirical MSBE, we want to show that $\mathrm{MSBE}(\hat{\boldsymbol{\theta}}_n^\lambda)$ concentrates around $\mathbb{E}_{\boldsymbol{W}}\left[\mathrm{MSBE}(\hat{\boldsymbol{\theta}}_n^\lambda)\right]$. Specifically, we want to show that both Z_2 and Z_3 concentrate around $\mathbb{E}_{\boldsymbol{W}}[Z_2]$ and $\mathbb{E}_{\boldsymbol{W}}[Z_3]$, respectively. Those concentration results can be derived with Lemma 7.2.2, by showing that $\frac{1}{m}\left[\boldsymbol{I}_{|\mathcal{S}|} - \gamma \boldsymbol{P}^\pi\right]\boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\Sigma}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_m(\lambda)$ is uniformly bounded under the operator norm for any random matrix $\boldsymbol{W}$ in the double asymptotic regime of Assumption 2. Since the empirical transition model matrix $\hat{\boldsymbol{A}}_m = \hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T$ (equation 6.14) is invertible under Assumption 2, we can easily bound the operator norm of $\frac{1}{m}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m(\lambda)$ as

$$\begin{aligned}
\left\|\tfrac{1}{m}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m(\lambda)\right\| &= \left\|\tfrac{1}{m}\hat{\boldsymbol{A}}_m^{-1}\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m(\lambda)\right\| \\
&= \left\|\hat{\boldsymbol{A}}_m^{-1}\hat{\boldsymbol{U}}_n\left[\boldsymbol{I}_n - \lambda \boldsymbol{Q}_m(\lambda)\right]\right\| \\
&\leq \tfrac{1}{\xi_{\min}}\big(1 + \lambda K_{\boldsymbol{Q}}\big),
\end{aligned}$$

where $K_{\boldsymbol{Q}}$ is the uniform bound of $\boldsymbol{Q}_m(\lambda)$ under the operator norm in the double asymptotic regime (a proof can be found in Appendix A.4). This results unfolds because $\|\hat{\boldsymbol{A}}_m^{-1}\| = \frac{1}{\nu_{\min}(\hat{\boldsymbol{A}}_m^T \hat{\boldsymbol{A}}_m)} \leq \frac{1}{\nu_{\min}(H(\hat{\boldsymbol{A}}_m))} \leq \frac{1}{\xi_{\min}}$ and $\|\hat{\boldsymbol{U}}_n\| \leq 1$. Unfortunately, we do not have such straightforward control on the operator norm of $\frac{1}{m}\boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\Sigma}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_m$ since $\boldsymbol{A}_n = \boldsymbol{U}_n(\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T$ is not invertible until all states are visited. Furthermore, only a $\mathcal{O}(\sqrt{m})$ upper bound can be derived for $\left\|\frac{1}{\sqrt{m}}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\right\|$ from Corollary A.7.1.1. A solution to upper bound the operator norm of $\frac{1}{m}\left[\boldsymbol{I}_{|\mathcal{S}|} - \gamma \boldsymbol{P}^\pi\right]\boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\Sigma}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_m(\lambda)$ is to use the model-based interpretation of LSTD (Boyan, 1999) with $\boldsymbol{A}_n$ as discussed in Section 4.4. In particular, Tsitsiklis and Van Roy (1996) and Nedić and Bertsekas (2003) showed that $\mathbb{E}\left[\boldsymbol{A}_n\right] \to \boldsymbol{D}_{\boldsymbol{\mu}^\pi}\left[\boldsymbol{I}_{|\mathcal{S}|} - \gamma \boldsymbol{P}^\pi\right]$ as $n \to \infty$. The control of the bound on the difference $\|\boldsymbol{A}_n - \boldsymbol{D}_{\boldsymbol{\mu}^\pi}\left[\boldsymbol{I}_{|\mathcal{S}|} - \gamma \boldsymbol{P}^\pi\right]\|$ as a function of n was studied by Tagorti and Scherrer (2015). We make the following assumption on this norm and the number of distinct visited states m.

Assumption 3.

- *As* $N, m \to \infty$, *we have* $\|\boldsymbol{A}_n - \boldsymbol{D}_{\boldsymbol{\mu}^\pi}[\boldsymbol{I}_{|\mathcal{S}|} - \gamma \boldsymbol{P}^\pi]\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$.
- *As* $N, m \to \infty$, *we have* $\frac{m}{|\mathcal{S}|} < \infty$.

With the additional Assumption 3, we can now bound the operator norm of $\frac{1}{m}\boldsymbol{D}_{\boldsymbol{\mu}^\pi}\left[\boldsymbol{I}_{|\mathcal{S}|} - \gamma \boldsymbol{P}^\pi\right]\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_m(\lambda)$ as stated by the following Lemma.

Lemma 7.4.1. *For all* m, *under Assumptions 1, 2 and 3, there exists* $K > 0$ *independent of* N *and* m *such that*

$$\left\|\tfrac{1}{m}\boldsymbol{D}_{\boldsymbol{\mu}^\pi}[\boldsymbol{I}_{|\mathcal{S}|} - \gamma \boldsymbol{P}^\pi]\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_m(\lambda)\right\| \leq K.$$

Proof. We have

$$\begin{aligned}
&\left\|\tfrac{1}{m}\boldsymbol{D}_{\boldsymbol{\mu}^\pi}[\boldsymbol{I}_{|\mathcal{S}|} - \gamma \boldsymbol{P}^\pi]\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_m(\lambda)\right\| \\
&\leq \left\|\tfrac{1}{m}\boldsymbol{U}_n(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_m(\lambda)\right\| + \left\|\tfrac{1}{m}\left[\boldsymbol{D}_{\boldsymbol{\mu}^\pi}[\boldsymbol{I}_{|\mathcal{S}|} - \gamma \boldsymbol{P}^\pi] - \boldsymbol{U}_n(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\right]\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_m(\lambda)\right\| \\
&\leq \underbrace{\left\|\boldsymbol{U}_n\left[\boldsymbol{I}_n - \lambda\boldsymbol{Q}_m(\lambda)\right]\right\|}_{(1)} + \underbrace{\left\|\tfrac{1}{\sqrt{m}}\boldsymbol{U}_n(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\Sigma}_{\mathcal{S}}^T - \tfrac{1}{\sqrt{m}}\boldsymbol{D}_{\boldsymbol{\mu}^\pi}\left[\boldsymbol{I}_{|\mathcal{S}|} - \gamma \boldsymbol{P}^\pi\right]\boldsymbol{\Sigma}_{\mathcal{S}}^T\right\|\left\|\tfrac{1}{\sqrt{m}}\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_m(\lambda)\right\|}_{(2)}.
\end{aligned}$$

From Lemma A.4.1, we know there exists $K_{\boldsymbol{Q}} > 0$ such that, for all m, we have $\|\boldsymbol{Q}_m(\lambda)\| \leq K_{\boldsymbol{Q}}$. For the left-hand part (1), we have

$$\left\|\boldsymbol{U}_n\left[\boldsymbol{I}_n - \lambda\boldsymbol{Q}_m\right]\right\| \leq 1 + \lambda K_{\boldsymbol{Q}}.$$

From Assumption 3, we have for the first term in the right-hand part (2)

$$\begin{aligned}
&\left\|\tfrac{1}{\sqrt{m}}\boldsymbol{U}_n(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\Sigma}_{\mathcal{S}}^T - \tfrac{1}{\sqrt{m}}\boldsymbol{D}_{\boldsymbol{\mu}^\pi}\left[\boldsymbol{I}_{|\mathcal{S}|} - \gamma \boldsymbol{P}^\pi\right]\boldsymbol{\Sigma}_{\mathcal{S}}^T\right\| \\
&\leq \left\|\boldsymbol{U}_n(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T - \boldsymbol{D}_{\boldsymbol{\mu}^\pi}\left[\boldsymbol{I}_{|\mathcal{S}|} - \gamma \boldsymbol{P}^\pi\right]\right\|\left\|\tfrac{1}{\sqrt{m}}\boldsymbol{\Sigma}_{\mathcal{S}}\right\| \\
&= \mathcal{O}(1).
\end{aligned}$$

This result unfolds from Corollary A.7.1.1 since $\|\boldsymbol{\Sigma}_{\mathcal{S}}\| = \mathcal{O}(|\mathcal{S}|) = \mathcal{O}(m)$. For the second term in (2), from Lemma A.4.4, we know there exists a real $K'_{\boldsymbol{Q}} > 0$ such that, for all m, we have

$$\left\|\tfrac{1}{\sqrt{m}}\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_m(\lambda)\right\| = \left\|\tfrac{1}{\sqrt{m}}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m(\lambda)\right\| \leq K'_{\boldsymbol{Q}}.$$

□

Using Lemma 7.2.2 and Lemma 7.4.1, we can show that $\mathrm{MSBE}(\boldsymbol{\theta}_n^\lambda)$ concentrates around $\mathbb{E}_{\boldsymbol{W}}\left[\mathrm{MSBE}(\boldsymbol{\theta}_n^\lambda)\right]$ in the double asymptotic regime. The following theorem provides a deterministic form for the asymptotic $\mathrm{MSBE}(\boldsymbol{\theta}_n^\lambda)$.

Theorem 7.4.2 (Asymptotic MSBE)**.** *Under Assumptions 1, 2, and 3, the deterministic asymptotic MSBE is*

$$\overline{\mathrm{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda) = \left\|\bar{\boldsymbol{r}} + \gamma\tfrac{1}{\sqrt{n}}\tfrac{N}{m}\tfrac{1}{1+\delta}\boldsymbol{P}^\pi\boldsymbol{\Phi}_{\mathcal{S}}\boldsymbol{U}_n\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{r} - \tfrac{1}{\sqrt{n}}\tfrac{N}{m}\tfrac{1}{1+\delta}\boldsymbol{\Phi}_{\mathcal{S}}\boldsymbol{U}_n\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{r}\right\|_{\boldsymbol{D}_{\boldsymbol{\mu}^\pi}}^2 + \Delta, \tag{7.13}$$

with second-order correction factor

$$\Delta = \frac{1}{n}\frac{\frac{1}{N}\operatorname{Tr}\big(\boldsymbol{\Lambda}_{\boldsymbol{P}}\big[\boldsymbol{\Theta}_{\mathcal{S}}\boldsymbol{\Psi}_2\boldsymbol{\Theta}_{\mathcal{S}}^T - 2\boldsymbol{\Theta}_{\mathcal{S}}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\Psi}_{\mathcal{S}} + \boldsymbol{\Psi}_{\mathcal{S}}\big]\big)}{1 - \frac{1}{N}\operatorname{Tr}\big(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m(\lambda)^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m(\lambda)\big)}\|\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2, \tag{7.14}$$

where

$$\boldsymbol{\Psi}_{\mathcal{S}} = \frac{N}{m}\frac{1}{1+\delta}\boldsymbol{\Phi}_{\mathcal{S}}, \quad \boldsymbol{\Lambda}_{\boldsymbol{P}} = [\boldsymbol{I}_{|\mathcal{S}|} - \gamma\boldsymbol{P}^{\pi}]^T\boldsymbol{D}_{\boldsymbol{\mu}^{\pi}}[\boldsymbol{I}_{|\mathcal{S}|} - \gamma\boldsymbol{P}^{\pi}], \quad \textit{and} \quad \boldsymbol{\Theta}_{\mathcal{S}} = \boldsymbol{\Psi}_{\mathcal{S}}\boldsymbol{U}_n\bar{\boldsymbol{Q}}_m(\lambda). \tag{7.15}$$

As $N, m \to \infty$ with asymptotic constant ratio N/m,

$$\mathrm{MSBE}(\hat{\boldsymbol{\theta}}_n^{\lambda}) - \overline{\mathrm{MSBE}}(\hat{\boldsymbol{\theta}}_n^{\lambda}) \xrightarrow{a.s} 0.$$

Proof. The proof can be found in Appendix A.3. □

Remark 24. *Similarly to the empirical $\widehat{\overline{\mathrm{MSBE}}}(\hat{\boldsymbol{\theta}}_n^{\lambda})$ in Theorem 7.3.2, the true $\overline{\mathrm{MSBE}}(\hat{\boldsymbol{\theta}}_n^{\lambda})$ is also influenced by correction terms δ and Δ. Note that the correction terms vanish when $N/m \to \infty$ or $\lambda \to \infty$.*

Remark 25. *When all states have been visited, the common subexpressions in the second-order correction factors $\hat{\Delta}$ and Δ dominate so that $\hat{\Delta}$, Δ become similar (for a proof, see Lemma A.3.7).*

An interpretation of terms in $\overline{\mathrm{MSBE}}(\hat{\boldsymbol{\theta}}_n^{\lambda})$ is provided in Chapter 8.

7.5 Asymptotic Mean-Squared Value Error

In the regularized LSTD with random features, the MSVE defined in equation 3.1 can be rewritten as

$$\mathrm{MSVE}(\hat{\boldsymbol{\theta}}_n^{\lambda}) = \big\|\boldsymbol{V}^{\pi} - \boldsymbol{\Sigma}_{\mathcal{S}}^T\hat{\boldsymbol{\theta}}_n^{\lambda}\big\|_{\boldsymbol{D}_{\boldsymbol{\mu}^{\pi}}}^2. \tag{7.16}$$

Using a similar approach than for Theorem 7.4.2, we can obtain a similar deterministic form for the MSVE in the double asymptotic regime of Assumption 2.

Corollary 7.5.0.1 (Asymptotic MSVE)**.** *Under Assumptions 1, 2, and 3, the deterministic asymptotic MSVE is*

$$\overline{\mathrm{MSVE}}(\hat{\boldsymbol{\theta}}_n^{\lambda}) = \Big\|\boldsymbol{V}^{\pi} - \frac{1}{\sqrt{n}}\frac{N}{m}\frac{1}{1+\delta}\boldsymbol{\Phi}_{\mathcal{S}}\boldsymbol{U}_n\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{r}\Big\|_{\boldsymbol{D}_{\boldsymbol{\mu}^{\pi}}}^2 + \Delta', \tag{7.17}$$

with second-order correction factor

$$\Delta' = \frac{1}{n}\frac{\frac{1}{N}\operatorname{Tr}\big(\boldsymbol{D}_{\boldsymbol{\mu}^{\pi}}\big[\boldsymbol{\Theta}_{\mathcal{S}}\boldsymbol{\Psi}_2\boldsymbol{\Theta}_{\mathcal{S}}^T - 2\boldsymbol{\Theta}_{\mathcal{S}}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\Psi}_{\mathcal{S}} + \boldsymbol{\Psi}_{\mathcal{S}}\big]\big)}{1 - \frac{1}{N}\operatorname{Tr}\big(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m(\lambda)^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m(\lambda)\big)}\|\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2. \tag{7.18}$$

As $N, m, d \to \infty$ with asymptotic constant ratio N/m,

$$\mathrm{MSVE}(\hat{\boldsymbol{\theta}}_n^{\lambda}) - \overline{\mathrm{MSVE}}(\hat{\boldsymbol{\theta}}_n^{\lambda}) \xrightarrow{a.s} 0.$$

Proof. Using $\boldsymbol{D}_{\boldsymbol{\mu}^\pi} = [\boldsymbol{I}_m - \gamma \boldsymbol{P}^\pi]^T \boldsymbol{D}_{\boldsymbol{\mu}^\pi} \boldsymbol{D}_{\boldsymbol{\mu}^\pi}^{-1} [\boldsymbol{I}_m - \gamma \boldsymbol{P}^\pi]^{-1T} \boldsymbol{D}_{\boldsymbol{\mu}^\pi} [\boldsymbol{I}_m - \gamma \boldsymbol{P}^\pi]^{-1} \boldsymbol{D}_{\boldsymbol{\mu}^\pi}^{-1} \boldsymbol{D}_{\boldsymbol{\mu}^\pi} [\boldsymbol{I}_m - \gamma \boldsymbol{P}^\pi]$ and a with similar proof than for Theorem 7.4.2, we find Corollary 7.5.0.1. □

Remark 26. *Similarly to* $\widehat{\overline{\mathrm{MSBE}}}(\hat{\boldsymbol{\theta}}_n^\lambda)$ *in Theorem 7.3.2 and* $\overline{\mathrm{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda)$ *in Theorem 7.4.2, the* $\overline{\mathrm{MSVE}}(\hat{\boldsymbol{\theta}}_n^\lambda)$ *is also influenced by the correction terms* δ *and* Δ'*. Note that the correction terms vanish when* $N/m \to \infty$ *or* $\lambda \to \infty$*.*

An interpretation of terms in $\overline{\mathrm{MSVE}}(\hat{\boldsymbol{\theta}}_n^\lambda)$ is provided in the following chapter.

Chapter 8

Implicit Regularization

Random feature models are efficient parametric approximations of kernel methods (Rahimi and Recht, 2007). In this chapter, we investigate the connections between regularized kernel LSTD and regularized LSTD using random features in the double asymptotic regime of Assumption 2. We begin in Section 8.1 by reviewing kernel methods, their corresponding Mercer feature space, and the regularized kernel LSTD. In Section 8.2, we revisit and rewrite the asymptotic results found in Chapter 7 in the Mercer feature space approximated by the random features. Reformulation of the asymptotic error functions of the regularized LSTD with random features are summarized in Theorem 8.3.1 and discussed in Section 8.3. In particular, we show how this reformulation provides a better understanding of correction terms that arise from the double asymptotic regime and highlights an implicit regularization induced by the model complexity N/m of Assumption 2.

8.1 Kernel Methods in Reinforcement Learning

Kernel methods are a class of algorithms in machine learning (Schölkopf and Smola, 2002). The term "kernel methods" comes from their use of *kernel functions*. These functions allow the methods to work in an implicit high-dimensional feature space without the need to compute the data's coordinates in the implicit feature space. Instead, kernel functions are similarity functions, i.e., they exploit similarities between all data pairs in the high-dimensional feature space to perform a wide range of tasks, such as classification, clustering, or principal component analysis.

8.1.1 The Reproducing Kernel Hilbert Space

Definition 8.1.1 (Reproducing Kernel Hilbert Space)**.** *Let $\mathcal{C}$ be a nonempty set and $\mathcal{H} = \{f, f : \mathcal{C} \to \mathbb{R}\}$ be a Hilbert space of functions. The Hilbert space $\mathcal{H}$ is a Reproducing Kernel Hilbert Space (RKHS) equipped with the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ if there exists a function $K : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$, called reproducing kernel, such that*

- *$\forall f \in \mathcal{H}, \forall x \in \mathcal{C}$, we have*
$$\langle K(\cdot, x), f \rangle_{\mathcal{H}} = f(x), \tag{8.1}$$
- *K spans $\mathcal{H}$, i.e., $\mathcal{H} = \overline{\text{span}\{K(x, \cdot) \mid x \in \mathcal{C}\}}$, where $\overline{\mathcal{X}}$ denotes the completion of the set $\mathcal{X}$.*

Let $K : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$ be the reproducing kernel of the RKHS $\big(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}}\big)$. For all $x, y \in \mathcal{C}$, we have from the reproducing property (equation 8.1)

$$K(x, y) = \langle K(\cdot, x), K(\cdot, y) \rangle_{\mathcal{H}} \tag{8.2}$$

as $K(x, \cdot), K(y, \cdot) \in \mathcal{H}$. From equation 8.2, we deduce that K is a positive-definite kernel.

Definition 8.1.2 (Positive-Definite Kernel)**.** *A positive-definite kernel is a symmetric function $K : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$, such that for any integer $n \geq 1$ and for any $x_1, \ldots, x_n \in \mathcal{C}$, the Gram kernel matrix $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ with entries $\boldsymbol{K}_{ij} = K(x_i, x_j)$ is positive semi-definite (PSD).*

According to equation 8.2, all reproducing kernels are positive-definite kernels. The converse is also true, as stated in the following theorem.

Theorem 8.1.1 (Moore-Aronszajn (Schölkopf and Smola, 2002))**.** *For any positive-definite kernel $K : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$, there exists a unique reproducing kernel Hilbert space (RKHS) $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ with reproducing kernel K.*

Linear parametric representations of functions are a particular case of RKHS. Consider $\big\{\boldsymbol{\sigma}_i(\cdot)\big\}_{i=1}^N$ the basis of features used in equation 3.8. The associated Hilbert space $\mathcal{H} = \text{span}\big\{\big\{\boldsymbol{\sigma}_i(\cdot)\big\}_{i=1}^N \mid x \in \mathcal{C}\big\}$ with the euclidean dot product $\langle \cdot, \cdot \rangle$ forms a subset of $\mathbb{R}^N$. In the Hilbert space $\mathcal{H}$, each function $f \in \mathcal{H}$ can be uniquely represented by a parameter vector $\boldsymbol{\theta} \in \mathbb{R}^N$. Furthermore, the following reproducing property holds for all $x \in \mathcal{C}$

$$f(x) = \big\langle \boldsymbol{\theta}, \boldsymbol{\sigma}(x) \big\rangle = \big\langle f, K(\cdot, x) \big\rangle,$$

for the linear positive-definite kernel $K : \mathcal{C} \times \mathcal{C} \to \mathbb{R}, (x, y) \mapsto \big\langle \boldsymbol{\sigma}(x), \boldsymbol{\sigma}(y) \big\rangle$. Thus, the space $\mathcal{H}$ of linear parametric representations is a RKHS whose reproducing kernel is K.

In any RKHS $\mathcal{H}$, the reproducing property can be interpreted as an extension of the linear parameterization. Specifically, for all $f \in \mathcal{H}$ and for all $x \in \mathcal{C}$, the linear parameterization expression $f(x) = \big\langle \boldsymbol{\theta}, \boldsymbol{\sigma}(x) \big\rangle$ can be replaced by

$$f(x) = \langle f, K(\cdot, x) \rangle_{\mathcal{H}} = \langle f, \boldsymbol{\sigma}(x) \rangle_{\mathcal{H}}, \tag{8.3}$$

where $\boldsymbol{\sigma}(x) = K(\cdot, x)$ depicts the *feature map* of x in $\mathcal{H}$. In equation 8.3, the representation of f is non-parametric, meaning that it is not represented by a vector $\boldsymbol{\theta} \in \mathbb{R}^N$, but directly by being an element of $\mathcal{H}$, which may be a possibly infinite-dimensional Hilbert space. As indicated by equation 8.2, any positive-definite kernel K can be expressed using the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and the feature map $\boldsymbol{\sigma} : x \mapsto K(\cdot, x)$ as

$$K(x, y) = \langle \boldsymbol{\sigma}(x), \boldsymbol{\sigma}(y) \rangle_{\mathcal{H}}, \qquad \forall x, y \in \mathcal{C}.$$

The representation of K via a feature map and the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is not unique. The following theorem represents K in terms of the Mercer feature map and the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$.

Theorem 8.1.2 (Mercer Theorem (Schölkopf and Smola, 2002))**.** *Let $\mathcal{C}$ be a real nonempty compact set equipped with a positive Borel measure μ and $\mathcal{H} = \{f, f : \mathcal{C} \to \mathbb{R}\}$ be a Hilbert space of functions. Let $K \in L^2(\mathcal{C} \times \mathcal{C}, \mu)$ be a positive-definite kernel such that*

$$\int_{\mathcal{C}} \int_{\mathcal{C}} K(x, y)^2 \mu(x) \mu(y) dx dy < \infty$$

and its associated Hilbert-Schmidt integral operator

$$T_K : L^2(\mathcal{C}, \mu) \to L^2(\mathcal{C}, \mu)$$
$$T_K(f)(x) = \int_{\mathcal{C}} K(x, x') f(x) \mu(x) dx.$$

Then, there is a set of orthonormal bases $\{\varphi_i\}_{i=1}^M$ *of* $L^2(\mathcal{C}, \mu)$ *consisting of eigenfunctions of* T_K *associated with the non-decreasing sequence of non-negative eigenvalues* $\{\nu_i\}_{i=1}^M$

$$T_K(\psi_i)(x) = \int_{\mathcal{C}} K(x, y) \varphi_i(x) dy = \nu_i \varphi_i(x).$$

K *can be represented for all* $x, y \in \mathcal{C}$ *as*

$$K(x, y) = \sum_{i=1}^{M} \nu_i \varphi_i(x) \varphi_i(y). \tag{8.4}$$

Either M *is an integer or infinite; in the latter case, the series converges absolutely and uniformly for almost all* (x, y).

Let $\left\{\omega_i(\cdot) = \sqrt{\nu_i}\varphi_i(\cdot)\right\}_{i=1}^M$ be the rescaled orthogonal eigenfunction basis of $L^2(\mathcal{C}, \mu)$. Equation 8.4 can be rewritten as

$$K(x, y) = \left\langle \boldsymbol{\omega}(x), \boldsymbol{\omega}(y) \right\rangle_{\mathcal{H}}, \qquad \forall x, y \in \mathcal{C},$$

for the feature map $\boldsymbol{\omega} : \mathcal{C} \to \mathcal{H}$, $x \mapsto \left(\omega_i(x)\right)_{i=1}^M$. In the literature, the feature map $\boldsymbol{\omega} : \mathcal{C} \to \mathcal{H}$ is referred to as the *Mercer feature map*.

Since $\mathcal{H}$ may be an infinite-dimensional Hilbert space, designing function approximation algorithms considering the representation of $\mathcal{H}$ with feature maps is challenging for practical computations. However, as detailed in the following section, this representation is never explicitly considered, and many computations only involve evaluations of the kernel function. This phenomenon is called the "*kernel trick*".

8.1.2 Regularized Kernel LSTD

Like their parameterized counterparts, non-parametric TD learning methods are value-based algorithms. The objective of non-parametric TD learning methods is to approximate the value function $V^\pi : \mathcal{S} \to \mathbb{R}$ of a MRP $(\mathcal{S}, P^\pi, R^\pi, \mu_0)$ with an element $\hat{V}$ in a RKHS $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ by minimizing the Mean-Squared Projected Bellman error (equation 3.4)

$$\text{MSPBE}(V) = \mathbb{E}_{s \sim \mu^\pi}\left[\left(\Pi \mathcal{T}_V^\pi V(s) - V(s)\right)^2\right],$$

where Π is the projection operator into the RKHS $\mathcal{H}$ defined as

$$\Pi f = \min_{f' \in \mathcal{H}} \mathbb{E}_{s \sim \mu^\pi}\left[f'(s) - f(s)\right],$$

which projects arbitrary value functions onto the RKHS $\mathcal{H}$. Since the transition function P^π is assumed unknown, non-parametric TD learning methods try to minimize an empirical version of the Mean-Squared Bellman error on n transitions $\mathcal{D}_{\text{train}} := \left\{(\boldsymbol{s}_i, r_i, \boldsymbol{s}_i')\right\}_{i=1}^n$ consisting of states, rewards, and next-states drawn from the MRP, where $\boldsymbol{s}_i' \sim P^\pi(\boldsymbol{s}_i, \cdot)$ and $r_i = R^\pi(\boldsymbol{s}_i, \boldsymbol{s}_i')$. In

particular, their objective is to find $\hat{V}_n \in \mathcal{H}$ that minimizes the following training error

$$\begin{aligned} E_{\text{train}}(V) &= \tfrac{1}{n}\sum_{i=1}^{n}\big(r_i + \gamma V(\boldsymbol{s}_i') - V(\boldsymbol{s}_i)\big)^2 + \lambda n \|V\|_{\mathcal{H}}^2 \\ &= \widehat{\text{MSBE}}(V) + \lambda n \|V\|_{\mathcal{H}}^2, \end{aligned} \tag{8.5}$$

where $\widehat{\text{MSBE}}(V) = \frac{1}{n}\sum_{i=1}^{n}\big(r_i + \gamma V(\boldsymbol{s}_i') - V(\boldsymbol{s}_i)\big)^2$ is the empirical MSBE defined on $\mathcal{D}_{\text{train}}$. The quadratic regularization term in equation 8.5 is crucial to apply the following representer theorem and guarantees the existence of a specific solution $\hat{V}_n \in \mathcal{H}$.

Theorem 8.1.3 (Representer Theorem (Schölkopf and Smola, 2002)). *Let $K : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$ be positive-definite real-value kernel on a nonempty set $\mathcal{C}$ with a corresponding RKHS $\big(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}}\big)$. Let $\mathcal{D}_{train} = \big\{(x_i, y_i) \in \mathcal{C} \times \mathbb{R}\big\}_{i=1}^{n}$ be training samples, a strictly increasing real-valued function $g : [0, \infty) \to \mathbb{R}$, and an arbitrary error function $E_{train} : (\mathcal{C} \times \mathbb{R})^n \to \mathbb{R} \cup \infty$. Then any minimzer $f^* \in \mathcal{H}$ of the regularized risk*

$$\min_{f \in \mathcal{H}} E_{train}\big((x_1, y_1), \ldots, (x_n, y_n)\big) + g\big(\|f\|_{\mathcal{H}}\big),$$

admits a representation of the form

$$f^*(\cdot) = \sum_{i=1}^{n} \boldsymbol{\alpha}_i K(\cdot, x_i),$$

for some $\boldsymbol{\alpha} \in \mathbb{R}^n$.

Let $K : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ be the reproducing kernel of the RKHS $\mathcal{H}$. From the representer theorem, solving equation 8.5 gives the regularized kernel LSTD solution (Duan et al., 2021)

$$\hat{V}_n(\boldsymbol{s}) = \boldsymbol{k}(\boldsymbol{s})\big[\boldsymbol{K}(\boldsymbol{X}_n, \boldsymbol{X}_n) - \gamma \boldsymbol{K}(\boldsymbol{X}_n', \boldsymbol{X}_n) + \lambda n \boldsymbol{I}_n\big]^{-1}\boldsymbol{r}, \qquad \forall \boldsymbol{s} \in \mathcal{S}; \tag{8.6}$$

where $\boldsymbol{K}(\boldsymbol{X}_n, \boldsymbol{X}_n), \boldsymbol{K}(\boldsymbol{X}_n', \boldsymbol{X}_n) \in \mathbb{R}^{n \times n}$ are defined for all $i, j \in [n]$ as

$$\big[\boldsymbol{K}(\boldsymbol{X}_n, \boldsymbol{X}_n)\big]_{ij} = K(\boldsymbol{s}_i, \boldsymbol{s}_j) \qquad \text{and} \qquad \big[\boldsymbol{K}(\boldsymbol{X}_n', \boldsymbol{X}_n)\big]_{ij} = K(\boldsymbol{s}_i', \boldsymbol{s}_j);$$

and $\boldsymbol{k}(s) \in \mathbb{R}^n$ is defined for all $\boldsymbol{s} \in \mathcal{S}$ as

$$\big[\boldsymbol{k}(\boldsymbol{s})\big]_i = K(\boldsymbol{s}, \boldsymbol{s}_i), \qquad \forall i \in [n].$$

Using auxiliary matrices $\hat{\boldsymbol{U}}_n, \hat{\boldsymbol{V}}_n \in \mathbb{R}^{m \times n}$ of equation 6.7, we can rewrite equation 8.6 as

$$\hat{V}_n(\boldsymbol{s}) = \tfrac{1}{\sqrt{n}}\hat{\boldsymbol{k}}(\boldsymbol{s})\hat{\boldsymbol{U}}_n\big[(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \boldsymbol{K}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n + \lambda \boldsymbol{I}_n\big]^{-1}\boldsymbol{r}, \qquad \forall \boldsymbol{s} \in \mathcal{S}, \tag{8.7}$$

where $\boldsymbol{K}_{\hat{\mathcal{S}}} \in \mathbb{R}^{m \times m}$ is the kernel Gram matrix on the set of distinct visited states $\hat{\mathcal{S}}$ and $\hat{\boldsymbol{k}} : \mathcal{S} \to \mathbb{R}^m$ is defined, for all $\boldsymbol{s} \in \mathcal{S}$, as

$$\big[\hat{\boldsymbol{k}}(\boldsymbol{s})\big]_i = K(\boldsymbol{s}, \hat{\boldsymbol{S}}_i), \qquad \forall i \in [m].$$

Note that the invertibility of the matrix $\big[(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \boldsymbol{K}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n + \lambda \boldsymbol{I}_n\big]$ is guaranteed under Assumption 2 (see Appendix A.5 for a formal proof).

8.2 Reformulation of the Main Results

In Chapter 7 — for the solution $\hat{\boldsymbol{\theta}}_n^\lambda$ of the regularized LSTD with random features on $\mathcal{D}_{\text{train}}$ (equation 6.12) — we have shown under Assumptions 1, 2 and 3 that

$$\begin{aligned}
\widehat{\overline{\text{MSBE}}}(\hat{\boldsymbol{\theta}}_n^\lambda) &\xrightarrow{a.s} \widehat{\overline{\text{MSBE}}}(\hat{\boldsymbol{\theta}}_n^\lambda) \quad \text{(Theorem 7.3.2)},\\
\text{MSBE}(\hat{\boldsymbol{\theta}}_n^\lambda) &\xrightarrow{a.s} \overline{\text{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda) \quad \text{(Theorem 7.4.2)},\\
\text{MSVE}(\hat{\boldsymbol{\theta}}_n^\lambda) &\xrightarrow{a.s} \overline{\text{MSVE}}(\hat{\boldsymbol{\theta}}_n^\lambda) \quad \text{(Corollary 7.5.0.1)};
\end{aligned}$$

where

$$\begin{aligned}
\widehat{\overline{\text{MSBE}}}(\hat{\boldsymbol{\theta}}_n^\lambda) &= \tfrac{1}{n}\left\|\boldsymbol{r} - \tfrac{1}{\sqrt{n}}\tfrac{N}{m}\tfrac{1}{1+\delta}\big(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n\big)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{r}\right\|^2 + \hat{\Delta} \quad \text{(equation 7.10)},\\
\overline{\text{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda) &= \left\|\bar{\boldsymbol{r}} - \tfrac{1}{\sqrt{n}}\tfrac{N}{m}\tfrac{1}{1+\delta}\big[\boldsymbol{I}_{|\mathcal{S}|} - \gamma\boldsymbol{P}^\pi\big]\boldsymbol{\Phi}_{\mathcal{S}}\boldsymbol{U}_n\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{r}\right\|^2_{\boldsymbol{D}_{\mu^\pi}} + \Delta \quad \text{(equation 7.13)},\\
\overline{\text{MSVE}}(\hat{\boldsymbol{\theta}}_n^\lambda) &= \left\|\boldsymbol{V}^\pi - \tfrac{1}{\sqrt{n}}\tfrac{N}{m}\tfrac{1}{1+\delta}\boldsymbol{\Phi}_{\mathcal{S}}\boldsymbol{U}_n\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{r}\right\|^2_{\boldsymbol{D}_{\mu^\pi}} + \Delta' \quad \text{(equation 7.17)}.
\end{aligned}$$

Each asymptotic error function is expressed as the sum of two terms. In Section 8.2.1, we show that the first term can be interpreted as the error function of a regularized kernel LSTD with an implicit l_2-regularization term $\tilde{\lambda}$, depending on the l_2-regularization parameter λ and the model complexity N/m. In Section 8.2.3, we reformulate the second-order correction factors $\hat{\Delta}, \Delta$, and Δ' into a unified expression in the Mercer feature space induced by the regularized kernel LSTD.

8.2.1 Connection with the Regularized Kernel LSTD

Asymptotic error functions found in Chapter 7 depend on the Gram feature matrices $\boldsymbol{\Phi}_{\hat{\mathcal{S}}}$ and $\boldsymbol{\Phi}_{\mathcal{S}}$ of the continuous kernel function $\Phi : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ defined for all states $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}$ as

$$\Phi(\boldsymbol{s}, \boldsymbol{s}') = \mathbb{E}_{\boldsymbol{w}}\left[\sigma(\boldsymbol{w}^T\boldsymbol{s})\sigma(\boldsymbol{w}^T\boldsymbol{s}')\right]. \tag{8.8}$$

We can observe that the first terms in $\widehat{\overline{\text{MSBE}}}(\hat{\boldsymbol{\theta}}_n^\lambda)$, $\overline{\text{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda)$ and $\overline{\text{MSVE}}(\hat{\boldsymbol{\theta}}_n^\lambda)$ are equivalent to the empirical MSBE, MSBE, and MSVE of the regularized kernel LSTD solution $\hat{V}_n^{\tilde{\lambda}}$ on $\mathcal{D}_{\text{train}}$ defined for all states $\boldsymbol{s} \in \mathcal{S}$ as

$$\hat{V}_n^{\tilde{\lambda}}(\boldsymbol{s}) = \tfrac{1}{\sqrt{n}}\hat{\boldsymbol{\phi}}(\boldsymbol{s})\hat{\boldsymbol{U}}_n\big[(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n + \tilde{\lambda}\boldsymbol{I}_n\big]^{-1}\boldsymbol{r}; \tag{8.9}$$

where

$$\tilde{\lambda} = \lambda\tfrac{m}{N}(1+\delta); \tag{8.10}$$

and $\hat{\boldsymbol{\phi}} : \mathcal{S} \to \mathbb{R}^m$ is mapping from states to vectors in $\mathbb{R}^m$, where each element i in $[m]$ is defined as

$$\big[\hat{\boldsymbol{\phi}}(\boldsymbol{s})\big]_i = \Phi(\boldsymbol{s}, \hat{\boldsymbol{S}}_i).$$

Asymptotic error functions found in Chapter 7 can be thus rewritten with $\hat{V}_n^{\tilde{\lambda}}$ as

$$\begin{aligned}
\widehat{\overline{\text{MSBE}}}(\hat{\boldsymbol{\theta}}_n^\lambda) &= \widehat{\text{MSBE}}\big(\hat{V}_n^{\tilde{\lambda}}\big) + \hat{\Delta},\\
\overline{\text{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda) &= \text{MSBE}\big(\hat{V}_n^{\tilde{\lambda}}\big) + \Delta,
\end{aligned}$$

$$\overline{\text{MSVE}}(\hat{\boldsymbol{\theta}}_n^{\lambda}) = \text{MSVE}\big(\hat{V}_n^{\tilde{\lambda}}\big) + \Delta'.$$

We will find it convenient to work with the Mercer map of the kernel $\Phi(\cdot,\cdot)$ to simplify expressions of second-order correction factors.

8.2.2 Reformulation of the Regularized Kernel LSTD in the Mercer Feature Space

From the Mercer theorem (Theorem 8.1.2), under the assumption that the state space $\mathcal{S}$ is compact, the continuous kernel $\Phi(\cdot,\cdot)$ can be represented for all states $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}$ as

$$\Phi(\boldsymbol{s},\boldsymbol{s}') = \sum_{i=1}^{M} \nu_i \varphi_i(\boldsymbol{s})\varphi_i(\boldsymbol{s}') = \sum_{i=1}^{M} \omega_i(\boldsymbol{s})\omega_i(\boldsymbol{s}');$$

where $\{\nu_i\}_{i=1}^M$, $\{\varphi_i(\cdot)\}_{i=1}^M$, and $\{\omega_i(\cdot) = \sqrt{\nu_i}\varphi_i(\cdot)\}_{i=1}^M$ are the eigenvalues, eigenfunctions, and rescaled eigenfunctions of the Hilbert-Schmidt integral operators $T_\Phi : L^2(\mathcal{S},\mu^\pi) \to L^2(\mathcal{S},\mu^\pi), f \mapsto T_\Phi(f)(\boldsymbol{s}') = \int_{\mathbb{R}^d} \Phi(\boldsymbol{s},\boldsymbol{s}')f(\boldsymbol{s})\mu^\pi(\boldsymbol{s})d\boldsymbol{s}$. Usually, M is infinite. As $\{\omega_i(\cdot) = \sqrt{\nu_i}\varphi_i(\cdot)\}_{i=1}^M$ forms an orthogonal basis in $L^2(\mathcal{S},\mu^\pi)$, we will find it convenient to work on a vector representation of functions in $L^2(\mathcal{S},\mu^\pi)$ with the Mercer feature map

$$\begin{aligned}\boldsymbol{\omega}: \quad & \mathcal{S} \to \mathbb{R}^M \\ & \boldsymbol{s} \mapsto \big[\omega_1(\boldsymbol{s}),\dots,\omega_M(\boldsymbol{s})\big]^T.\end{aligned}$$

For any state matrix $\boldsymbol{A} \in \mathbb{R}^{d\times p}$, we denote by $\boldsymbol{\Omega}_{\boldsymbol{A}} \in \mathbb{R}^{M\times p}$ the Mercer feature matrix of $\boldsymbol{A}$ so that $[\boldsymbol{\Omega}_{\boldsymbol{A}}]_{ij} = \omega_i(\boldsymbol{A}_j)$, for $\boldsymbol{A}_j$ the j^{th} column of $\boldsymbol{A}$. With those new notations, we can decompose $\boldsymbol{\Phi}_{\hat{\mathcal{S}}}$ and $\boldsymbol{\Phi}_{\mathcal{S}}$ as

$$\boldsymbol{\Phi}_{\hat{\mathcal{S}}} = \boldsymbol{\Omega}_{\hat{\mathcal{S}}}^T\boldsymbol{\Omega}_{\hat{\mathcal{S}}} \qquad \text{and} \qquad \boldsymbol{\Phi}_{\mathcal{S}} = \boldsymbol{\Omega}_{\mathcal{S}}^T\boldsymbol{\Omega}_{\mathcal{S}}.$$

By using the Mercer feature map, we can write the vector representation of the regularized kernel LSTD solution $\hat{V}_n^{\tilde{\lambda}}$ on $\mathcal{D}_{\text{train}}$ (equation 8.9) as

$$\hat{V}_n^{\tilde{\lambda}}(\boldsymbol{s}) = \boldsymbol{\omega}(\boldsymbol{s})^T\bar{\boldsymbol{\theta}}_n(\tilde{\lambda}), \qquad \forall \boldsymbol{s} \in \mathcal{S};$$

where

$$\bar{\boldsymbol{\theta}}_n(\tilde{\lambda}) = \tfrac{1}{\sqrt{n}}\big[\boldsymbol{\Omega}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Omega}_{\hat{\mathcal{S}}}^T + \tilde{\lambda}\boldsymbol{I}_n\big]^{-1}\boldsymbol{\Omega}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{r} \tag{8.11}$$

is the vector representation of $\hat{V}_n^{\tilde{\lambda}}$ in the Mercer feature space. Under Assumption 2 and from the model-based interpretation of $\bar{\boldsymbol{\theta}}_n(\tilde{\lambda})$ (see Remark 11), we can rewrite $\bar{\boldsymbol{\theta}}_n(\tilde{\lambda})$ as

$$\bar{\boldsymbol{\theta}}_n(\tilde{\lambda}) = \big[\boldsymbol{\Omega}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Omega}_{\hat{\mathcal{S}}}^T + \tilde{\lambda}\boldsymbol{I}_n\big]^{-1}\boldsymbol{\Omega}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\hat{\boldsymbol{U}}_n^T\hat{\bar{\boldsymbol{r}}}, \tag{8.12}$$

where $\hat{\bar{\boldsymbol{r}}}_i = \sum_{j=1}^m [\hat{\boldsymbol{P}}_n]_{ij}R^\pi(\hat{\boldsymbol{S}}_i,\hat{\boldsymbol{S}}_j)$, for all $i \in [m]$. The value function $V_n : \hat{\mathcal{S}} \to \mathbb{R}$ of the empirical MRP $(\hat{\mathcal{S}},\hat{\boldsymbol{P}}_n,R^\pi,\mu_0)$ is the unique-fixed point of the Bellman equation (equation 4.9) and its vector form $\boldsymbol{V}_n \in \mathbb{R}^m$ on $\hat{\mathcal{S}}$ is defined as

$$\boldsymbol{V}_n = \big[\boldsymbol{I}_m - \gamma\hat{\boldsymbol{P}}_n\big]^{-1}\hat{\bar{\boldsymbol{r}}} = \big[\boldsymbol{I}_m - \gamma\big[\hat{\boldsymbol{U}}_n\hat{\boldsymbol{U}}_n^T\big]^{-1}\hat{\boldsymbol{U}}_n\hat{\boldsymbol{V}}_n^T\big]^{-1}\hat{\bar{\boldsymbol{r}}}.$$

Putting $\boldsymbol{V}_n$ into equation 8.12 gives

$$\begin{aligned}\bar{\boldsymbol{\theta}}_n(\tilde{\lambda}) &= \left[\boldsymbol{\Omega}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Omega}_{\hat{\mathcal{S}}}^T + \tilde{\lambda}\boldsymbol{I}_n\right]^{-1}\boldsymbol{\Omega}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\hat{\boldsymbol{U}}_n^T\left[\boldsymbol{I}_m - \gamma\left[\hat{\boldsymbol{U}}_n\hat{\boldsymbol{U}}_n^T\right]^{-1}\hat{\boldsymbol{U}}_n\hat{\boldsymbol{V}}_n^T\right]\boldsymbol{V}_n \\ &= \left[\boldsymbol{\Omega}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Omega}_{\hat{\mathcal{S}}}^T + \tilde{\lambda}\boldsymbol{I}_n\right]^{-1}\boldsymbol{\Omega}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{V}_n.\end{aligned} \tag{8.13}$$

Since the eigenbasis $\{\omega_i(\cdot)\}_{i=1}^M$ forms an orthogonal basis of $L^2(\mathcal{S}, \mu^\pi)$, we can rewrite the continuous extension $V_n^{\mathcal{S}} : \mathcal{S} \to \mathbb{R}$ on the state space $\mathcal{S}$ of $V_n : \hat{\mathcal{S}} \to \mathbb{R}$ in its vector form as

$$V_n^{\mathcal{S}}(\boldsymbol{s}) = \boldsymbol{\omega}(s)^T\bar{\boldsymbol{\theta}}_n^*, \quad \forall \boldsymbol{s} \in \mathcal{S}, \tag{8.14}$$

where $\bar{\boldsymbol{\theta}}_n^* \in \mathbb{R}^M$. Using this vector from, equation 8.12 can be rewritten as

$$\bar{\boldsymbol{\theta}}_n(\tilde{\lambda}) = \boldsymbol{\Pi}(\tilde{\lambda})\bar{\boldsymbol{\theta}}_n^*,$$

where $\boldsymbol{\Pi}(\tilde{\lambda}) \in \mathbb{R}^{M\times M}$ is the *hat matrix*[1] defined as

$$\boldsymbol{\Pi}(\tilde{\lambda}) = \left[\boldsymbol{\Omega}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Omega}_{\hat{\mathcal{S}}}^T + \tilde{\lambda}\boldsymbol{I}_M\right]^{-1}\boldsymbol{\Omega}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Omega}_{\hat{\mathcal{S}}}^T. \tag{8.15}$$

Remark 27. *Note that for $\tilde{\lambda} = 0$, we have $\boldsymbol{\Pi}(0) = \boldsymbol{I}_M$ and $\hat{V}_n^{\tilde{\lambda}}(\boldsymbol{s}) = \boldsymbol{\omega}(\boldsymbol{s})^T\boldsymbol{\theta}_n^* = V_n^{\mathcal{S}}(\boldsymbol{s})$, for all states $\boldsymbol{s} \in \mathcal{S}$. In particular, for all states $\boldsymbol{s} \in \hat{\mathcal{S}}$, we have $\hat{V}_n^{\tilde{\lambda}}(\boldsymbol{s}) = V_n^{\mathcal{S}}(\boldsymbol{s}) = V_n(\boldsymbol{s})$, which corresponds to the value function $V_n : \hat{\mathcal{S}} \to \mathbb{R}$ of the empirical MRP $(\hat{\mathcal{S}}, \hat{\boldsymbol{P}}_n, R^\pi, \mu_0)$.*

Remark 28. *The hat matrix $\boldsymbol{\Pi}(\lambda)$ can be interpreted for the regularized LSTD as an extension of the reconstruction operator of Jacot et al. (2020b) or as the learning transfer matrix of Simon et al. (2023a) proposed in ridge regression.*

8.2.3 Reformulation of the Second-Order Correction Factors

Using the Mercer map of the kernel $\Phi(\cdot,\cdot)$, we can reformulate the following second-order correction factors (found in Chapter 7)

$$\begin{aligned}\hat{\Delta} &= \frac{\lambda^2}{n}\frac{\frac{1}{N}\operatorname{Tr}\left(\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m(\lambda)^T\right)}{1-\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m(\lambda)^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m(\lambda)\right)}\|\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2 \quad \text{(equation 7.11)},\\ \Delta &= \frac{1}{n}\frac{\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Lambda}_{\boldsymbol{P}}\left[\boldsymbol{\Theta}_{\mathcal{S}}\boldsymbol{\Psi}_2\boldsymbol{\Theta}_{\mathcal{S}}^T - 2\boldsymbol{\Theta}_{\mathcal{S}}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\Psi}_{\mathcal{S}} + \boldsymbol{\Psi}_{\mathcal{S}}\right]\right)}{1-\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m(\lambda)^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m(\lambda)\right)}\|\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2 \quad \text{(equation 7.14)}\\ \Delta' &= \frac{1}{n}\frac{\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{D}_{\boldsymbol{\mu}^\pi}\left[\boldsymbol{\Theta}_{\mathcal{S}}\boldsymbol{\Psi}_2\boldsymbol{\Theta}_{\mathcal{S}}^T - 2\boldsymbol{\Theta}_{\mathcal{S}}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\Psi}_{\mathcal{S}} + \boldsymbol{\Psi}_{\mathcal{S}}\right]\right)}{1-\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m(\lambda)^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m(\lambda)\right)}\|\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2 \quad \text{(equation 7.18)}.\end{aligned}$$

Indeed, as detailed in Appendix A.7.1, we can show that

$$\begin{aligned}\hat{\Delta} &= \bar{\Delta}\left((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Omega}_{\hat{\mathcal{S}}}^T\right),\\ \Delta &= \bar{\Delta}\left(\boldsymbol{D}_{\mu^\pi}^{\frac{1}{2}}\left[\boldsymbol{I}_{|\mathcal{S}|} - \gamma\boldsymbol{P}^\pi\right]\boldsymbol{\Omega}_{\mathcal{S}}^T\right),\\ \Delta' &= \bar{\Delta}\left(\boldsymbol{D}_{\mu^\pi}^{\frac{1}{2}}\boldsymbol{\Omega}_{\mathcal{S}}^T\right);\end{aligned}$$

where $\bar{\Delta}(\boldsymbol{M})$ is defined, for any Mercer feature matrix $\boldsymbol{M} \in \mathbb{R}^{p\times M}$ of dimension $p > 0$, as

$$\bar{\Delta}(\boldsymbol{M}) = \frac{1}{n}\frac{\tilde{\lambda}}{\lambda}\frac{\frac{1}{N}\left\|\boldsymbol{\Pi}(\tilde{\lambda})\bar{\boldsymbol{\theta}}_n^*\right\|^2}{1-\frac{1}{N}\left\|\boldsymbol{\Pi}(\tilde{\lambda})\right\|_F^2}\left\|\boldsymbol{M}\left[\boldsymbol{I}_M - \boldsymbol{\Pi}(\tilde{\lambda})\right]\right\|_F^2.$$

[1] The notion of hat matrix was defined for the first time in ridge regression in (Hoaglin and Welsch, 1978)

Second-order correction factors $\hat{\Delta}, \Delta$ and Δ' share a similar expression defined by $\bar{\Delta}(\cdot)$. In particular, they depend on the hat matrix $\mathbf{\Pi}(\hat{\lambda})$ (equation 8.15) of the regularized kernel LSTD solution $\hat{V}_n^{\tilde{\lambda}}$ on $\mathcal{D}_{\text{train}}$ (equation 8.9) and on its vector representation $\bar{\boldsymbol{\theta}}_n(\tilde{\lambda}) = \mathbf{\Pi}(\tilde{\lambda})\boldsymbol{\theta}_n^*$ in the Mercer feature space.

8.3 Interpretation

The following theorem summarizes the reformulation of results found in Chapter 7 in the Mercer feature space described in Section 8.2.2.

Theorem 8.3.1. *Under Assumptions 1, 2 and 3, we define the implicit l_2-regularization parameter*

$$\tilde{\lambda} = \lambda \tfrac{m(1+\delta)}{N}. \tag{8.16}$$

and the continuous kernel function $\Phi : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ defined as

$$\Phi(\boldsymbol{s}, \boldsymbol{s}') = \mathbb{E}_{\boldsymbol{w}}\left[\sigma(\boldsymbol{w}^T\boldsymbol{s})\sigma(\boldsymbol{w}^T\boldsymbol{s}')\right], \quad \forall \boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}.$$

For the kernel $\Phi(\cdot,\cdot)$ and the l_2-regularization parameter $\tilde{\lambda}$, the regularized kernel LSTD solution $\hat{V}_n^{\tilde{\lambda}}$ on $\mathcal{D}_{train}$ can be represented in its vector form with the Mercer feature map $\boldsymbol{\omega} : \mathcal{S} \to \mathbb{R}^M$ as

$$\hat{V}_n^{\tilde{\lambda}}(\boldsymbol{s}) = \boldsymbol{\omega}(\boldsymbol{s})^T\bar{\boldsymbol{\theta}}_n(\tilde{\lambda}), \quad \forall \boldsymbol{s} \in \mathcal{S};$$

where $\boldsymbol{\omega}(\boldsymbol{s})$ is the feature vector representation of the state $\boldsymbol{s}$ in the Mercer feature space; and $\bar{\boldsymbol{\theta}}_n(\tilde{\lambda}) = \mathbf{\Pi}(\tilde{\lambda})\bar{\boldsymbol{\theta}}_n^* \in \mathbb{R}^M$ is the vector representation of $\hat{V}_n^{\tilde{\lambda}}$ in the Mercer feature space (equation 8.15), which depends on the vector representation $\bar{\boldsymbol{\theta}}_n^*$ of the continuous extension of the value function of the empirical MRP induced by $\mathcal{D}_{train}$ (equation 8.14) in the Mercer feature space.

As $N, m \to \infty$, with asymptotic constant ratio N/m, we have

$$\begin{aligned}
\widehat{\text{MSBE}}(\hat{\boldsymbol{\theta}}_n^{\lambda}) &\xrightarrow{a.s} \widehat{\text{MSBE}}\big(\hat{V}_n^{\tilde{\lambda}}\big) + \bar{\Delta}\big((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\mathbf{\Omega}_{\hat{\mathcal{S}}}^T\big),\\
\text{MSBE}(\hat{\boldsymbol{\theta}}_n^{\lambda}) &\xrightarrow{a.s} \text{MSBE}\big(\hat{V}_n^{\tilde{\lambda}}\big) + \bar{\Delta}\big(\boldsymbol{D}_{\mu^\pi}^{\frac{1}{2}}\left[\boldsymbol{I}_{|\mathcal{S}|} - \gamma\boldsymbol{P}^\pi\right]\mathbf{\Omega}_{\mathcal{S}}^T\big),\\
\text{MSVE}(\hat{\boldsymbol{\theta}}_n^{\lambda}) &\xrightarrow{a.s} \text{MSVE}\big(\hat{V}_n^{\tilde{\lambda}}\big) + \bar{\Delta}\big(\boldsymbol{D}_{\mu^\pi}^{\frac{1}{2}}\mathbf{\Omega}_{\mathcal{S}}^T\big);
\end{aligned}$$

where $\bar{\Delta}(\cdot)$ is the second-order function defined for any Mercer feature matrix $\boldsymbol{M} \in \mathbb{R}^{p\times M}$ of dimension $p > 0$, as

$$\bar{\Delta}(\boldsymbol{M}) = \tfrac{1}{n}\tfrac{\tilde{\lambda}}{\lambda}\frac{\frac{1}{N}\left\|\mathbf{\Pi}(\tilde{\lambda})\bar{\boldsymbol{\theta}}_n^*\right\|^2}{1-\frac{1}{N}\left\|\mathbf{\Pi}(\tilde{\lambda})\right\|_F^2}\left\|\boldsymbol{M}\left[\boldsymbol{I}_M - \mathbf{\Pi}(\tilde{\lambda})\right]\right\|_F^2.$$

The behavior of the implicit l_2-regularization parameter $\tilde{\lambda}$ defined in equation 8.16 with respect to the model complexity N/m and the l_2-regularization parameter is described by the following lemma.

Lemma 8.3.2. *The implicit l_2-regularization $\tilde{\lambda}$ defined in equation 8.16 satisfies the following properties:*

- *$\tilde{\lambda}$ is a decreasing function with respect to N/m,*

- $\tilde{\lambda} \to 0$ *as* $N/m \to \infty$,
- $\frac{\tilde{\lambda}}{\lambda} = \frac{1}{\frac{N}{m} - \frac{\mathrm{Tr}(\mathbf{\Pi}(\tilde{\lambda}))}{m}}$ *is a decreasing function with respect to* λ,
- $\frac{\tilde{\lambda}}{\lambda} \to \frac{m}{N}$ *as* $\lambda \to \infty$.

The following remarks attempt to provide an interpretation of the results of the above Lemma and Theorem.

Remark 29. *In the asymptotic N-limit, where $N/m \to \infty$, we find that $\tilde{\lambda} \to 0$ and $\bar{\Delta}\big((\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \mathbf{\Omega}_{\mathcal{S}}^T\big), \bar{\Delta}\big(\boldsymbol{D}_{\mu^\pi}^{\frac{1}{2}} \big[\boldsymbol{I}_{|\mathcal{S}|} - \gamma \boldsymbol{P}^\pi\big] \mathbf{\Omega}_{\mathcal{S}}^T\big), \bar{\Delta}\big(\boldsymbol{D}_{\mu^\pi}^{\frac{1}{2}} \mathbf{\Omega}_{\mathcal{S}}^T\big) \to 0$. In particular, we have*

$$\widehat{\mathrm{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda) \xrightarrow{a.s} \widehat{\mathrm{MSBE}}\big(V_n^{\mathcal{S}}\big) = 0,$$
$$\mathrm{MSBE}(\hat{\boldsymbol{\theta}}_n^\lambda) \xrightarrow{a.s} \mathrm{MSBE}\big(V_n^{\mathcal{S}}\big),$$
$$\mathrm{MSVE}(\hat{\boldsymbol{\theta}}_n^\lambda) \xrightarrow{a.s} \mathrm{MSVE}\big(V_n^{\mathcal{S}}\big),$$

where $V_n^{\mathcal{S}} : \mathcal{S} \to \mathbb{R}$ is the continuous extension of the value function $V_n : \hat{\mathcal{S}} \to \mathbb{R}$ in $\mathcal{S}$ (equation 8.14) of the empirical MRP induced by $\mathcal{D}_{train}$. We refind the well-known result that random feature models can be used as efficient parametric approximations of kernel methods in the asymptotic N-limit (Rahimi and Recht, 2007; Rudi and Rosasco, 2017). Note also that in the asymptotic N-limit, we perfectly interpolate the training data since $\widehat{\mathrm{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda) \xrightarrow{a.s} 0$.

Remark 30. *In the double asymptotic regime of Assumption 2, the model complexity N/m induces an implicit regularization $\tilde{\lambda}$, which prevents the model from perfectly interpolating the training data. The implicit regularization $\tilde{\lambda}$ depends on the model complexity N/m and is especially high in the under-parameterized regime for low model complexities N/m. As the model complexity N/m increases, the implicit regularization $\tilde{\lambda}$ decreases, and the random feature model better interpolates training data. A similar implicit regularization was observed in supervised learning for ridge regression with random features in double asymptotic regimes (Jacot et al., 2020a; Cheng and Montanari, 2022; Bach, 2024).*

Remark 31. *In the literature, the quantity $\mathrm{Tr}(\mathbf{\Pi}(\tilde{\lambda})) = \sum_{i=1}^m \frac{\nu_i(\hat{\boldsymbol{A}}_m \mathbf{\Phi}_{\hat{\mathcal{S}}})}{\nu_i(\hat{\boldsymbol{A}}_m \mathbf{\Phi}_{\hat{\mathcal{S}}}) + \tilde{\lambda}}$ is usually referred to as the number of degrees of freedom or the effective dimension of kernel methods (Hastie et al., 2009; Caponnetto and De Vito, 2007). Indeed, $\mathrm{Tr}(\mathbf{\Pi}(\tilde{\lambda}))$ describes how many dimensions are "effectively" used by the regularized kernel LSTD and defines its model complexity. The degrees of freedom $\mathrm{Tr}(\mathbf{\Pi}(\tilde{\lambda}))$ consumed by the regularized kernel LSTD estimator is monotone decreasing in $\tilde{\lambda}$. Note that $\mathrm{Tr}(\mathbf{\Pi}(\tilde{\lambda})) = m$ when $\tilde{\lambda} = 0$ (no regularization) and $\mathrm{Tr}(\mathbf{\Pi}(\tilde{\lambda})) \to 0$ as $\tilde{\lambda} \to \infty$. In our setting, since the implicit l_2-regularization parameter is a decreasing function with respect to the model complexity N/m, the number of degrees of freedom $\mathrm{Tr}(\mathbf{\Pi}(\tilde{\lambda}))$ consumed by the regularized kernel LSTD estimator is monotone decreasing in N/m. Consequently, models with low model complexities N/m induce higher constraints on the regularized kernel LSTD predictor, resulting in poorer performance. The oppositive behavior occurs for models with high model complexities N/m. Furthermore, note that $\mathrm{Tr}(\mathbf{\Pi}(\tilde{\lambda})) \leq \min(N, m)$ since $\frac{\tilde{\lambda}}{\lambda} \geq 0$. This shows that choosing a number of features N automatically lowers the effective dimension of the related kernel method. Finally, we can show that in the ridgeless under-parameterized regime (when $\lambda \to 0$), the effective dimension $\mathrm{Tr}(\mathbf{\Pi}(\tilde{\lambda})) \to N$.*

Remark 32. *In analogy with supervised learning (Hsu et al., 2012; Bach, 2024), the term $\big\|\mathbf{\Pi}(\tilde{\lambda})\big\|_F^2 = \mathrm{Tr}\big(\mathbf{\Pi}(\tilde{\lambda})^T \mathbf{\Pi}(\tilde{\lambda})\big) = \sum_{i=1}^M \sigma_i(\mathbf{\Pi}(\tilde{\lambda}))^2$, where $\sigma_i(\mathbf{\Pi}(\tilde{\lambda}))^2$ depicts the i-th singular value of $\mathbf{\Pi}(\tilde{\lambda})$, can*

be related to the second-degree of freedom of the regularized kernel LSTD. This quantity is also indicative of the number of dimensions or degrees of freedom that are "effectively" used by the regularized kernel LSTD.

Chapter 9

Numerical Experiments

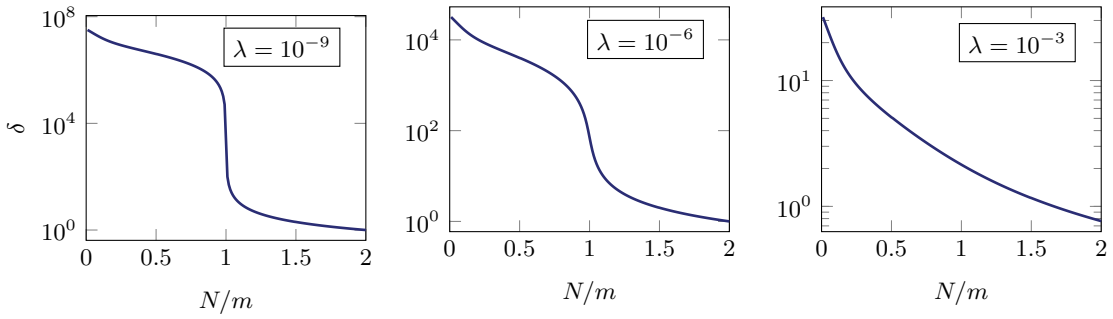
In this chapter, we present our experimental results and show our theory closely matches empirical results for regularized LSTD on a range of both toy and small real-world environments, where both the number of states visited m and the number of parameters N are fixed, but for which our asymptotic predictions still gives accurate predictions. In Section 9.1, we describe the experimental setup for our experiments. In Section 9.2, we discuss the behavior of the correction factor δ from Theorem 7.2.3 with respect to the model complexity N/m and the l_2-regularization parameter λ. In particular, we highlight a sharp decrease of the correction factor δ around $N/m = 1$ for small l_2-regularization parameters. We experimentally associate this sharp decrease with the double descent phenomenon observed for the MSBE and the MSVE in Section 9.3, which results in a peak in the MSBE and MSVE around $N/m = 1$. From our experiments, we also identify two distinct regimes: an *under-parameterized regime* where $N/m < 1$ and an *over-parameterized regime* where $N/m > 1$. Each regime exhibits different behaviors in the empirical MSBE, the true MSBE, and the MSVE. In Section 9.4, we empirically study the effect of the number of distinct unvisited states on the double descent phenomenon. In Section 9.5, we investigate the influence of the discount factor on the double descent phenomenon.

9.1 Experimental Setup

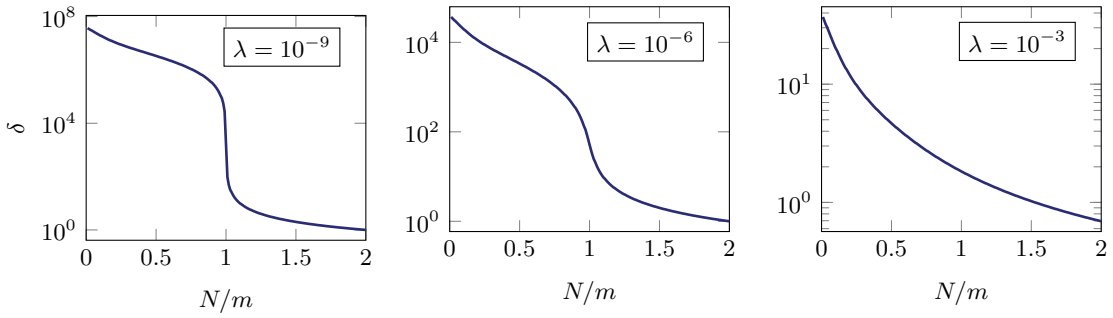
For computation, we use the recursive regularized LSTD (see Section 4.4) implementation of Dann et al. (2014) on three MRPs: a synthetic ergodic MRP (500 states); a gridworld MRP (400 states) obtained from a random policy in a 20×20 gridworld (Ahmed, 2018); and a Taxi-v3 MRP (356 states) obtained from a learned policy acting in the OpenAI gym Taxi-v3 environment (Towers et al., 2023). In all MRPs, states are described by d-Gaussian vectors where $d = 50$. For the random features of equation 6.5, $\boldsymbol{W}$ is drawn from a Gaussian distribution, and $\sigma(\cdot) = \max(0, \cdot)$ is the ReLU activation function. For all experiments, $\mathcal{D}_{\text{train}} := \{(\boldsymbol{s}_i, r_i, \boldsymbol{s}'_i)\}_{i=1}^n$ is derived from a sample path of n transitions with the same seed (42). For each instance i, we sample random features using the seed i. The following Figures in this section show averages over 30 instances.

9.2 Correction Factor δ

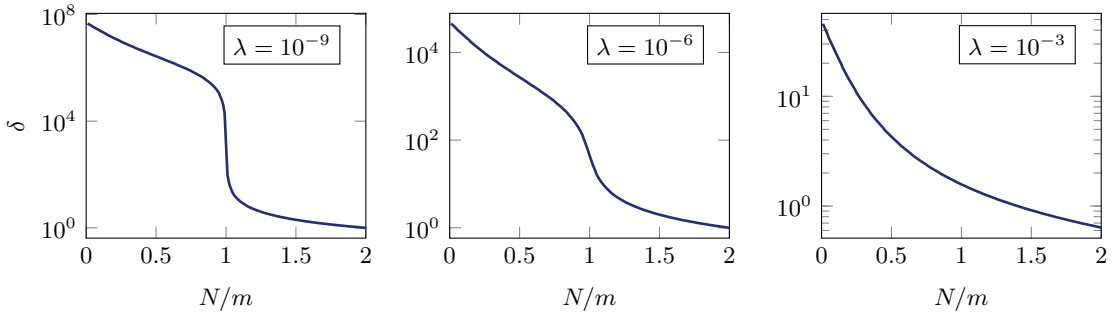
Correction Factor δ vs Model Complexity. The correction factor δ (equation 7.8) plays a key role in the asymptotic errors studied in Section 7 and 8. Figure 9.1 shows δ as a function of the model complexity N/m and for different values of the l_2-regularization parameter λ. It confirms that, as stated in Remark 17 and Lemma A.6.3, δ is a decreasing function with respect to the model complexity N/m. Furthermore, for a small λ, we observe a sharp decrease near $N/m = 1$. E.g., for $\lambda = 10^{-9}$, δ falls from an order of 10^7 when $N/m < 1$ to an order of 10^1 when $N/m > 1$. For larger values of λ, the correction factor δ decreases more smoothly and has smaller values.



(a) Synthetic Ergodic MRP



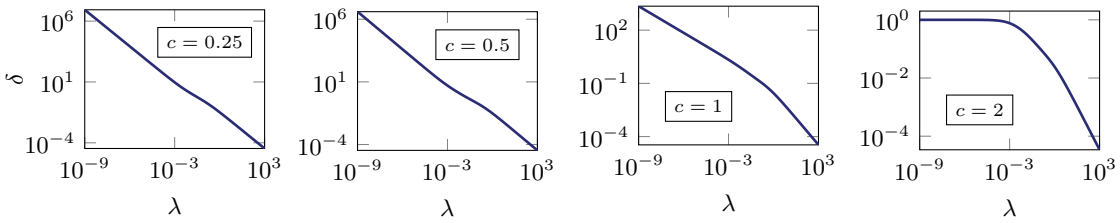
(b) Gridworld MRP



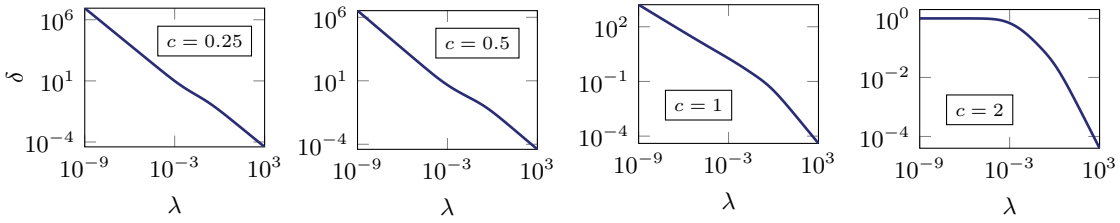
(c) Taxi-v3 MRP

Figure 9.1: **The correction factor δ is a decreasing function of the number of parameters N. For small l_2-regularization parameter λ, we observe a sharp decrease near $N/m = 1$, for m distinct visited states. As λ increases, the function becomes smoother and smaller (note the different scales of the y-axis).** δ is computed with equation 7.8 in synthetic ergodic, Girdworld and Taxi-v3 MRPs with $\gamma = 0.95, m = 499, n = 3000$, $\gamma = 0.95, m = 386, n = 5000$ and $\gamma = 0.95, m = 310, n = 5000$, respectively.

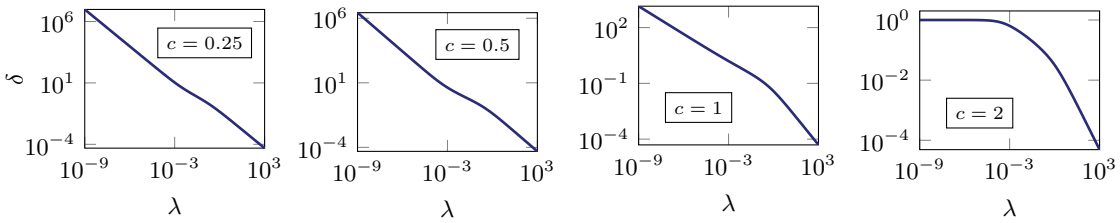
Correction Factor δ vs L_2-Regularization Parameter λ. Figure 9.2 depicts δ as a function of the l_2-regularization parameter for different model complexities N/m. It confirms that δ decreases monotonically as the l_2-regularization parameter λ increases, as shown in Remark 17 and Lemma A.6.4. As the model complexity N/m increases, we observe a larger initially flat region and smaller values of δ. Such behavior indicates that the impact of the l_2-regularization parameter λ becomes less significant as the model complexity N/m increases.



(a) Synthetic Ergodic MRP



(b) Gridworld MRP

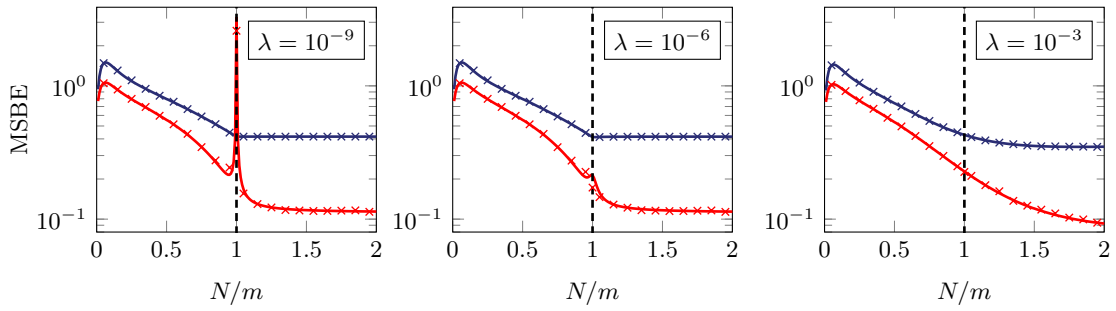


(c) Taxi-v3 MRP

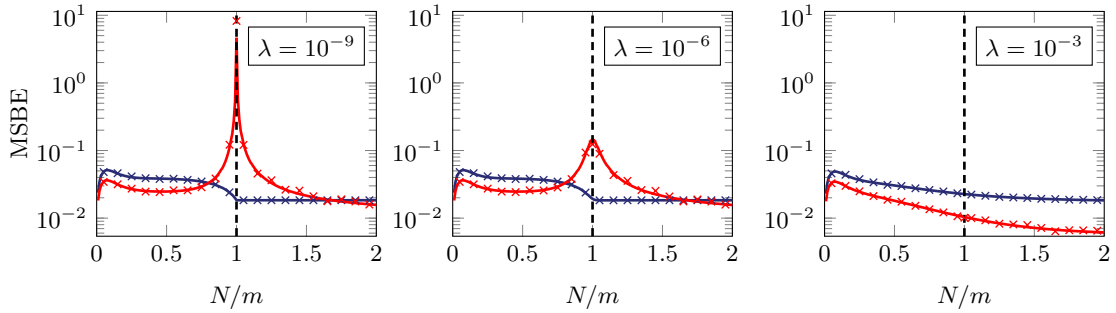
Figure 9.2: **The correction factor δ is a decreasing function of the l_2-regularization parameter λ. As the model complexity $c = N/m$ increases, the impact of regularization parameter λ becomes less significant (note the different scales of the y-axis).** δ is computed with equation 7.8 in synthetic ergodic, Girdworld and Taxi-v3 MRPs with $\gamma = 0.95, m = 499, n = 3000$, $\gamma = 0.95, m = 386, n = 5000$ and $\gamma = 0.95, m = 310, n = 5000$, respectively.

9.3 The Double Descent Phenomenon

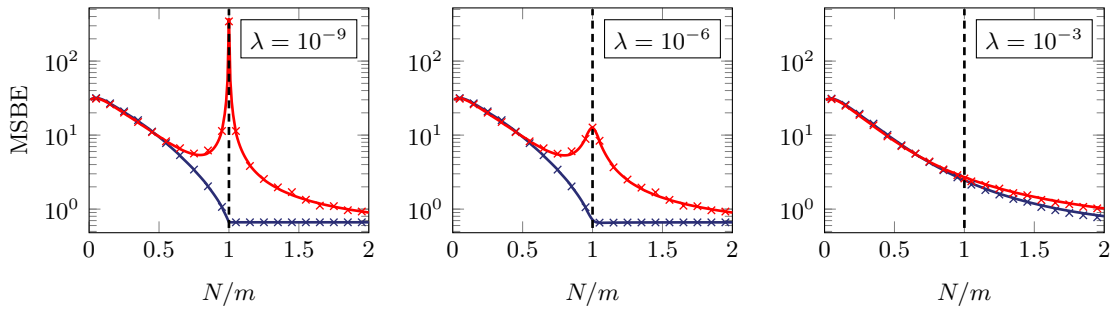
The Double Descent Phenomenon in the MSBE. As a consequence of the sharp transition of the correction factor δ for small l_2-regularization parameters λ depicted in Figure 9.1, Theorem 7.3.2 and Theorem 7.4.2 predict a change in behavior of the empirical $\widehat{\text{MSBE}}$ and true MSBE near $N/m = 1$. Figure 9.3 shows both $\widehat{\text{MSBE}}$ and MSBE as a function of the model complexity N/m with different l_2-regularization penalties λ. Although the equations for $\widehat{\text{MSBE}}$ and MSBE were derived for the asymptotic regime $N, m \to \infty$ defined in Assumption 2, we observe an almost perfect match with the numerically evaluated original definitions in equation 6.8 and equation 6.2. From the experiments, we can identify two distinct regimes: the *under-parameterized regime* where



(a) Synthetic Ergodic MRP



(b) Gridworld MRP

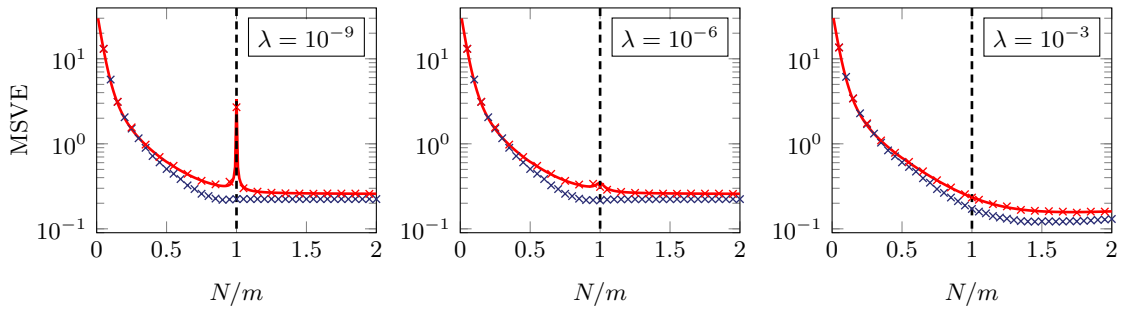


(c) Taxi-v3 MRP

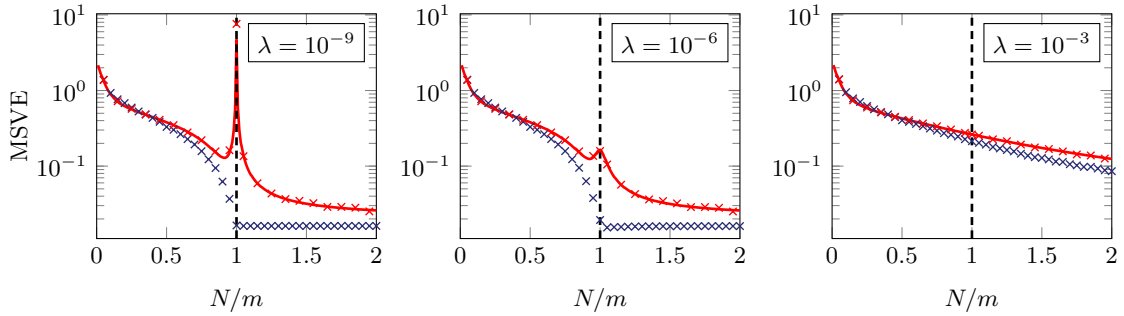
Figure 9.3: **The double descent phenomenon occurs in the true** MSBE **(red) of regularized LSTD, peaking around the interpolation threshold ($N/m = 1$ for N parameters, m distinct visited states) when the empirical** $\widehat{\text{MSBE}}$ **(blue) vanishes. It diminishes as the l_2-regularization parameter λ increases.** Continuous lines indicate the theoretical values from Theorem 7.3.2 and Theorem 7.4.2, the crosses are numerical results averaged over 30 instances after the learning with regularized LSTD in synthetic ergodic, Gridworld and Taxi-v3 MRPs with $\gamma = 0.95, m = 499, n = 3000$; $\gamma = 0.95, m = 386, n = 5000$; and $\gamma = 0.95, m = 310, n = 5000$, respectively.

$N/m < 1$ and the *over-parameterized regime* where $N/m > 1$. For small l_2-regularization parameters λ, the empirical $\widehat{\text{MSBE}}$ is close to its minimum at the *interpolation threshold*. At the interpolation threshold $N/m = 1$, the predictor almost perfectly interpolates the training data. For over-parameterized models with $N/m \geq 1$, the empirical $\widehat{\text{MSBE}}$ decreases more slowly with respect to N/m and remains almost constant. In contrast, for small λ, the true MSBE exhibits a peak around the interpolation threshold ($N/m = 1$), leading to a double descent phenomenon. In the under-parameterized regime ($N/m < 1$), the MSBE exhibits the classic U-shaped curve. Meanwhile, in the over-parameterized regime ($N/m > 1$), the MSBE decreases with respect to the model complexity N/m. While for the Taxi-v3 MRP, the empirical $\widehat{\text{MSBE}}$ is smaller than the true MSBE, this is not necessarily the case in other environments, where the empirical $\widehat{\text{MSBE}}$ can be larger overall than the true MSBE. For larger λ, the double descent in the true MSBE disappears, and the difference between the true MSBE and the empirical $\widehat{\text{MSBE}}$ is less pronounced, although it may not vanish. Note that the minimum error is achieved in all experiments in the over-parameterized regime. All the above observations are in accordance with established results in the supervised learning literature (Liao et al., 2020).

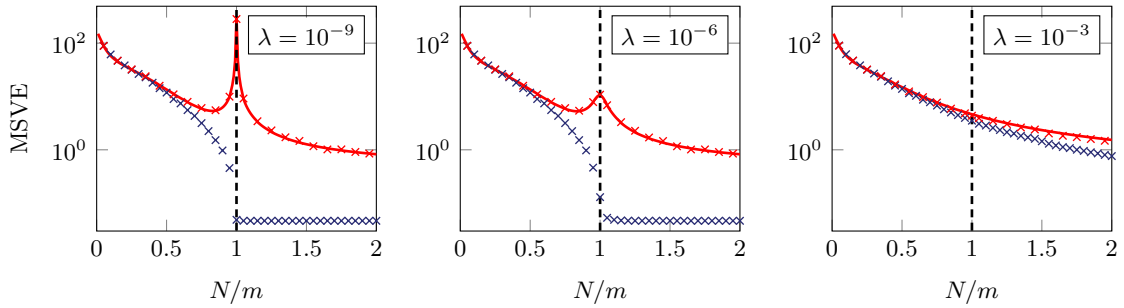
The Double Descent Phenomenon in the MSVE. Figure 9.4 shows both the empirical $\widehat{\text{MSVE}}$ and the true MSVE as a function of the model complexity N/m for different l_2-regularization penalties λ in the synthetic ergodic, Girdworld and Taxi MRPs. We observe an almost perfect match with the numerically evaluated original definition in equation 7.16. Similarly to the true MSBE, for small l_2-regularization penalties λ, the MSVE peaks around the interpolation threshold $N/m = 1$, leading to a double descent phenomenon. In contrast, the empirical $\widehat{\text{MSVE}}$ is close to its minimum at $N/m = 1$ and almost constant for $N/m \geq 1$, with no double descent observed. For larger λ, the double descent in the true MSVE disappears, and the difference between the true MSVE and the empirical $\widehat{\text{MSVE}}$ is less pronounced, although it may not vanish. Unlike the true MSBE, we observe that the empirical $\widehat{\text{MSVE}}$ is constantly smaller than the true MSVE.



(a) Toy Ergodic MRP



(b) Gridworld

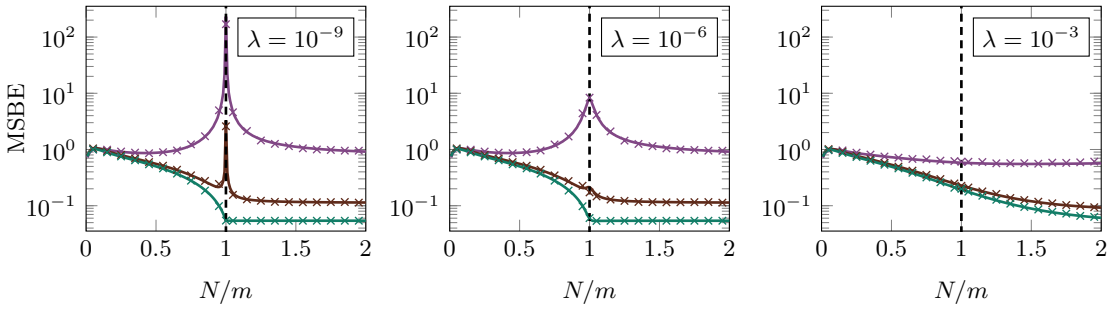


(c) Taxi-v3

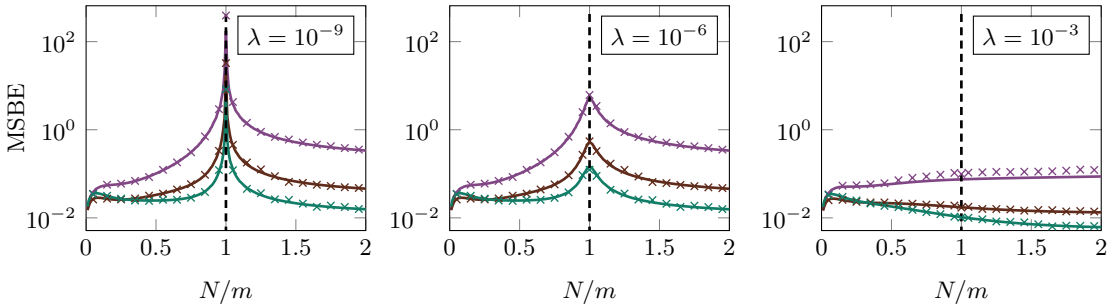
Figure 9.4: **The double descent phenomenon occurs in the true** MSVE **(red) of regularized LSTD, peaking around the interpolation threshold** ($N/m = 1$ **for** N **parameters,** m **distinct visited states) when the empirical** $\widehat{\text{MSVE}}$ **(blue) vanishes. It diminishes as the** l_2**-regularization parameter** λ **increases.** Continuous lines indicate the theoretical values from Corollary 7.5.0.1, the crosses are numerical results averaged over 30 instances after the learning with regularized LSTD in synthetic ergodic, Gridworld and Taxi-v3 MRPs for $\gamma = 0.95, m = 499, n = 3000$; $\gamma = 0.95, m = 386, n = 5000$; and $\gamma = 0.95, m = 310, n = 5000$, respectively.

9.4 Influence of the Number of Unvisited States

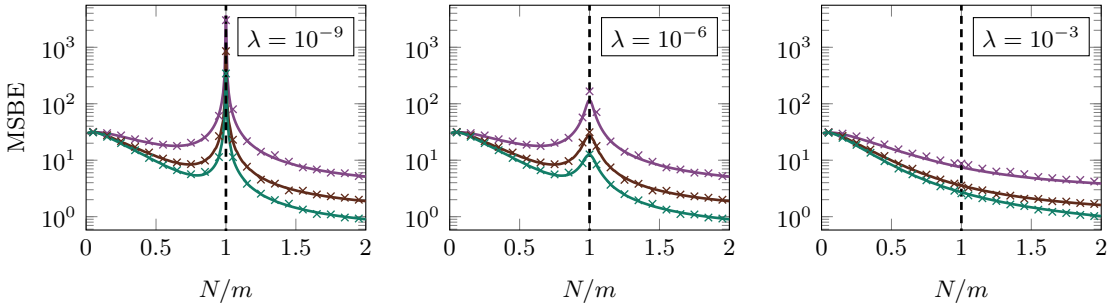
Once all states have been visited, MSBE and $\widehat{\text{MSBE}}$ exhibit a similar behavior (see Remark 25), with no peak at the interpolation threshold ($N/m = 1$) and no double descent phenomenon. The experiments in Figure 9.5 depict this behavior. They also illustrate that the double descent phenomenon diminishes as the number of distinct unvisited states goes to zero, yet it remains visible. In particular, in experiments on the synthetic ergodic MRP, we observe that the double descent phenomenon remains evident for small l_2-regularization parameters λ, even when only one state remains unvisited (maroon curve).



(a) Synthetic Ergodic MRP for $m = 0.86|\mathcal{S}|$ (purple), $m = 0.998|\mathcal{S}|$ (maroon), $m = |\mathcal{S}|$ (green).



(b) Gridworld MRP for $m = 0.59|\mathcal{S}|$ (purple), $m = 0.92|\mathcal{S}|$ (maroon), $m = 0.97|\mathcal{S}|$ (green).

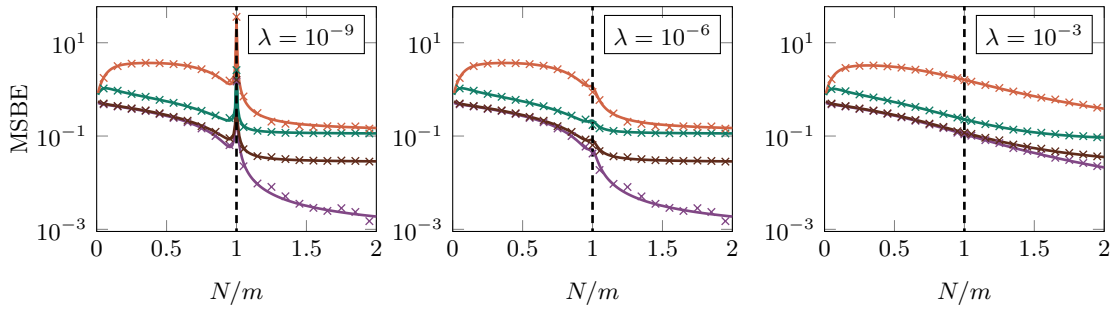


(c) Taxi-v3 MRP for $m = 0.57|\mathcal{S}|$ (purple), $m = 0.79|\mathcal{S}|$ (maroon), $m = 0.87|\mathcal{S}|$ (green).

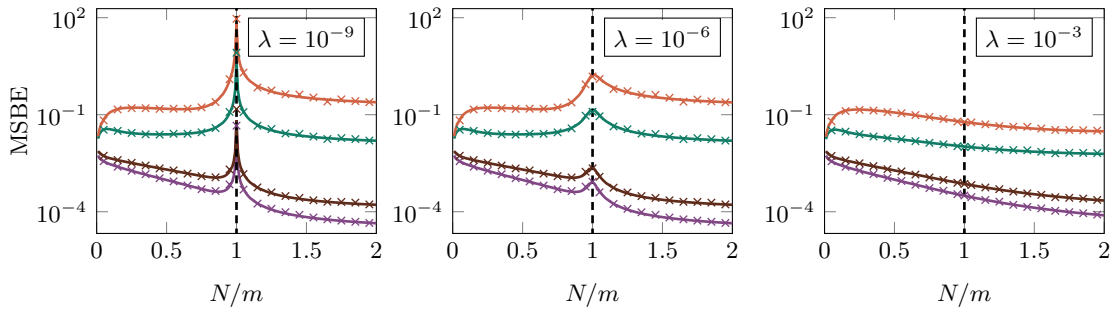
Figure 9.5: **With more distinct states m visited, the double descent in the MSBE diminishes, disappearing for $m = |\mathcal{S}|$.** Continuous lines indicate the theoretical values of MSBE from Theorem 7.4.2 for different numbers of distinct visited states m; the crosses are numerical results averaged over 30 instances after the learning with regularized LSTD in synthetic ergodic, Gridworld and Taxi-v3 MRPs with $\gamma = 0.95, d = 50$.

9.5 Influence of the Discount Factor

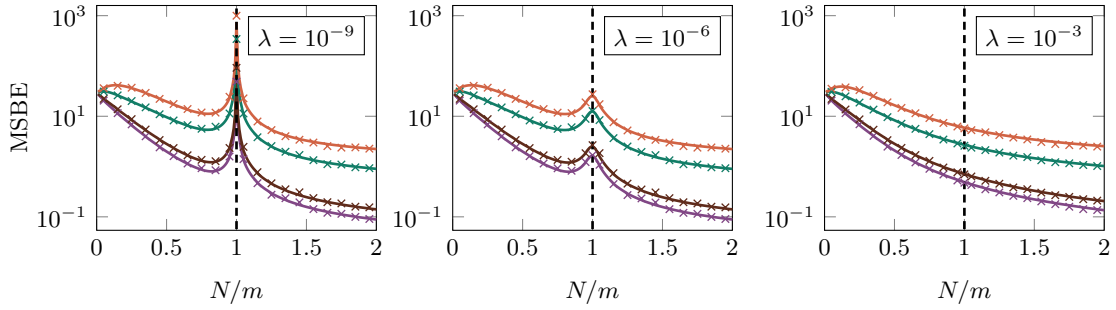
The experiments in Figure 9.6 illustrate that the discount factor γ has little impact on the double descent phenomenon. As the discount factor increases, we observe an increase of the MSBE since learning becomes more difficult. Note that the curves (pruple) for $\gamma = 0$ are also depicted, indicating situations where the solution of the regularized LSTD is equivalent to the solution of ridge regression on the reward function $R^\pi : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$.



(a) Synthetic Ergodic MRP



(b) Gridworld MRP



(c) Taxi-v3 MRP

Figure 9.6: **The discount factor γ has little effect on the double descent in the MSBE.** Continuous lines indicate the theoretical values of MSBE from Theorem 7.4.2 for $\gamma = 0$ (purple), $\gamma = 0.5$ (maroon), $\gamma = 0.95$ (green), and $\gamma = 0.99$ (orange); the crosses are numerical results averaged over 30 instances after the learning with regularized LSTD in synthetic ergodic, Gridworld and Taxi-v3 MRPs for $\gamma = 0.95, m = 499, n = 3000$; $\gamma = 0.95, m = 386, n = 5000$; and $\gamma = 0.95, m = 310, n = 5000$, respectively

Part III

Features Encoding in Deep Reinforcement Learning

In classic Reinforcement Learning (RL), encoding the inputs with a Fourier feature mapping is a standard way to facilitate generalization and add prior domain knowledge. In Deep RL, such input encodings are less common since they could, in principle, be learned by the network and may therefore seem less beneficial. In this part, we present experiments on Multilayer Perceptrons (MLP) that indicate that even in Deep RL, Fourier features can lead to significant performance gains in both rewards and sample efficiency. Furthermore, we observe that they increase the robustness with respect to hyperparameters, lead to smoother policies, and benefit the training process by reducing learning interference, encouraging sparsity, and increasing the expressiveness of the learned features. However, a major bottleneck with conventional Fourier features is that the number of features increases exponentially with the state dimension. As a remedy, we propose a simple, light version that only has a linear number of features yet empirically provides similar benefits. Our experiments cover both shallow/deep, discrete/continuous, and on/off-policy RL settings.

Part III is organized as follows:

- In Chapter 10, we start by presenting features encoding in linear function approximation and the use of neural networks in deep RL to automatically learn features from raw data without prior knowledge. As highlighted in this chapter, although neural networks are universal approximators in theory, they suffer from some limitations in practice. In particular, we present experiments that indicate neural networks behave as under-parameterized models regularized through early stopping.
- In Chapter 11, to overcome the spectral bias and improve the learning of high-frequency components in RL, we suggest the use of two preprocessings based on the Fourier series for neural networks. The first preprocessing suggested is the Fourier Feature (FF) mapping, based on the Fourier series and introduced by Konidaris et al. (2011) for linear value function approximation. For the second preprocessing, we propose a lighter, scalable version of the FF preprocessing called Fourier Light Features (FLF) in which the dimension of the feature space grows linearly with the dimension of the state space. In the following of this chapter, we present experiments indicating that the use of FF/FLF can lead to significant performance gains in terms of rewards and sample efficiency, and outperform other traditional preprocessings. We observe that both FLF and FF achieve similar performance, while FLF has fewer features than FF. Furthermore, we observe that such preprocessings increase the robustness with respect to hyperparameters.
- In Chapter 12, we empirically investigate the effects of the Fourier encodings on the learning process. In particular, we show that the proposed preprocessings lead to smoother neural networks, mitigate learning interference, promote sparsity, and increase the expressivity of learned features.

This part is mainly based on our work *Fourier Features in Reinforcement Learning with Neural Networks*, with David Filliat and Goran Frehse, accepted for publication in the *Transactions on Machine Learning Research (TMLR)*, 2024.

Chapter 10

Features Encoding

In this chapter, we start by presenting in Section 10.1 features encoding in linear function approximation and the use of neural networks in deep RL to automatically learn features from raw data without prior knowledge. Although neural networks are universal approximators in theory, they suffer from some limitations in practice. These limitations include not only the number of parameters, as discussed in the last part, but also the amount of optimization that can be achieved in practice. In Section 10.2, we present experiments that indicate neural networks behave as underparameterized models regularized through early stopping. In particular, we observe that this form of regularization induces a *spectral bias*, in which the fitting high-frequency components of the value function requires exponentially more gradient update steps than the low-frequency ones. In Section 10.3, we propose to mitigate the spectal bias and improve the learning of high-frequency components by using feature encodings as preprocessing as shown in Figure 10.4.

10.1 Features Encoding in Linear Function Approximations

In linear function approximation, the performance of linear parameterized models mainly depends on how the data are represented in the feature space. As discussed in Section 3.3, the accuracy of predictions largely depends on the space of functions that linear function approximations can represent. A huge amount of practical and theoretical work has been dedicated to understanding feature selection and generation for linear value function approximation (Parr et al., 2007; 2008; Song et al., 2016; Ghosh and Bellemare, 2020). Choosing appropriate features for a task is a critical way of adding prior domain knowledge. The representation is hand-designed according to the task and projected into a higher-dimensional space to facilitate a linear separation (Sutton and Barto, 2018). However, determining what features to use remains a complex challenge as they depend on the problem being solved. In RL, a large amount of works proposed feature encodings for linear function approximation, e.g., Polynomial Features (Lagoudakis and Parr, 2003), Tile Coding (Albus, 1971), Krylov basis (Petrik, 2007) or Fourier Features (Konidaris et al., 2011). However, one main bottleneck of such feature encodings is that they do not scale well to high-dimensional inputs, as their size grows exponentially with the input dimension.

In recent years, artificial *neural networks* have led to breakthroughs due to their ability to learn

features from raw data without prior knowledge. In particular, the layer preceding the linear output layer called the *penultimate layer* can be interpreted as a learnable feature extractor or encoder in the linear function approximation framework, as depicted in Figure 10.4. Neural networks allow machine learning algorithms to learn feature representations specific to tasks, using raw sensory data and without prior knowledge (Mnih et al., 2015; Schulman et al., 2017; Lillicrap et al., 2015; Haarnoja et al., 2018).

10.2 Limitations of Neural Networks in Deep RL

Although neural networks may be very useful and powerful for learning feature representations from raw data without prior knowledge, they suffer from some limitations in practice. A recent study by Dong et al. (2020) suggests that neural networks have limitations when predicting value functions in RL. Through theoretical analysis and empirical evidence, the authors highlighted that MDPs with simple dynamics can have very complex optimal action-value function $Q^* : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and optimal policies $\pi^* : \mathcal{S} \to \mathcal{A}$ characterized by high-frequency variations. Complexities of the optimal action-value function $Q^* : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ arise from to the recursive application of the Bellman optimality operator and the nature of unrolling of the dynamics. In this section, to highlight the limitations of neural networks in RL, we propose to experimentally study the learning of optimal value functions on the toy MDP distribution $P_{\mathcal{M}}$ introduced by Dong et al. (2020). For each MDP $\mathcal{M} \sim P_{\mathcal{M}}$, the MDP $\mathcal{M}$ is defined by a tuple $(\mathcal{S}, \mathcal{A}, P, R, \mu_0)$ for which we have

- the state space $\mathcal{S} = [0, 1)$;
- the discrete action space $\mathcal{A} = \{0, 1\}$;
- the deterministic dynamics of the MDP $P : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, where $P(\cdot, 0), P(\cdot, 1)$ are randomly sampled from the space of piece-wise linear functions with a fixed number k of "kinks";
- the reward function $R : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ defined for all $s, s' \in \mathcal{S}$ as $R(s, s') = s$;
- and the initial state ditribution μ_0 defined as the uniform distribution on $\mathcal{S}$.

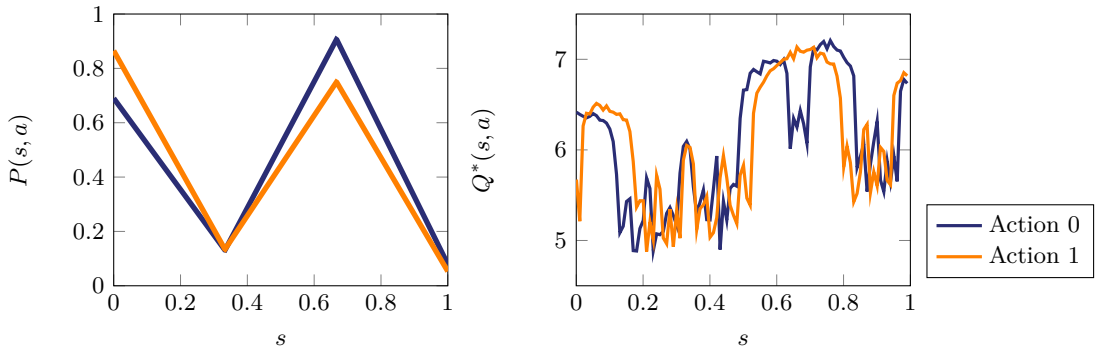


Figure 10.1: Example of $\mathcal{M} \sim P_{\mathcal{M}}$ drawn from the MDP distribution of Dong et al. (2020) for a number of "kinks" $k = 2$.

Figure 10.1 depicts the dynamics and the optimal Q-function of a toy MDP $\mathcal{M}$ drawn from the distribution $P_{\mathcal{M}}$, for a number of "kinks" $k = 2$. Even though the dynamics of the MDP $\mathcal{M}$

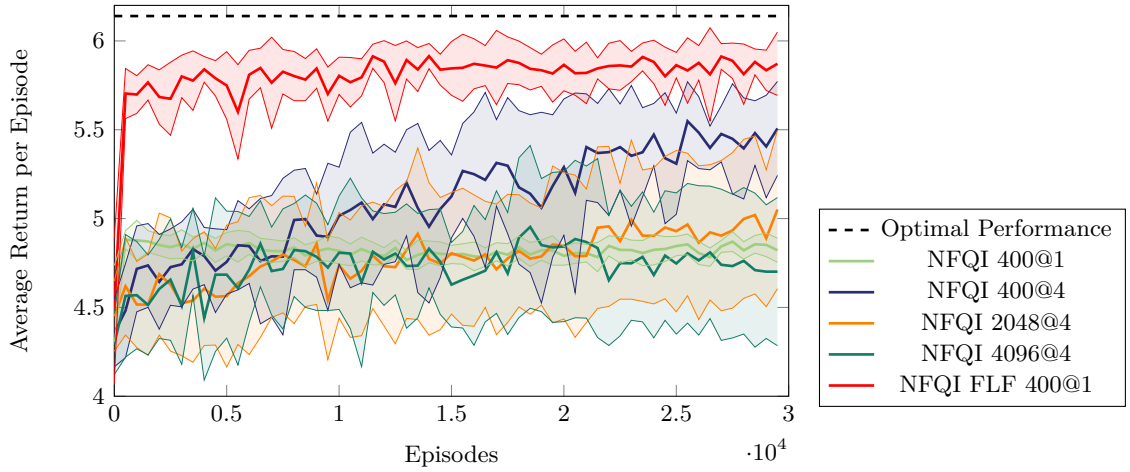


Figure 10.2: **MDP with simple dynamics may have complex optimal Q-function. MLPs without function expansion underperform on MDPs.** Evaluation curves of different MLP architectures on the toy MDP described in Figure 10.1. Curves are averaged over 10 training runs. Shading indicates the 95% confidence interval (CI). The tested architectures are a 1-layer MLP with 400 hidden neurons and 4-layer MLPs with 400, 2048, and 4096 hidden neurons.

are piece-wise linear functions, the optimal action-value function $Q^* : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is complex with high-frequency components. Figure 10.3 depicts predictions of action-value functions from different Multilayer Perceptrons (MLP) architectures trained with the Neural Fitted Q-Iteration (FQI) algorithm (Riedmiller, 2005) on the toy MDP $\mathcal{M}$ depicted in Figure 10.1. All predictors (including those with a large architecture) underfit the optimal value function and fail to capture the high-frequency components after learning. These observations are consistent with experimental results in supervised learning, where Rahaman et al. (2019) highlighted a learning bias of deep networks towards low-frequency functions, i.e., functions that vary globally without local fluctuations. Considerable effort has been made in supervised learning to provide a theoretical explanation of this spectral bias (Bietti and Mairal, 2019; Bordelon et al., 2020; Cao et al., 2021; Xu et al., 2022; Canatar et al., 2024). In RL, the spectal bias was also experimentally observed and studied in value function approximation by Yang et al. (2021). However, the phenomenon seems more complex and depends on the dynamics of the MDP (Lyle et al., 2021). Such spectral bias may prevent MLPs from accurately learning high-frequency components of complex value functions with high-frequency components, resulting in poor performance as depicted by Figure 10.2. This phenomenon is further exacerbated by the fact that neural networks in TD learning algorithms tend to generalize poorly and memorize experiences during the training (Lyle et al., 2021; 2022; Nikishin et al., 2022).

10.3 Features Encoding with Neural Networks & Contributions

In this part, we propose adding a features encoding block to the MLP architecture to enhance the learning of high-frequency components. In Deep Learning, it is common to apply min-max normalization (Bishop et al., 1995) or batch normalization (Ioffe and Szegedy, 2015) on data. However, preprocessing inputs with hand-designed features are less common since such features could, in principle, be learned by the network and thus may seem less beneficial. In recent work,

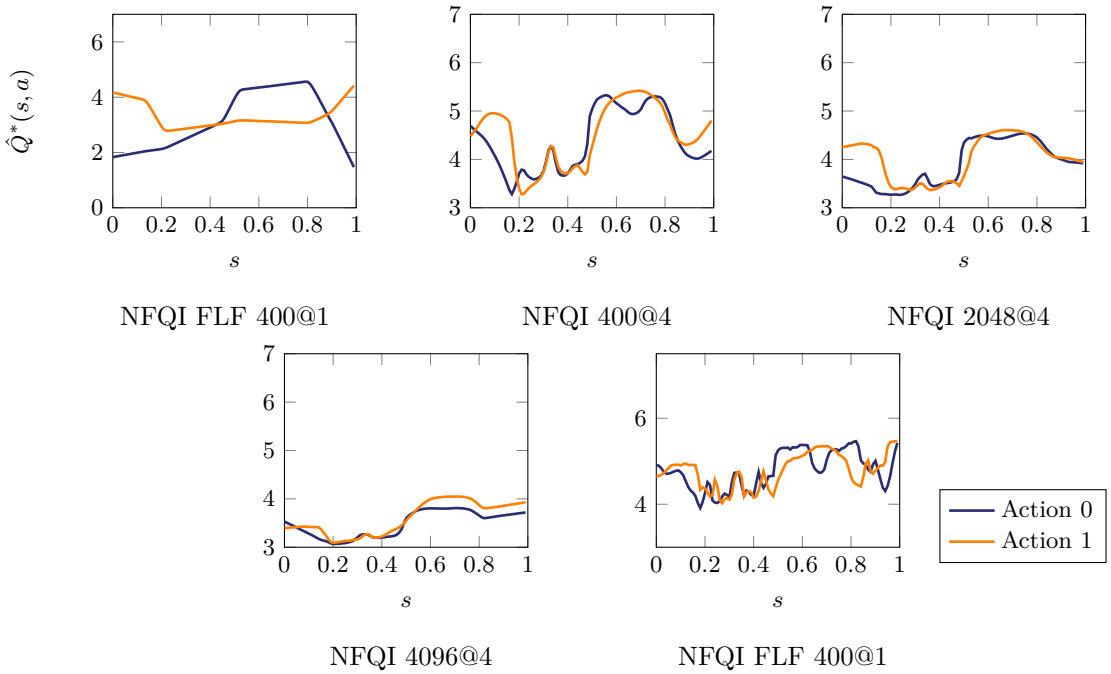


Figure 10.3: **MLPs without features encoding underfit the optimal Q-value function of the toy MDP $\mathcal{M}$ described in Figure 10.1**. Predictions of MLPs trained with the neural Fitted Q-Iteration are averaged over 10 training runs. The tested architectures are a 1-layer MLP with 400 hidden neurons and 4-layer MLPs with 400, 2048, and 4096 hidden neurons.

it has been shown that preprocessing inputs with Random Fourier Features (Rahimi and Recht, 2007) help *Multilayer Perceptrons* (MLP) to control the frequencies that the network tends to learn first (Tancik et al., 2020; Wang et al., 2021) and improves the training performance for neural networks (Mehrkanoon and Suykens, 2018; Mitra and Kaddoum, 2021). In Deep RL, it has been observed that Tile Coding can improve performance, sample efficiency, and robustness to hyperparameter variations by mitigating learning interference (Ghiassian and Huizhen Yu, 2018; Ghiassian et al., 2020; Liu et al., 2019b). In the following of this part, we empirically study preprocessing inputs with a functional expansion based on the Fourier series for MLPs in Deep RL, as illustrated in Figure 10.4. The study is based on kinematic observation-based benchmarks, where observations are expressed as state vectors whose components are the agent's kinematic quantities. Although the advantages of Fourier Features have been investigated in the case of standard RL Konidaris et al. (2011), to the best of our knowledge, their use was not yet studied in Deep RL at the time of the study. Concurrent works propose tuning the scale of learnable Fourier features to ensure that high-frequency components of the value function are captured (Li and Pathak, 2021; Yang et al., 2021). Another recent work studied the use of the Fourier series with MLP in computer vision (Benbarka et al., 2022).

Our main contributions in this part can be summarized as follows:

- While Fourier Features are standard in classic Reinforcement Learning, we suggest that Fourier Features are beneficial in kinematic observation-based RL problems with neural networks. We observe significant performance gains in both rewards and sample efficiency and extend the range of usable hyperparameters. In our experiments, Fourier Features outperform other common types of input preprocessing.

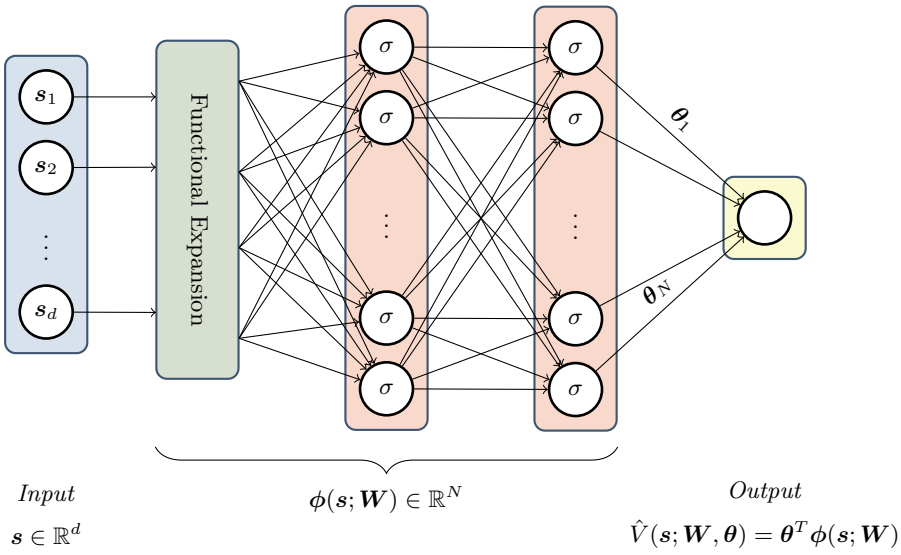


Figure 10.4: **Example of a 2-layers MLP with features encoding for value-based algorithms**. The state $\boldsymbol{s} \in \mathbb{R}^d$ is processed with a functional expansion (e.g, Fourier features) before being passed into the MLP. For a given state $\boldsymbol{s} \in \mathcal{S}$, features returned by the penultimate layer of the MLP are denoted by $\boldsymbol{\phi}(\boldsymbol{s}; \boldsymbol{W})$, where $\boldsymbol{W}$ depicts the weights of the MLP excluding those of the output layer. Output of the neural network $\hat{V}(\boldsymbol{s}; \boldsymbol{W}, \boldsymbol{\theta})$ is a linear function $\hat{V}(\boldsymbol{s}; \boldsymbol{W}, \boldsymbol{\theta}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{s}; \boldsymbol{W})$, where $\boldsymbol{\theta} \in \mathbb{R}^N$ denotes the weights of the last layer.

- We empirically investigate the effects of Fourier features on the learning process and show that Fourier features lead to smoother neural networks, mitigate learning interference, promote sparsity, and increase the expressivity of learned features.
- We propose a light, scalable version of Fourier Features to avoid the exponential explosion of traditional Fourier Features while maintaining much of their benefits.

Chapter 11

Features Encodings Based on Fourier Series

In this chapter, to overcome the spectral bias phenomenon described in Section 10.2 and improve the learning of high-frequency components in RL, we suggest the use of two preprocessings based on the Fourier series for neural networks as depicted by Figure 10.4. In Section 11.1, we propose as preprocessing the Fourier Feature (FF) mapping, based on the Fourier series and introduced by Konidaris et al. (2011) for linear value function approximation. However, the major bottleneck of this Fourier preprocessing is that the dimension of the feature space grows exponentially with the dimension of the state space, which limits its use in high-dimensional problems. To remedy this issue, we propose in Section 11.1.1 a lighter, scalable version of the FF preprocessing called Fourier Light Features (FLF). In Section 11.2, we present experiments indicating that the use of FF/FLF can lead to significant performance gains in terms of rewards and sample efficiency, and outperform other traditional preprocessings.

11.1 Fourier Features

Konidaris et al. (2011) introduced Fourier Features (FF) in RL in linear value function approximation by using the terms of the multivariate Fourier series as features. In practice, Fourier features are easy to use and perform better for linear function approximation than other popular feature encodings, such as Tile Coding or Polynomial Basis (Konidaris et al., 2011). Formally, they are generated by the *order-m Fourier Feature* function expansion $\text{FF} : [0,1]^d \to \mathbb{R}^p$, mapping a normalized state $\boldsymbol{s} \in \mathcal{S} \subseteq [0,1]^d$ into a p-dimensional feature space (Konidaris et al., 2011), with $p = (m+1)^d$. For $i \in [p]$, the feature i is given by

$$\text{FF}_i(\boldsymbol{s}) = \cos(\pi \boldsymbol{s}^T \boldsymbol{c}^i), \tag{11.1}$$

where each coefficient vector $\boldsymbol{c}^i$ takes a value in $\{0, \dots, m\}^d$ (one-to-one). Examples of Fourier Features are provided in Figure 11.1.

Remark 33. *The inner product $\boldsymbol{s}^\top \boldsymbol{c}^i$ in equation 11.1 determines the frequency along dimension i and creates interactions between state variables.*

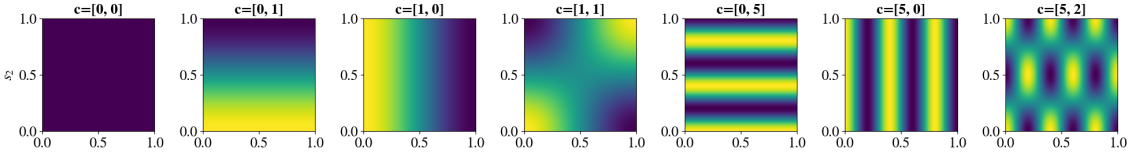


Figure 11.1: Example of Fourier Features over 2 variables ($d = 2$). Darker colors indicate a value closer to 1, and lighter colors indicate a value closer to -1. Note that $c = [0, 0]$ results in a constant function. When $c = [0, k_y]$ or $[k_x, 0]$ for positive integers k_x and k_y, the basis function depends on only one of the variables, with the value of the non-zero component determining frequency. Only when $c = [k_x, k_y]$ does it depend on both; this basis function represents an interaction between the two state variables. The ratio between k_x and k_y describes the direction of the interaction, while their values determine the basis function's frequency along each dimension.

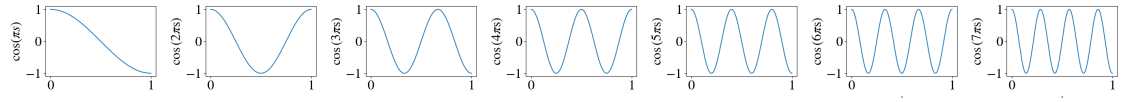


Figure 11.2: Example of Fourier Light Features for $d = 1$.

Remark 34. *Fourier series approximate periodic functions with a linear combination of cosine and sine functions. However, value functions are generally not periodic. The trick used in equation 11.1 to use Fourier series in non-periodic functions is normalizing the inputs. By normalizing the state space into* $[-1, 1]^d$*, we can use a linear combination of the sine and cosine functions to approximate the value function, even if the value function is aperiodic. Indeed, in such a case, the value function under study can be interpreted as the single period (with period* $\mathbf{1} = [1, \ldots, 1]^T$*) of a periodic function. To drop the sine terms and obtain equation 11.1, the value function under study can be interpreted as the single half-period (with period* $\mathbf{1} = [1, \ldots, 1]^T$*) of a symmetric periodic function, by normalizing the state space into* $[0, 1]^d$*.*

The major bottleneck of Fourier features is that their number p grows exponentially with the dimension d of the state space $\mathcal{S}$ as $p = (m+1)^d$. To remedy this, we propose the following subset of Fourier features that do not join state variables during preprocessing and scale linearly with the state dimension d.

11.1.1 Fourier Light Features

We define *order-m Fourier Light Features* (FLF) as the $d(m + 1)$ Fourier Features generated by the *order-m Fourier Light Feature* functional expansion $\text{FLF} : [0, 1]^d \to \mathbb{R}^{d(m+1)}$, which maps a normalized state $\boldsymbol{s} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_d]^T \in [0, 1]^d$ into a $d(m + 1)$-dimensional feature space as follows

$$\text{FLF}(\boldsymbol{s}) = \Big[\text{FF}(\boldsymbol{s}_1) \mid \ldots \mid \text{FF}(\boldsymbol{s}_d)\Big]^T, \tag{11.2}$$

with $\text{FF} : \mathbb{R} \to [0, 1]^{m+1}$ defined in equation 11.1 as $\text{FF}(\boldsymbol{s}_i) = \big[1, \cos(\pi \boldsymbol{s}_i), \ldots, \cos(m\pi \boldsymbol{s}_i)\big]$, $\forall i \in [1, d]$. The choice not to mix state variables in this version of Fourier features is motivated by the fact that Fourier features are not directly used for making predictions but rather serve as a preprocessing for the data injected into the neural network. We let neural networks choose how state variables will be mixed while learning the features used for predictions. Examples of order-7 FLF for $d = 1$ are depicted in Figure 11.2.

11.2 Empirical Performance

In this section, we present the empirical performance of the Fourier features encoding described in Section 11.1 and the Fourier light features encoding introduced in Section 11.1.1. In particular, we apply these encodings on the off-policy Deep-Q Network (DQN) algorithm (Mnih et al., 2015) in discrete action environments and on the on-policy Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017) in continuous action environments. Both types of environments provide kinematic observations, which are expressed as a state vector whose components are the agent's kinematic quantities. In Section 11.2.1, we detail the experimental setup used for experiments. In Section 11.2.2, we present the empirical performance of neural networks equipped with Fourier preprocessings. In Section 11.2.3, we study the influence of these preprocessings on hyperparameters of RL algorithms. In Section 11.2.4, we compare the performance of neural networks using Fourier encodings with those usign other popular features encodings, such as Tile Coding.

11.2.1 Experimental Setup

In our experiments, all observations are kinematic observations, expressed as a state vector whose components are the agent's kinematic quantities. We used the DQN and PPO implementations provided by StableBaselines-3 (Raffin et al., 2019) (version 0.10.0) and Pytorch 1.8.0. In all experiments, optimization was performed using the Adam optimizer (Kingma and Ba, 2014), parameters were initialized using the Xavier initializer (Glorot and Bengio, 2010), and the ReLU function was used as the activation function.

Experiments on Discrete Environments. Experiments with DQN are performed on five discrete-action environments provided by OpenAI Gym (Brockman et al., 2016): Acrobot-v1, CartPole-v1, LunarLander-v2, MountainCar-v0, and Catcher-v1 (Tasfi, 2016). Since hyperparameters for the discrete control tasks were not included in Stable Baselines Zoo (Raffin, 2020) at the time of experiments, we tuned the DQN hyperparameters for each task with Optuna 2.4.0 (Akiba et al., 2019). In particular, we used the Tree-structured Parzen Estimator (TPE) algorithm (Bergstra et al., 2011) to sample hyperparameters from the ranges provided in Table 11.1. Given that TPE is very sensitive to the scores of the first trials, we ran 5 independent hyperparameter research, each consisting of 500 trials. Each trial corresponds to a training of $120,000$ timesteps with hyperparameters sampled from TPE, and its score is evaluated on the return of 100 rollouts of the learned policy. Due to the unreliability of Deep RL algorithms (Henderson et al., 2018; Islam et al., 2017), we selected the 15 best hyperparameter configurations found by Optuna and ran 5 additional trainings of $150,000$ timesteps for each. The optimal hyperparameter setting was selected based on the highest average final return across these trainings.

Experiments on Continuous Environments. Experiments with PPO are performed on five continuous-action control tasks provided by Mujoco (Todorov et al., 2012): HalfCheetah-v2, Hopper-v2, InvertedDoublePendulum-v2, Swimmer-v2, and Walker-2d-v2. PPO hyperparameters are taken from StableBaselines Zoo (Raffin, 2020).

Experiments with Fourier Features Encodings. Fourier Features (FF) and Fourier Light Features (FLF) encodings take normalized states as inputs. Before being passed into these feature

Table 11.1: Range of DQN Hyperparameters Used for Optimization with Optuna.

Hyperparameter	Range
Number of Hidden Layers	1
Number of Neurons per Hidden Layer	$\{16, 32, 64, 128, 256\}$
Batch Size	$\{16, 32, 64, 100, 128, 256, 512\}$
Replay Buffer Size	$\{1e4, 5e4, 1e5, 1e6\}$
Discount Factor	$\{0.9, 0.95, 0.98, 0.99, 0.995, 0.999, 0.9999\}$
Learning rate	$[1e-5, 1]$
Target Update Frequency	$\{0.9, 0.95, 0.98, 0.99, 0.995, 0.999, 0.9999\}$
Train Frequency	$\{1, 4, 8, 16, 128, 256, 1000\}$
Exploration Fraction	$[0, 0.5]$
Final Value of Random Action Probability	$[0, 0.2]$
Fourier Order (FF)	$\{1, 2, 3, 4, 5\}$

encodings, observations returned by environments are normalized with a min-max normalization. For computation reasons, only the learning rate and the Fourier order are re-optimized with Optuna. Note that in experiments involving both FF and FLF encodings, the FLF order is selected with Optuna to range from 1 to the FLF order corresponding to the number of traditional FF determined in prior research on FF.

11.2.2 Overall Performance

We apply Fourier Features (FF-NN) and Fourier Light Features (FLF-NN) to the off-policy Deep-Q Network (DQN) algorithm (Mnih et al., 2015) in discrete action environments and to the on-policy Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017) in continuous action environments. We then compare these implementations to the respective algorithms without encoding (NN). Figure 11.3 shows the averaged returns per timesteps for DQN on four discrete-action environments from OpenAI Gym (Brockman et al., 2016). Figure 11.4 shows the averaged returns per timesteps of PPO on five continuous-action control tasks from Mujoco (Todorov et al., 2012). For experiments on continuous action environments, we only test FLF because the number of standard Fourier Features explodes due to the higher state dimension. In all discrete tasks, except for the LunarLander-v2 task, both FLF and FF improve the sample efficiency, i.e., both FLF and FF have better performance in terms of cumulative rewards with fewer environment interactions. In the LunarLander-v2 task, their performance does not deteriorate. It is worth noting that in the MountainCar-v0 task, FLF and FF significantly increase the final cumulative reward, as traditional neural networks in this experiment are unable to converge to the optimal policy. The increase of final cumulative reward for FLF and FF is not observed in other discrete environments, as they are relatively simple. In all continuous tasks, FLF considerably outperforms the baseline in terms of both cumulative rewards and sample efficiency. Where the dimension is small enough so that we can apply FF, we obtain similar performance by FF and FLF. This observation suggests only the subset FLF of FF is required to enhance performance, as assumed in Section 11.1.1. Note also that in experiments with neural Fitted Q-Iterations of Section 10.2, FLF-NN better fits the optimal Q-value function than simple MLP architectures, as show in Figure 10.2. This better approximation can be explained by the learning of the high-frequency components of the optimal Q-function with FLF-NN and results in better performance.

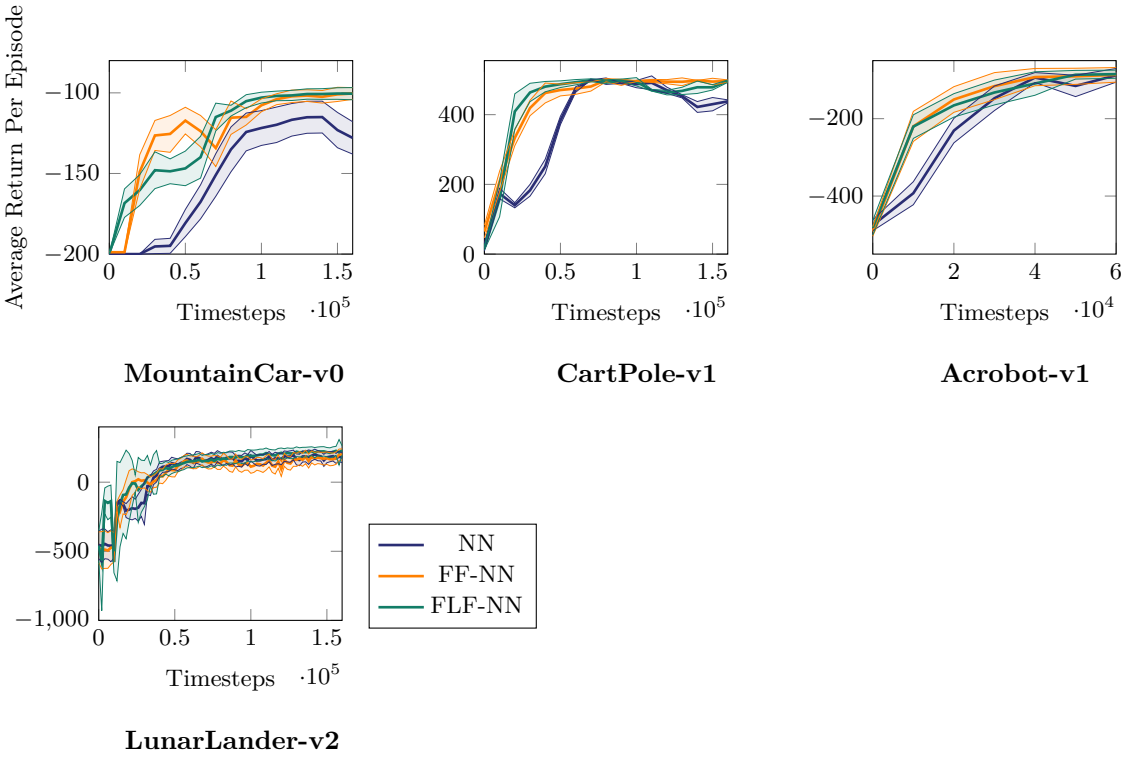


Figure 11.3: **The use of features encoding based on Fourier series improve performance and sample efficiency of DQN on discrete control tasks. Similar behavior is observed for FF-NN and FLF-NN.** Evaluation learning curves of NN (blue), FF-NN (orange), and FLF-NN (green), reporting episodic return versus environment timesteps. Results are averaged over 30 training (different seeds), with shading indicating the 95% confidence interval (CI).

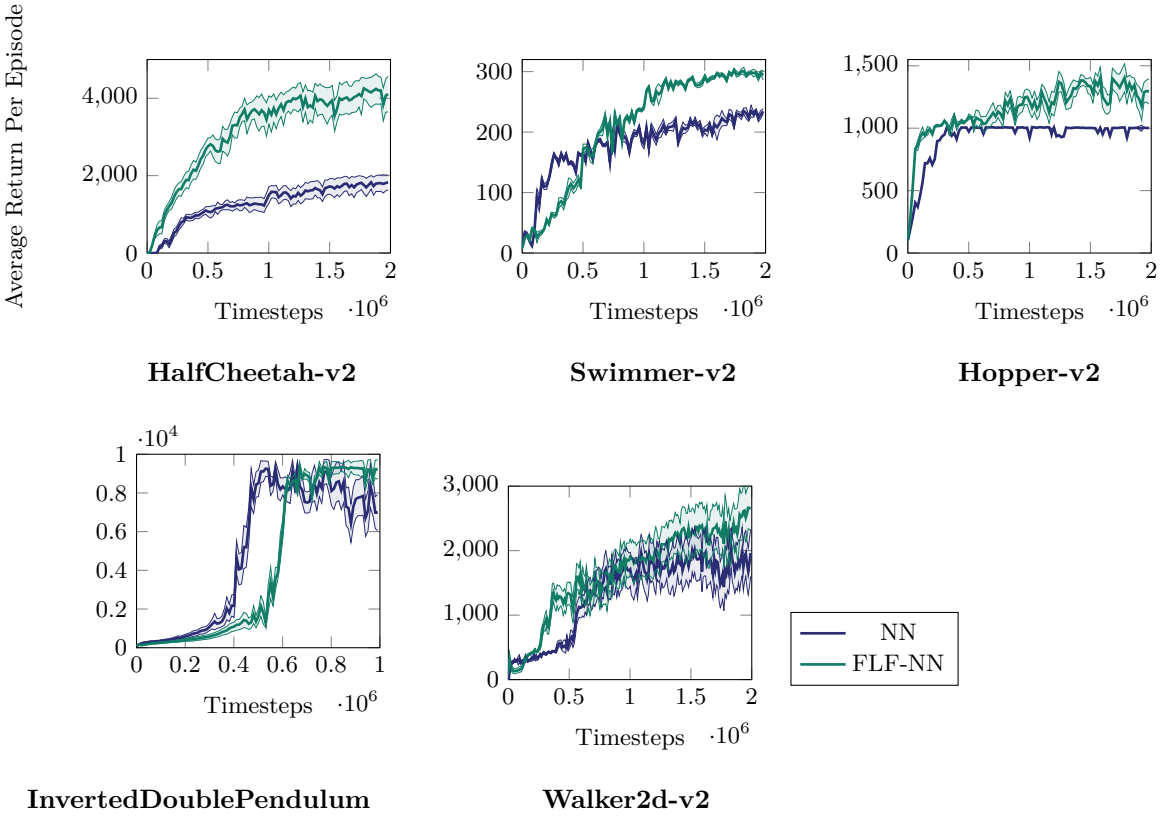


Figure 11.4: **The use of Fourier Light Features improves the performance and sample efficiency of PPO on continuous control tasks.** Evaluation learning curves of NN (blue) and FLF-NN (green), reporting episodic return versus environment timesteps. Results are averaged over 10 training with shading indicating the 95% confidence interval (CI).

11.2.3 Robustness to Hyperparameter Changes

RL algorithms can be very sensitive to hyperparameter changes (Henderson et al., 2018; Islam et al., 2017). The following experiments indicate that Fourier features reduce the sensitivity to hyperparameters. Figure 11.5 illustrates how the performance varies with the learning rate while keeping other hyperparameters constant. It shows that FF-DQN and FLF-DQN perform well over a larger range, although they require a smaller learning rate than NN. As discussed in Section 3.4, experience replay buffers (Lin, 1992; Mnih et al., 2015) and target networks (Mnih et al., 2015) were introduced in RL to mitigate interference problems and have become critical in the training of many deep RL algorithms including DQN. However, it is at the cost of higher computational and memory costs and slower offline learning (Plappert et al., 2018). Zhang and Sutton (2017) highlighted difficulties in properly tuning the buffer size where either too small or too big buffer can have a negative effect on performance. In Figure 11.6 and 11.7, we vary only the buffer size and target update frequency, respectively, while keeping other hyperparameters fixed. In the cases where standard DQN shows large performance variations for different buffer sizes and frequencies, we observe that FF-DQN is both better and less sensitive. This indicates a more stable learning process, with potentially less interference (see Section 12.1), and makes Fourier Features even more interesting for nonstationary and online problems.

11.2.4 Comparisons with Other Features Encodings

FF/FLF provide clear benefits, but it is natural to ask whether other classical feature encodings used in linear value function approximation might provide similar benefits. In this section, we compare the performance of Fourier features with the three following standard features encodings:

- *Polynomial Features* (PF-NN). Polynomials are one of the simplest families of feature encoding used for interpolation and regression. The feature vector consists of all polynomial combinations of the state variables with a degree less than or equal to a specified degree (Lagoudakis and Parr, 2003; Sutton and Barto, 2018).

- *Random Fourier Features (RFF).* Random Fourier Features are used to approximate an arbitrary stationary kernel-invariant by exploiting Bochner's theorem (Rahimi and Recht, 2007). Recent works have shown promising results where RFFs enhance the performance of deep neural networks (Mehrkanoon and Suykens, 2018), reduce the probability of misclassification (Mitra and Kaddoum, 2021), or facilitate the learning of high-frequency components (Tancik et al., 2020). In RL, RFFs have been used with Natural Policy Gradient to outperform performance obtained with NNs (Rajeswaran et al., 2017). The i-th feature of the Random Fourier Feature mapping $\text{RFF} : \mathbb{R}^d \to \mathbb{R}^p$ is

$$\text{RFF}_i(\boldsymbol{s}) = \tfrac{2}{\sqrt{p}} \cos(\boldsymbol{s}^T \boldsymbol{c}_i + \boldsymbol{b}_i), \tag{11.3}$$

  where $\boldsymbol{c} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_p)$, $\boldsymbol{b} \sim \mathcal{U}(0, 2\pi)$. The term $2/\sqrt{p}$ is used as a normalization factor to reduce the variance of the estimates. RFFs and Fourier features have a very similar definition, except that the vector $\boldsymbol{c}$ creating interaction between state variables is sampled from a normal distribution in RFFs.

- *Tile Coding (TC).* Tile Coding (Albus, 1971; Sutton and Barto, 2018) is a generalization of state aggregation, in which we cover the state space $\mathcal{S}$ with overlapping grids, known

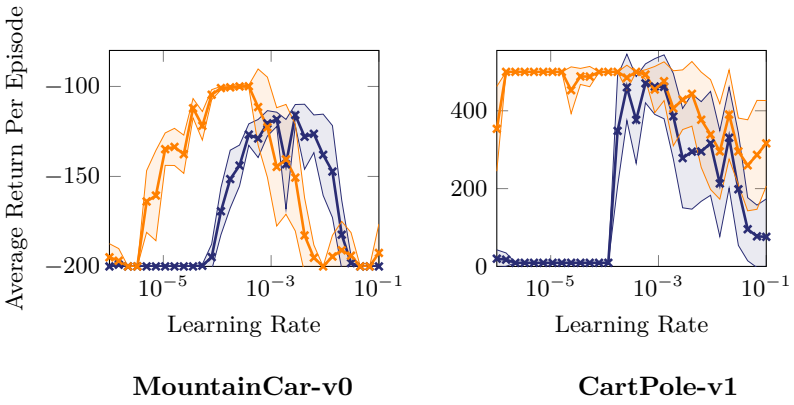


Figure 11.5: Learning rate variations over $n = 10$ trainings.

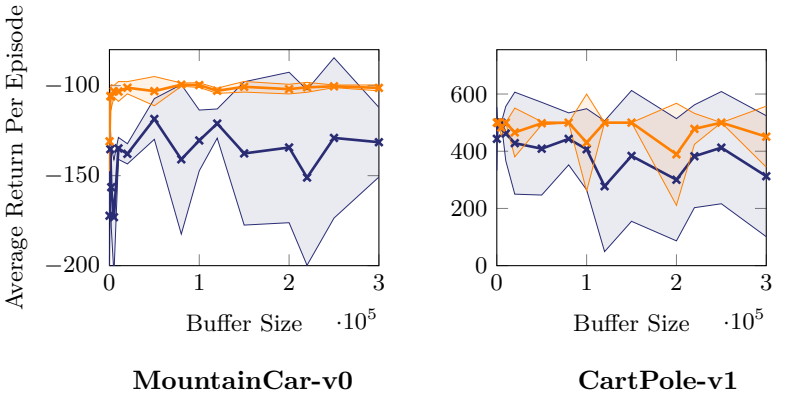


Figure 11.6: Buffer size variations over $n = 10$ trainings.

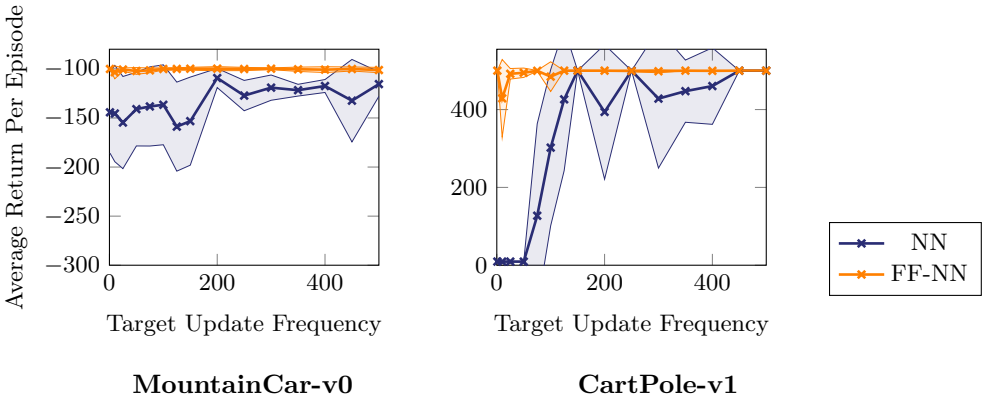


Figure 11.7: Target update variations over $n = 10$ trainings.

Figure 11.8: **Fourier Features are more robust to learning rate, buffer size and target update frequency.** Cumulative reward over different hyperparameter variations, for NN (blue) and FF-NN (orange) on MountainCar-v0 and CartPole-v1. Results are averaged over 10 trainings and shading indicating the 95% confidence interval (CI).

as *tilings*. Each grid divides the state space into small squares, referred to as *tiles*. The representation of a state for each tile is a one-hot vector of dimension the number of tiles, with one for the tile where the state is in and zero otherwise. Concatenation of one-hot vectors for each tiling forms Tile Coding features. A nice property of Tile Coding is that it generalizes not only to the trained state but also to any other states that share the same tiles. In Deep RL, Ghiassian et al. (2020) proposed to preprocess neural network inputs with Tile Coding to promote the sparsity of learned representations and obtain better performance.

Figure 11.9 depicts the averaged results per timesteps for DQN applied to MountainCar-v0 and CartPole-v1 tasks. In the experiments shown in Figure 11.9, none of the other features encodings achieve the performance of FF-NN/FLF-NN, even though we tuned their hyperparameters through an extensive search. It is even worse since PF and RFF degrade performance. In particular, the ranking on final average rewards is as follows: PF-NN $<$ RFF-NN $<$ NN $<$ TC-NN $<$ FF-NN $\simeq$ FLF-NN, where $<$ means lower performance. Note that TC-NN outperforms NN in the MountainCar-v0 task but exhibits similar performance in the CartPole-v1 task. From those experiments, we deduce that applying feature encodings to neural networks does not consistently lead to performance improvements and may even degrade performance.

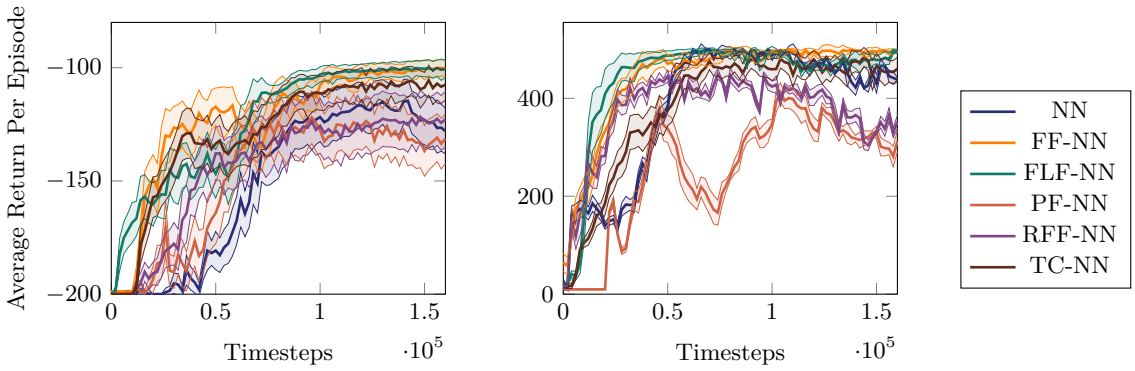


Figure 11.9: **Fourier Features/Fourier Light Features perform better than other standard features encodings on discrete control tasks with DQN.** Evaluation learning curves of NN (blue), FF-NN (orange), FLF-NN (green), PF-NN (red), RFF-NN (purple) and TC-NN (brown) reporting episodic return versus environment timesteps. Results are averaged over 30 trainings with shading indicating the standard deviation.

Chapter 12

Observed Effects on Training Neural Networks

In this chapter, we empirically investigate the effects of the Fourier encodings presented in Chapter 11 on the learning process of neural networks. In particular, we study the effects of Fourier features on DQN with MLP architectures using ReLU activation functions. In this setting, for any state-action pairs $(\boldsymbol{s}, a)$ in $\mathcal{S} \times \mathcal{A}$, the output of the neural network can be expressed as

$$\hat{Q}(\boldsymbol{s}, a; \boldsymbol{W}, \boldsymbol{\theta}) = \boldsymbol{\theta}^T \boldsymbol{\Phi}(\boldsymbol{s}, a; \boldsymbol{W}), \tag{12.1}$$

where $\boldsymbol{\theta} \in \mathbb{R}^N$ depicts the weights of the last linear output layers of the neural network; $\boldsymbol{W}$ depicts the weights of the preceding layers; and $\boldsymbol{\Phi}(\boldsymbol{s}, a; \boldsymbol{W}) \in \mathbb{R}^N$ represents the output of the penultimate layer, which captures the representations learned by the network. We denote by $\boldsymbol{\Theta} = \big[\boldsymbol{W}, \boldsymbol{\theta}\big] \in \mathbb{R}^{N'}$ the vector of parameters of the MLP architecture. In the following of this chapter, we study the effects of Fourier features on catastrophic interference in Section 12.1, on the sparsity of the learned features in Section 12.2, and the expressiveness of neural networks in Section 12.3.

12.1 Catastrophic Interference

Catastrophic interference occurs in function approximation when the learner "forgets" what it has learned in the past by overwriting previous updates to better fit the learned function to recent data (McCloskey and Cohen, 1989; French, 1991). In RL, this problem is further exacerbated by the fact that the agent uses its own estimates as targets and changes its policy during the training. Indeed, if estimates change incorrectly due to interference, there could be a cascading effect. Therefore, such interference can significantly slow down the learning and even prevent the network from converging to an optimal solution.

12.1.1 Learning Interference

In the following, we denote by $\boldsymbol{\Theta}_t$ the parameters of DQN at time t during the training. A typical measure of interference is the learning interference defined below.

Definition 12.1.1 (Learning Interference (Lopez-Paz and Ranzato, 2017; Riemer et al., 2018))**.** *At time $t+1$, after updating model parameters of DQN from $\boldsymbol{\Theta}_t$ to $\boldsymbol{\Theta}_{t+1}$ with its loss function l, the learning interference function $\mathrm{LI}_{t+1} : \mathcal{S} \times \mathcal{A} \times \mathbb{R} \times \mathcal{S} \to \mathbb{R}$ at time $t+1$ is defined for any transition $\boldsymbol{x} = (\boldsymbol{s}, a, r, \boldsymbol{s}') \in \mathcal{S} \times \mathcal{A} \times \mathbb{R} \times \mathcal{S}$ as*

$$\mathrm{LI}_{t+1}(\boldsymbol{x}) = l(\boldsymbol{x}; \boldsymbol{\Theta}_{t+1}) - l(\boldsymbol{x}; \boldsymbol{\Theta}_t). \tag{12.2}$$

Remark 35. *The positiveness or negativeness of $\mathrm{LI}_{t+1}(\boldsymbol{x})$ determines whether the update of model parameters of DQN improves or degrades predictions on the transition $\boldsymbol{x} = (\boldsymbol{s}, a, r, \boldsymbol{s}')$.*

Considering a Stochastic-Gradient Descent approach (equation 3.9), the update rule using the transition $\boldsymbol{x}_t = (\boldsymbol{s}_t, a_t, r_t, \boldsymbol{s}'_t) \in \mathcal{S} \times \mathcal{A} \times \mathbb{R} \times \mathcal{S}$ is given by

$$\boldsymbol{\Theta}_{t+1} \leftarrow \boldsymbol{\Theta}_t - \alpha_t \nabla_{\boldsymbol{\Theta}} l(\boldsymbol{x}_t; \boldsymbol{\Theta}_t),$$

where α_t is the learning rate at time t. Using the Taylor series expansion and assuming the learning rate α_t is small, we can rewrite equation 12.1.1 for $\boldsymbol{x} = (\boldsymbol{s}, a, r, \boldsymbol{s}') \in \mathcal{S} \times \mathcal{A} \times \mathbb{R} \times \mathcal{S}$ as

$$\begin{aligned} \mathrm{LI}_{t+1}(\boldsymbol{x}) &= l(\boldsymbol{x}; \boldsymbol{\Theta}_{t+1}) - l(\boldsymbol{x}; \boldsymbol{\Theta}_t) \\ &\approx \nabla_{\boldsymbol{\Theta}} l(\boldsymbol{x}; \boldsymbol{\Theta}_t)^T \big(\boldsymbol{\Theta}_{t+1} - \boldsymbol{\Theta}_t\big) \quad \text{(Taylor series expansion)} \\ &= -\alpha_t \nabla_{\boldsymbol{\Theta}} l(\boldsymbol{x}; \boldsymbol{\Theta}_t)^T \nabla_{\boldsymbol{\Theta}} l(\boldsymbol{x}_t; \boldsymbol{\Theta}_t). \end{aligned} \tag{12.3}$$

The quantity $\nabla_{\boldsymbol{\Theta}} l(\boldsymbol{x}; \boldsymbol{\Theta}_t)^T \nabla_{\boldsymbol{\Theta}} l(\boldsymbol{x}; \boldsymbol{\Theta}_t)$ is a key quantity to measure interference and is referred to in the literature as the *gradient alignment* (Bengio et al., 2020; Lopez-Paz and Ranzato, 2017; Riemer et al., 2018; Schaul et al., 2019). To quantify the learning interference, we prefer estimating the *stiffness* of the gradient alignment (Fort et al., 2020).

Definition 12.1.2 (Stiffness (Fort et al., 2020))**.** *Let $l(\cdot; \boldsymbol{\Theta}_t)$ be the loss function of a DQN of parameters $\boldsymbol{\Theta}_t$. The stiffness $\rho(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\Theta}_t)$ is defined for all transitions $\boldsymbol{x}_1 = (\boldsymbol{s}_1, a_1, r_1, \boldsymbol{s}'_1), \boldsymbol{x}_2 = (\boldsymbol{s}_2, a_2, r_2, \boldsymbol{s}'_2) \in \mathcal{S} \times \mathcal{A} \times \mathbb{R} \times \mathcal{S}$ as*

$$\rho(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\Theta}_t) = \cos\Big(\nabla_{\boldsymbol{\Theta}} l(\boldsymbol{x}_1; \boldsymbol{\Theta}_t), \nabla_{\boldsymbol{\Theta}} l(\boldsymbol{x}_2; \boldsymbol{\Theta}_t)\Big), \tag{12.4}$$

where $\cos(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^T \boldsymbol{v} / ||\boldsymbol{u}|| ||\boldsymbol{v}||$ is the cosine similarity of $\boldsymbol{u}$ and $\boldsymbol{v}$.

Remark 36. *From equation 12.3, the positiveness or negativeness of $\rho(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\Theta}_t)$ determines whether the update with the transition $\boldsymbol{x}_2$ is constructive (i.e. positive generalization) or destructive (i.e. interference) on the transition $\boldsymbol{x}_1$.*

Using the stiffness measure, we define three proxy measures for measuring the gradient interference of DQN with parameters $\boldsymbol{\Theta}$ and experience replay buffer $\mathcal{B}$:

- *Average Stiffness* (AS)
$$\mathrm{AS}(\boldsymbol{\Theta}, \mathcal{B}) = \mathbb{E}_{\boldsymbol{x}_1, \boldsymbol{x}_2 \sim \mathcal{B}}\big[\rho(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\Theta})\big];$$
- *Average Interference* (AI)
$$\mathrm{AI}(\boldsymbol{\Theta}, \mathcal{B}) = \mathbb{E}_{\boldsymbol{x}_1, \boldsymbol{x}_2 \sim \mathcal{B}}\big[\rho(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\Theta}) \mid \rho(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\Theta}) < 0\big],$$

which only considers (negatively) interfering samples and determines the average of interference;

- *Interference Risk* (IR):

$$\mathrm{IR}(\boldsymbol{\Theta}, \mathcal{B}) = \mathbb{E}\big[\rho(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\Theta}) \mid \rho(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\Theta}) \leq \mathrm{VaR}_{0.9}\big(\rho(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\Theta}) \mid \rho(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\Theta}) \leq 0\big)\big],$$

where $\mathrm{VaR}_{0.9}\big(\rho(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\Theta}) \mid \rho(\boldsymbol{x}_1, \boldsymbol{x}_2; \boldsymbol{\Theta}) \leq 0\big)$ is the 0.9-quantile of the distribution of interference measure.

12.1.2 Experiments

For experiments described in Section 11.2, we estimated the proxy measures AS, AI, and IR defined in the previous section to measure the gradient interference of DQN. For every $1,000$ environment timestep, we estimate the proxy measures using 64 samples drawn from the experience replay buffer. Our results averaged across all timesteps are reported in Table 12.1, and curves showing the evolution of interference during the training can be found in Appendix B.3. In all cases, the proxy measure AS shows that an update with a state-action pair has less impact on other neural network predictions with the use of Fourier Features/Fourier Light Features compared to raw inputs. This is confirmed by higher (better) AI and IR scores. Our observations indicate that using Fourier Features helps to generalize appropriately without overgeneralizing, leading to more stable training and better performance.

Table 12.1: **Fourier Features and Fourier Light Features mitigate learning interference on discrete control tasks**. Interference measures with Average of Stiffness (AS), Average of Interference (AI), and Interference Risk (IR) averaged across all timesteps for DQN fed with raw inputs (NN), Fourier Features (FF-NN), and Fourier Light Features (FLF-NN) on discrete control tasks. The symbol ↓ (↑) indicates that a lower (higher) score is better. Best interference measures are in **bold.**

Architecture			MountainCar-v0	Acrobot-v1	CartPole-v1	LunarLander-v2
NN	AS	↓	0.24	0.09	0.22	0.06
	AI	↑	−0.83	−0.60	−0.92	−0.56
	IR	↑	−0.91	−0.92	−0.99	−0.94
FF-NN	AS	↓	0.10	**0.03**	**0.05**	0.06
	AI	↑	**−0.47**	**−0.37**	**−0.73**	**−0.49**
	IR	↑	**−0.87**	**−0.80**	−0.98	**−0.92**
FLF-NN	AS	↓	**0.05**	**0.04**	**0.05**	**0.04**
	AI	↑	−0.54	**−0.38**	−0.86	−0.67
	IR	↑	**−0.87**	**−0.79**	−0.98	−0.94

Table 12.2 indicates measures of gradient interference obtained with DQN for experiments involving the traditional feature encodings considered in experiments in Section 11.2.4. Results indicate that the use of Polynomial Features, Random Fourier Features, and Tile Coding highly interferes during the training, resulting in poor Average of Interference (AI) and Interference Risk (IR) scores. These results are consistent with the performance scores of Section 11.2.4.

Table 12.2: Interference measures with Average of Stiffness (AS), Average of Interference (AI), and Interference Risk (IR) averaged across all timesteps for DQN fed with raw inputs (NN), Fourier Features (FF-NN), Fourier Light Features (FLF-NN), Polynomial Features (PF-NN), Random Fourier Features (RFF-NN), and Tile Coding (TC-NN) on discrete control tasks. The symbol ↓ (↑) indicates that a lower (higher) score is better. Best interference measures are in **bold.**

Tasks			NN	FF-NN	FLF-NN	PF-NN	RFF-NN	TC-NN
MountainCar-v0	AS	↓	0.24	0.1	**0.05**	0.21	0.14	0.1
	AI	↑	−0.83	**−0.47**	−0.54	−0.81	−0.88	−0.86
	IR	↑	−0.91	**−0.87**	**−0.87**	−0.95	−0.93	−0.93
CartPole-v1	AS	↓	0.22	**0.05**	**0.05**	0.13	0.49	**0.07**
	AI	↑	−0.92	**−0.73**	−0.86	−0.84	−0.97	−0.95
	IR	↑	−0.99	−0.98	−0.98	**−0.87**	−0.99	−0.97

12.2 Sparsity

In the tabular RL with lookup table representations discussed in Section 2.3, catastrophic interferences are unlikely since features and information are not shared across other states. The main limitations of this approach are that lookup table representations do not scale well with the size of the state space, and learning does not generalize across states. Therefore, it is desirable to learn representations $\boldsymbol{\Phi}(\cdot, \cdot; \boldsymbol{W})$ in equation 12.1 that can generalize across different states while ensuring "locality in the generalization", i.e., changing the features for a given state only affects the representation of other similar states. Sparse representations are such representations that promote the "locality in the generalization" and reduce interference.

12.2.1 Sparse Representations

In *sparse representations*, only a few features are active (nonzero) for any given input, so each update only impacts a few weights and is less likely to interfere with other updates (Liu et al., 2019b; Hernandez-Garcia and Sutton, 2019; Ghiassian et al., 2020; Pan et al., 2020). Another beneficial effect of sparsity is the promotion of locality, where similar inputs should produce similar features. Thus, it may be easier for the agent to make accurate predictions for an explored local region, as the local dynamics are likely to be simpler functions than the global dynamics. A recent line of works shows that learning sparse representations improves performance and reduces catastrophic interference (Liu et al., 2019b; Hernandez-Garcia and Sutton, 2019; Ghiassian et al., 2020; Pan et al., 2020). In this section, we investigate whether Fourier feature encodings may help the learning of sparse representations as it was observed by Ghiassian et al. (2020) in their study on Tile Coding with neural networks. This property could explain the good performance of FF/FLF reported in Section 11.2.2 and the reduction of learning interference measures discussed in Section 12.1.2.

12.2.2 Metrics

We quantify sparsity in the learned representations $\boldsymbol{\Phi}(\cdot, \cdot; \boldsymbol{W})$ defined in equation 12.1 with two proxy measures: the *normalized overlap* and the *instance sparsity* (Liu et al., 2019b; Hernandez-Garcia and Sutton, 2019; Pan et al., 2020). To compute these measures, we denote by D the

number of *dead neurons*, i.e., the number of neurons with a zero response value for any input. We refer to the remaining $A = N - D$ neurons as *alive*. Dead neurons may occur since the ReLU activation function is used and outputs 0.

Definition 12.2.1 (Normalized Activation Overlap (Hernandez-Garcia and Sutton, 2019))**.** *Let* $(\boldsymbol{s}_1, a_1)$ *and* $(\boldsymbol{s}_2, a_2)$ *be two state-action pairs in* $\mathcal{S} \times \mathcal{A}$*. The normalized activation overlap for two learned representations* $\boldsymbol{\Phi}(\boldsymbol{s}_1, a_1; \boldsymbol{W}) \in \mathbb{R}^N$ *and* $\boldsymbol{\Phi}(\boldsymbol{s}_2, a_2; \boldsymbol{W}) \in \mathbb{R}^N$ *is defined as*

$$\mathrm{NO}\big(\boldsymbol{\Phi}(\boldsymbol{s}_1, a_1; \boldsymbol{W}), \boldsymbol{\Phi}(\boldsymbol{s}_2, a_2; \boldsymbol{W})\big) = \frac{1}{A} \sum_{i=1}^{N} \mathbf{1}_{\boldsymbol{\Phi}_i(\boldsymbol{s}_1, a_1; \boldsymbol{W}) > 0 \wedge \boldsymbol{\Phi}_i(\boldsymbol{s}_2, a_2; \boldsymbol{W}) > 0}. \tag{12.5}$$

Remark 37. *The normalized overlap reflects the amount of shared activation between any two representations. When the normalized overlap between two representations is zero, there is no interference between their corresponding inputs.*

Remark 38. *The normalization with the number of neurons alive avoids misleadingly low scores in cases where only a few are alive.*

The other proxy measure used to quantify sparsity is the instance sparsity metric.

Definition 12.2.2 (Instance Sparsity (Liu et al., 2019b))**.** *Let* $(\boldsymbol{s}, a)$ *be a state-action pair in* $\mathcal{S} \times \mathcal{A}$*. The instance sparsity for a learned representation* $\boldsymbol{\Phi}(\boldsymbol{s}, a; \boldsymbol{W}) \in \mathbb{R}^N$*, denoted by* $\mathrm{IS}\big(\boldsymbol{\Phi}(\boldsymbol{s}, a; \boldsymbol{W})\big)$*, is defined as the percentage of active units in the feature vector* $\boldsymbol{\Phi}(\boldsymbol{s}, a; \boldsymbol{W})$*, i.e.,*

$$\mathrm{IS}\big(\boldsymbol{\Phi}(\boldsymbol{s}, a; \boldsymbol{W})\big) = \frac{1}{A} \sum_{i=1}^{N} \mathbf{1}_{\boldsymbol{\Phi}_i(\boldsymbol{s}, a; \boldsymbol{W}) > 0}.$$

12.2.3 Experiments

Every 1,000 environment timesteps, we compute the normalized overlap and instance sparsity proxy measures to estimate the sparsity of the learned representations $\boldsymbol{\Phi}(\cdot, \cdot; \boldsymbol{W}_t)$ for the experiments described in Section 11.2.2. In particular, we compute the percentage of dead neurons, the normalized overlap, and the instance sparsity during the training over the same dataset of state-action pairs $\mathcal{D} := \big\{(\boldsymbol{s}_i, a_i)\big\}_{i=1}^{n}$. State-action pairs $(\boldsymbol{s}_i, a_i)$ in $\mathcal{D}$ are drawn i.i.d from rollouts obtained with sub-optimal pre-trained policies and random policies. This construction of $\mathcal{D}$ aims to cover state-action pairs likely to be used during the learning. Therefore, estimating the percentage of dead neurons with $\mathcal{D}$ is more conservative than the true percentage of dead neurons since it includes alive neurons that are inactive in $\mathcal{D}$. Nevertheless, we believe that measuring sparsity scores over $\mathcal{D}$ makes more sense since it removes neurons only active in parts of the state space that are less likely to be visited by the agent.

Our results are summarized in Table 12.3; the corresponding curves as a function of environment timesteps can be found in Appendix B.1. In all tasks, the use of Fourier Features results in lower (and thus better) normalized overlap and instance sparsity. There are no dead neurons when using Fourier Features/Fourier Light Features, suggesting a better use of the neural network capacity. However, the use of Fourier Light Features increases the sparsity in one instance (CartPole-v1), even though the learning performance with Fourier Light Features is better than simple neural networks in all instances. Hence, sparsity does not seem to be the only beneficial effect of the use of Fourier Features/Fourier Light Features. Furthermore, as discussed in Section 12.1.2, neural

Table 12.3: **Fourier Features and Fourier Light Features promote sparsity on discrete control tasks**. Sparsity scores with the percentage of dead neurons (DN), normalized activation overlap (NO), and instance sparsity (IS) obtained for DQN fed with raw inputs (NN), Fourier Features (FF-NN), and Fourier Light Features (FLF-NN), averaged across environment timesteps. Averages are taken across all timesteps and margins of error of the 95% confidence interval (CI) are computed over 30 trainings. Lower sparsity scores are better and better scores are in **bold**.

Architecture		**MountainCar-v0**	**Acrobot-v1**	**CartPole-v1**	**LunarLander-v2**
NN	DN	0.47 ± 0.09	**0.0**	0.07 ± 0.02	0.0
	NO	0.72 ± 0.08	0.49 ± 0.04	0.63 ± 0.04	0.30 ± 0.01
	IS	0.78 ± 0.07	0.64 ± 0.02	0.66 ± 0.03	0.46 ± 0.02
FF-NN	DN	**0.0**	0.01	**0.0**	**0.0**
	NO	$\mathbf{0.37 \pm 0.06}$	$\mathbf{0.05 \pm 0.02}$	$\mathbf{0.52 \pm 0.02}$	$\mathbf{0.23 \pm 0.03}$
	IS	$\mathbf{0.57 \pm 0.05}$	$\mathbf{0.13 \pm 0.04}$	$\mathbf{0.60 \pm 0.02}$	$\mathbf{0.40 \pm 0.02}$
FLF-NN	DN	**0.0**	**0.0**	**0.0**	**0.0**
	NO	0.43 ± 0.10	0.16 ± 0.02	0.79 ± 0.07	0.39 ± 0.04
	IS	0.62 ± 0.08	0.30 ± 0.03	0.85 ± 0.06	0.55 ± 0.04

Table 12.4: Sparsity scores with percentage of dead neurons (DN), normalized activation overlap (NO) and instance sparsity (IS) obtained for DQN fed with raw inputs (NN), Fourier Features (FF-NN), Fourier Light Features (FLF-NN), Polynomial features (PF-NN), Random Fourier Features (RFF-NN) and Tile Coding (TC-NN) on discrete control tasks averaged across all timesteps. Averages and margins of error of the 95% CI are over 30 trainings. Lower sparsity scores are better and better scores are in **bold**.

Task		NN	FF-NN	FLF-NN	PF-NN	RFF-NN	TC-NN
MountainCar-v0	DN	0.47 ± 0.09	**0.0**	**0.0**	0.66 ± 0.08	0.48 ± 0.04	**0.0**
	NO	0.72 ± 0.08	$\mathbf{0.37 \pm 0.06}$	0.43 ± 0.10	0.80 ± 0.08	0.87 ± 0.06	0.77 ± 0.13
	IS	0.78 ± 0.07	$\mathbf{0.57 \pm 0.05}$	0.62 ± 0.08	0.84 ± 0.08	0.90 ± 0.05	0.86 ± 0.09
CartPole-v1	DN	0.07 ± 0.02	0.01 ± 0.0	**0.0**	0.23 ± 0.02	0.88 ± 0.07	**0.0**
	NO	0.63 ± 0.04	$\mathbf{0.52 \pm 0.02}$	0.79 ± 0.07	0.73 ± 0.07	0.58 ± 0.03	0.66 ± 0.06
	IS	0.66 ± 0.03	$\mathbf{0.60 \pm 0.02}$	0.85 ± 0.06	0.75 ± 0.05	$\mathbf{0.61 \pm 0.03}$	0.70 ± 0.04

networks using Fourier Light Features have less interference than those using Fourier Features, even if the latter provides sparser representations. Such results suggest that even if sparsity mitigates catastrophic interference, the use of Fourier Light Features may have other beneficial effects that reduce catastrophic interference.

Table 12.4 reports sparsity measures obtained with DQN for experiments involving the traditional feature encodings considered in Section 11.2.4. Results suggest that the other features encoding degrade sparsity. Even the use of Tile Coding, known to promote sparsity (Ghiassian et al., 2020), produces less sparse representations than standard DQN. Neural networks with Tile Coding, just as Fourier Features/Fourier Light Features, do not have dead neurons, while the number of dead neurons is increased with Polynomial Features and Random Fourier Features.

12.3 Expressiveness

A neural network needs to extract expressive and fine-grained local features to achieve good performance. This is particularly true when consecutive raw inputs are similar, and small differences

between inputs may lead to different actions. Enforcing sparsity can also promote *expressiveness* through the identification of key attributes by encouraging the input to be well-described by a small subset of attributes. In RL, an *implicit under-parameterization* phenomenon has been highlighted for value-based algorithms for Deep RL algorithms using bootstrapping estimates (Kumar et al., 2020; Luo et al., 2020; Lyle et al., 2021). The implicit under-parameterization phenomenon results in an excessive aliasing of learned features, i.e., learned features are mapped into a much smaller subspace than the feature space that could be generated by the neural network. Consequently, neural networks behave as under-parameterized networks, generate less rich features, and lead to poorer performance.

12.3.1 Effective Rank

To measure the expressiveness of a learned feature matrix $\mathbf{\Phi} \in \mathbb{R}^{N \times n}$, we compute the *effective rank* srank_δ of $\mathbf{\Phi}$.

Definition 12.3.1 (Effective Rank (Kumar et al., 2020))**.** *Let $\mathbf{\Phi} \in \mathbb{R}^{N \times n}$ be a learned feature matrix of n samples obtained with the N learned features returned by the penultimate layer of a neural network. The effective rank srank_δ of $\mathbf{\Phi}$ estimates the proportion of the sum of the k highest singular values $\sigma_1(\mathbf{\Phi}) \geq \ldots \geq \sigma_k(\mathbf{\Phi}) \geq 0$ of $\mathbf{\Phi}$ that capture $1-\delta$ (usually $\delta = 0.01$) of the sum of all singular values:*

$$\text{srank}_\delta(\mathbf{\Phi}) = \frac{1}{\min(N,n)} \min\left\{ k : \frac{\sum_{i=1}^{k} \sigma_i(\mathbf{\Phi})}{\sum_{i=1}^{\min(N,n)} \sigma_i(\mathbf{\Phi})} \geq 1 - \delta \right\}, \tag{12.6}$$

Remark 39. *Intuitively, this quantity represents the number of "effective" unique components of the feature matrix $\mathbf{\Phi}$ that form the basis for linearly approximating the targets. When the network aliases inputs by mapping them to a smaller subspace, $\mathbf{\Phi}$ has only a few active singular directions, and $\text{srank}_\delta(\mathbf{\Phi})$ takes thus a small value.*

Recent studies have shown an implicit under-parametrization phenomenon in neural value-based algorithms, with the measure of the low effective rank metric Kumar et al. (2020); Luo et al. (2020); Lyle et al. (2021). This issue is exacerbated in RL due to the lack of direct and accurate targets. Instead of using true targets, value-based algorithms approximate them with bootstrapping, i.e., by sequentially fitting outputs to target value estimates generated from the function learned in previous iterations. As these targets rely on estimates, they can not be used to extract expressive representations.

12.3.2 Experiments

For every 1,000 environment timesteps, we compute the effective rank measure on a learned feature matrix $\mathbf{\Phi}_t \in \mathbb{R}^{N \times n}$ to quantify the expressiveness of the learned representations $\mathbf{\Phi}(\cdot, \cdot; \boldsymbol{W}_t)$ for the experiments described in Section 11.2.2. The learned feature matrix $\mathbf{\Phi}_t \in \mathbb{R}^{N \times n}$ is built on samples of the dataset $\mathcal{D} = \left\{(\boldsymbol{s}_i, a_i)\right\}_{i=1}^{N}$ defined in Section 12.2.3; where the i^{th} row of $\mathbf{\Phi}_t$ is defined as $(\mathbf{\Phi}_t)_i = \mathbf{\Phi}(\boldsymbol{s}_i, a_i; \boldsymbol{W}_t)$. Figure 12.1 shows the normalized effective rank over the environment timesteps of training. The learned features are more expressive for neural networks using Fourier Features/Fourier Light Features in all instances. This may induce a better use of the network capacity and explain better performance. These results are consistent with the absence of dead neurons reported in Table 12.3 when using Fourier Features/Fourier Light Features.

Features learned with neural networks using Fourier Light Features are more expressive than those with Fourier Features in most instances. In Figure 12.1, all curves exhibit a similar trend. This observation is consistent with the findings of Kumar et al. (2020), who noted that the effective rank is a decreasing function with respect to the number of iterations for Deep RL algorithms using bootstrapping estimates. However, our experiments indicate that the decrease in the effective rank is less pronounced with Fourier Features/Fourier Light Features in most instances. This suggests a more stable learning process with less catastrophic interference for neural networks using Fourier Features and Fourier Light Features.

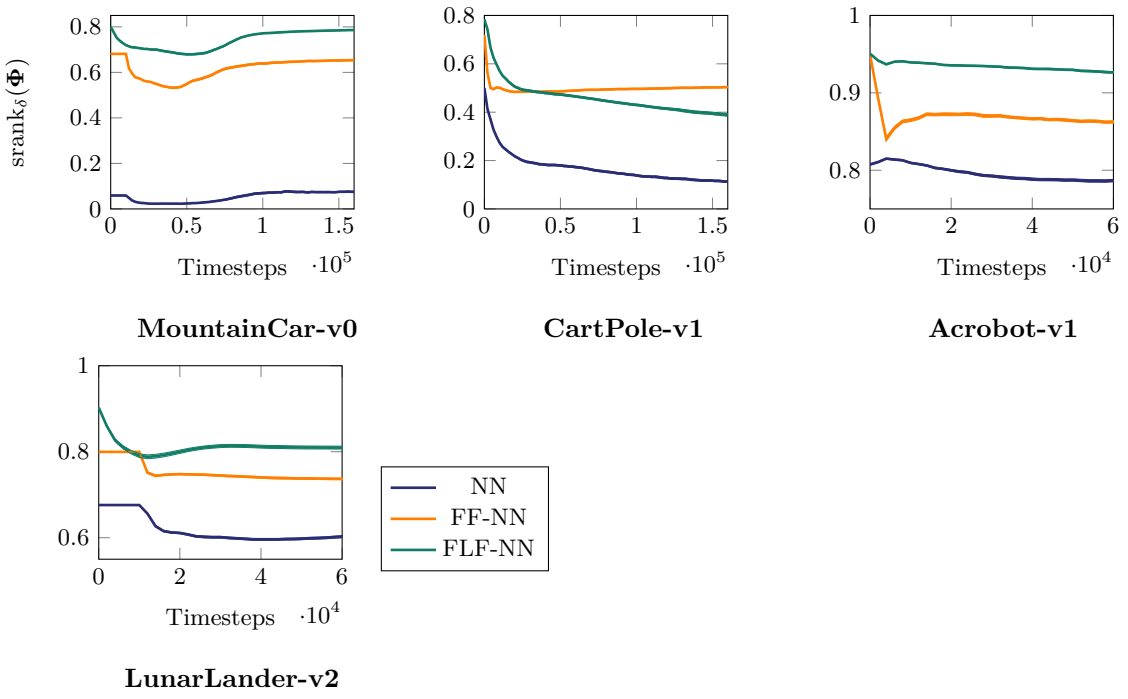


Figure 12.1: **The use of Fourier Features and Fourier Light Features enhances the expressiveness of the learned features on discrete control tasks.** Normalized effective rank srank$_\delta(\mathbf{\Phi}_t)$ over environment timesteps during the training for neural networks fed raw inputs (blue), Fourier Features (orange), and Fourier Light Features (green). Results are averaged over 30 trainings with shading indicating the 95% CI.

Figure 12.2 reports normalized effective rank measures obtained with DQN for experiments involving traditional feature encodings considered in Section 11.2.4. Neural networks with Polynomial Features and Random Fourier Features generate poorer learned features than neural networks fed with raw inputs. As expected, given the absence of dead neurons, the use of Tile-Coding produces richer features than neural networks without feature encodings and is on par with the use of Fourier Features/Fourier Light Features.

12.4 Smoothness

In deep learning, larger weight values lead to overfitting, which generally results in poor performance on unseen data (Neyshabur et al., 2015; Bartlett et al., 2017; Neyshabur et al., 2017). The idea follows Occam's razor that models with small weight norms are simpler and perform better

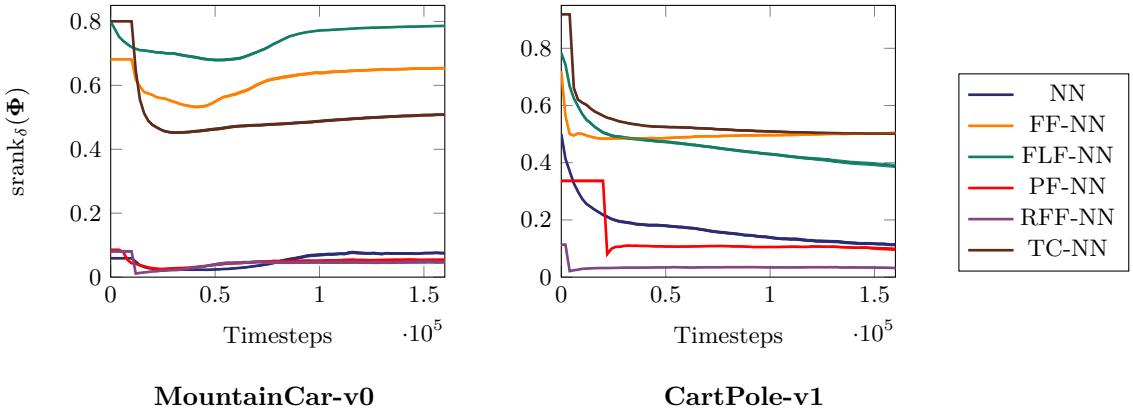


Figure 12.2: Normalized effective rank $\text{srank}_\delta(\mathbf{\Phi}_t)$ over environment timesteps during the training for neural networks fed with raw inputs (blue), Fourier features (orange), Fourier Light Features (green), Fourier Light Features (green), Polynomial Features (red), Random Fourier Features (purple) and Tile Coding features (brown). Results are averaged over 30 trainings with shading indicating the 95% CI.

than complex models. Not allowing individual weight norms to grow can also discourage large changes in output during the training. Similarly, regularization approaches that enforce small weight norms, such as weight decay, tend to produce better results in RL (Farebrother et al., 2018; Liu et al., 2020; Cobbe et al., 2019). A common approach to improving the smoothness of a neural network is to normalize weights to ensure that the learned layers are 1-Lipschitz. This kind of normalization not only improves the smoothness of the model but also enhances convergence (Salimans and Kingma, 2016; Gogianu et al., 2021) and reduces the generalization gap (Rosca et al., 2020; Gouk et al., 2021; Wang et al., 2019).

In order to know if Fourier features imporve the sommothness of neural networks, we need to compute their Lipschitz constant. The exact computation of the Lipschitz constant for a neural network is NP-hard (Scaman and Virmaux, 2018), but lower bounds and upper bounds can be estimated. In experiments described in Section 11.2.2, a lower bound is obtained by taking the largest norm of the gradient of DQN predictions with respect to the input (Rosca et al., 2020) across a dataset of $300,000$ state-action pairs. To estimate an upper bound, we compute the Lipschitz constants of each layer in isolation and multiply them (Gouk et al., 2021). Under the l_2 and l_1 norm, the upper bound of the Lipschitz constant of an MLP is given by the spectral norm and the maximum absolute column sum norm measure of the weight matrix (Neyshabur, 2017; Gouk et al., 2021). Figure 12.3 depict estimations of these bounds over environment timesteps of DQN training. Bounds are estimated every $1,000$ timestep during the training of DQN for experiments described in Section 11.2.2, and results are averaged over 30 trainings. In three out of the four tasks shown in Figure 12.3, neural networks with Fourier Features have a lower Lipschitz constant. Additional metrics, based on the l_1, l_2, and l_∞ norm of different layers, indicate that neural networks with Fourier Light Features can lie between neural networks with Fourier Features and simple neural networks, sometimes even surpassing neural networks with Fourier Features; see Appendix B.2.

Lipschitz bounds of DQN for experiments involving traditional features encodings considered in Section 11.2.4 are shown in Figure 12.4. Observations suggest that all feature encodings improve the Lipschitz bound of the neural network, with Tile Coding/Fourier Features/Fourier Light Features

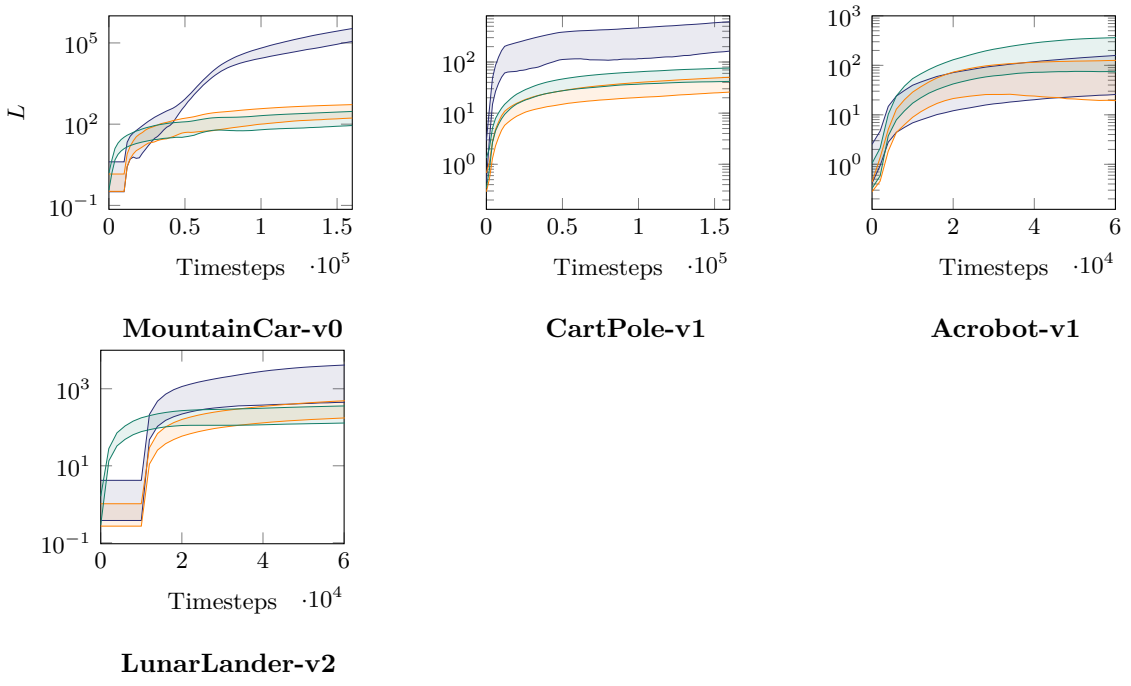


Figure 12.3: **Preprocessing inputs with Fourier Features or Fourier Light Features may improve the smoothness of the neural network.** Lower and upper bounds on the Lipschitz constant L of neural networks over environment timesteps during the training, for neural networks fed with raw inputs (blue), Fourier Features (orange), and Fourier Light Features (green). Bounds are averaged over 30 trainings. A lower score is better.

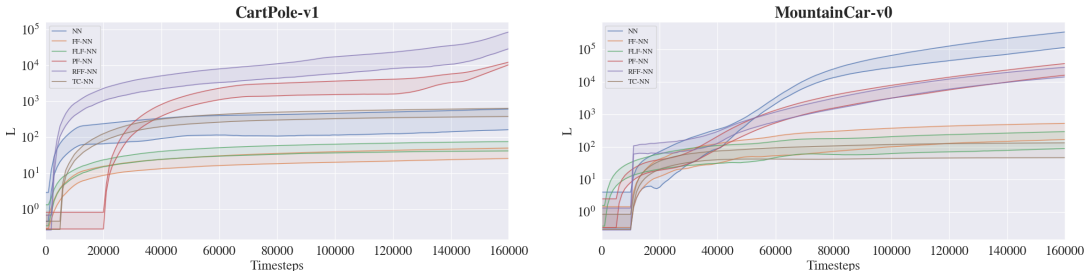


Figure 12.4: Lower and upper bounds on the Lipschitz constant of neural networks over environment timesteps during the training, for neural networks fed with raw inputs (blue), Fourier Features (orange), Fourier Light Features (green), Polynomial Features (red), Random Fourier Features (purple) and Tile Coding features (brown). Bounds are averaged over 30 trainings with shading indicating the 95% CI

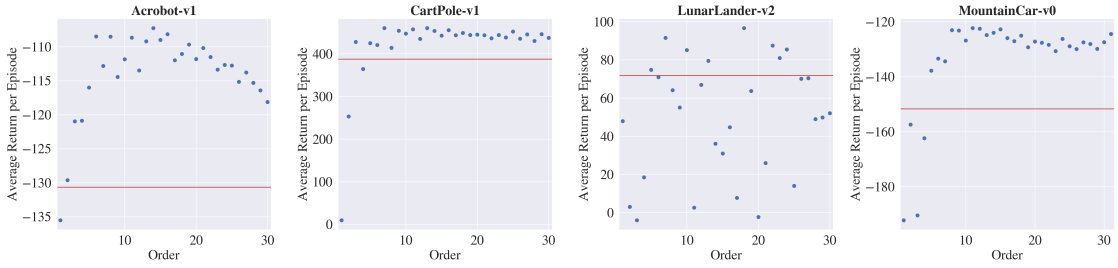


Figure 12.5: Cumulative rewards over varying FLF orders, averaged across all timesteps for 5 trainings with DQN fed with Fourier Light features. The red line indicates the performance for DQN without any preprocessing.

giving the best performance, followed by Polynomial Features/Random Fourier Features. In the shown instance, smoother networks correlate with better learning performance (see Section 11.2.4).

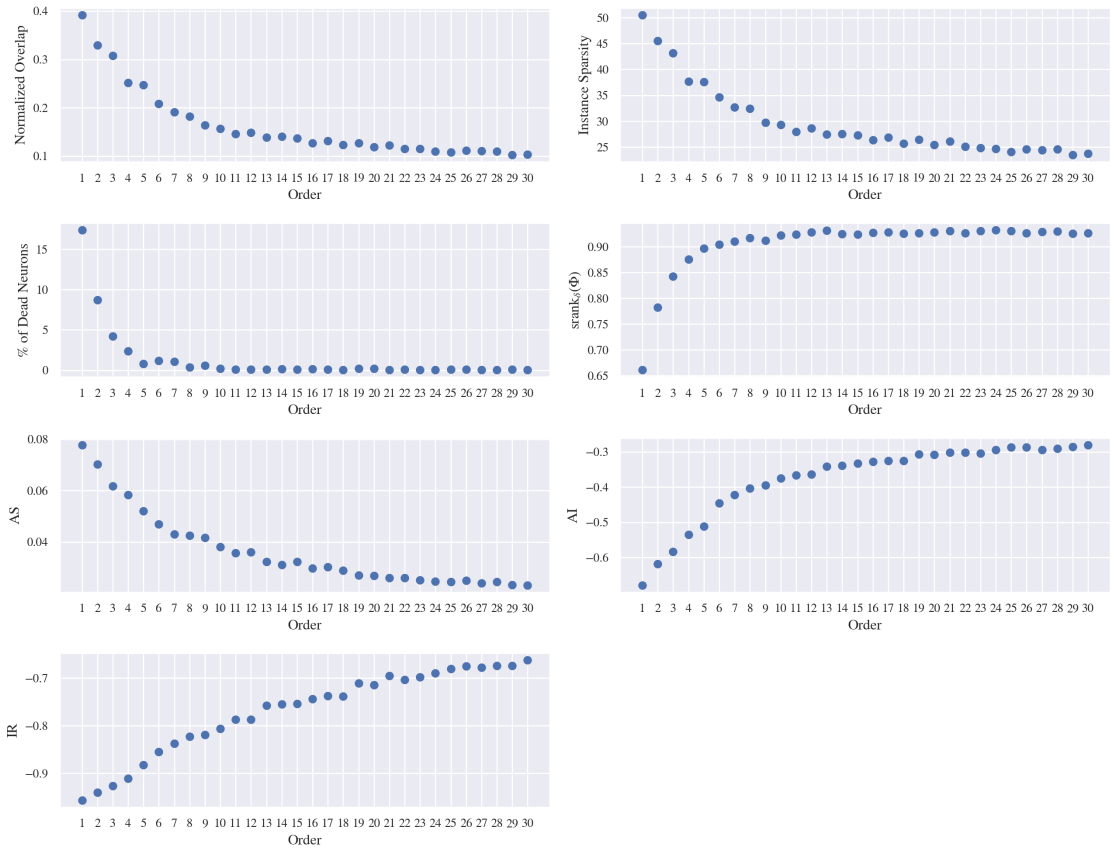
12.5 Correlations with the Fourier Light Features Order

In this section, we investigate the correlation between the performance/metrics presented in this section and the Fourier Light Features order. For our experiments, we adopt the same hyperparameter settings as those used in experiments described in Section 11.2.2. Figure 12.5 depicts the performance across different Fourier Light Features orders, where the performance is defined as the cumulative rewards from policy rollouts obtained after the training of DQN. We observe that increasing the Fourier Light Features order increases the performance up to a certain point, beyond which performance degrades. Fourier Light Features order can be considered as an additional hyperparameter for Deep RL algorithms. Only for the LunarLander-v2 task, the correlation between the performance after the training and the Fourier Light Features order is unclear.

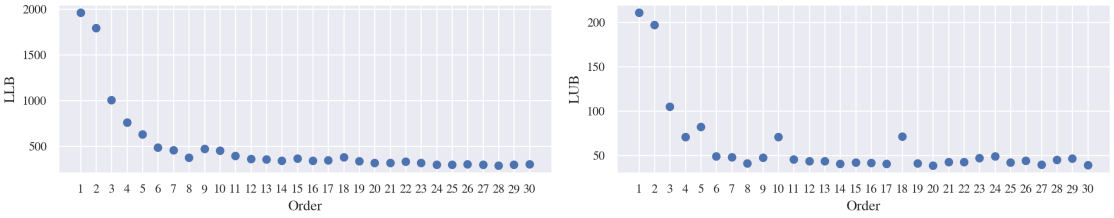
Table 12.5 summarizes the Spearman's rank correlation coefficients and p-values over a Fourier Light Features order for metrics studied in this chapter. Results indicate that increasing the Fourier Light Features order is strongly correlated with a better metric in almost all tasks and that this correlation is significant. In the CartPole-v1 task, the correlations are weaker, but a closer look at the graphs, shown in Figure 12.6, suggests that this is due to outliers and saturation in the metric.

Table 12.5: **Increasing the Fourier Light Features order improves the metrics**. The table shows Spearman's rank correlation coefficient r_S between different metrics and the FLF order. The p-value of the hypothesis test indicates high confidence in the result in almost all cases. The metrics are the percentage of dead neurons (DN), normalized activation overlap (NO), instance sparsity (IS), Average of Stiffness (AS), Average of Interference (AI), Interference Risk (IR), Lipschitz Lower Bound (LLB), Lipschitz Upper Bound (LUB), averaged across all environment timesteps for 5 trainings with DQN fed with Fourier Light features (FLF-NN), over an order varying from 1 to 30. ↓ and ↑ indicate the direction in which the metric is better.

		MountainCar-v0		**Acrobot-v1**		**CartPole-v1**		**LunarLander-v2**	
Metric		r_S	p-value	r_S	p-value	r_S	p-value	r_S	p-value
DN	↓	-0.876	2.198×10^{-10}	-0.882	1.173×10^{-10}	-0.658	7.768×10^{-5}	-0.869	2.188×10^{-10}
NO	↓	-0.991	4.529×10^{-26}	-0.93	1.143×10^{-13}	-0.77	6.692×10^{-7}	-0.562	1.009×10^{-3}
IS	↓	-0.992	2.215×10^{-26}	-0.724	6.082×10^{-6}	-0.75	1.795×10^{-6}	-0.131	4.836×10^{-1}
$\text{srank}_\delta(\Phi)$	↑	0.766	8.290×10^{-7}	0.998	7.749×10^{-36}	0.984	2.379×10^{-22}	1.0	0.000
AS	↓	-0.996	1.256×10^{-31}	-0.955	2.547×10^{-16}	-0.153	4.201×10^{-1}	-0.962	6.575×10^{-18}
AI	↑	0.994	3.122×10^{-28}	0.968	1.990×10^{-18}	0.501	4.780×10^{-3}	0.942	2.538×10^{-15}
IR	↑	0.996	1.256×10^{-31}	0.996	6.515×10^{-31}	0.668	5.566×10^{-5}	0.989	1.090×10^{-25}
LLB	↓	-0.962	2.380×10^{-17}	-0.352	5.631×10^{-2}	-0.828	1.610×10^{-8}	-0.977	5.119×10^{-21}
LUB	↓	-0.575	8.863×10^{-4}	0.268	1.521×10^{-1}	-0.66	7.228×10^{-5}	-0.98	7.797×10^{-22}



(a) Normalized overlap (NO), instance sparsity (IS), percentage of dead neurons (DN), srank, Average of Stiffness (AS), Average of Interference (AI), Interference Risk (IR), for CartPole-v1 task



(b) Lipschitz Lower Bound (LLB) and Lipschitz Upper Bound (LUB) for Acrobot-v1 task

Figure 12.6: Selected metrics over varying FLF orders, for two discrete control tasks

Conclusions and Perspectives

Conclusions

In this thesis, we have contributed to the domain of neural networks in deep RL in two ways, presented in two separate parts. We now briefly summarize our contributions.

In Part II, we have analyzed the performance of regularized LSTD with random features in a novel double asymptotic regime presented in Chapter 6, where the number of parameters N and distinct visited states m go to infinity with a constant ratio N/m. From the perspective of neural networks, by leveraging the lazy training regime, the performance of regularized LSTD with random features in high-dimensional problems can be interpreted as an approximation of the performance of deep TD learning algorithms using large single-hidden-layer neural networks. In Chapter 7, we have identified the resolvent of a non-symmetric positive-definite matrix that emerges as a crucial factor in the performance analysis of TD learning algorithms in terms of the error functions, and we have provided its deterministic equivalent form in the double asymptotic regime. Using this deterministic equivalent, we have derived deterministic limit forms of the empirical Mean-Squared Bellman Error (MSBE) on the collected transitions, the Mean-Squared Bellman Error (MSBE), and the Mean-Squared Value Error (MSVE). We have demonstrated that those deterministic forms expose correction terms that arise from the constant ratio N/m and that vanish as the ratio N/m or the l_2 regularization parameter increases. In Chapter 8, we have shown the asymptotic deterministic errors of regularized LSTD using random features are equivalent to the errors of a regularized kernel LSTD with implicit regularization. We have highlighted that correction factors can be interpreted and linked to classical notions from non-parametric statistics, e.g., with the effective dimension. In Chapter 9, we have observed our theoretical predictions match with experimental results for regularized LSTD with random features for any ratio N/m in real-world environments. From experiments, we have distinguished two regimes induced by corrections factors: an under-($N/m < 1$) and an over-($N/m > 1$) parameterized regime. In particular, we have observed a double descent phenomenon in the overparameterized regime for the MSBE and the MSVE induced by the correction factors. We have shown the correction terms vanish, and so does the double descent phenomenon when the l_2-regularization is increased, or the number of unvisited states goes to zero.

One remaining issue is the lack of interpretability of the corrections terms. In supervised learning, the theory of non-parametric models provides an interpretation of the correction terms, e.g., through the notion of degree of freedoms. A similar connection is missing in Reinforcement Learning because of the bootstrapping. Our work from Part II provides a stepping stone for further study of the influence of regularization, of the neural network architecture, and of the spectral components learned by neural networks. Some of these avenues will be discussed in further detail in Section 12.5.

In Part III, we have proposed and experimentally studied the effect of a type of Fourier encoding on Deep RL algorithms to mitigate the spectral bias. In particular, in Chapter 11, we have found that using preprocessings based on the Fourier series for neural networks provides a systematic increase in the final performance, sample efficiency, learning stability, and robustness to hyperparameters. Furthermore, we have proposed a light version of Fourier features, with only a linear number of features compared to the input size, that leads to similar benefits. In Chapter 12, we have conducted a detailed empirical analysis on the effects of Fourier encodings on the learning process. In particular, we have observed that the use of Fourier encodings improve the sparsity, expressiveness, and smoothness of neural networks, and reduce their catastrophic interference during learning. Ideally, the experimental analysis could be completed by a theoretical investigation. But this does not seem straightforward, since the nonlinearity of the proposed preprocessings complicates the analysis. Preprocessing inputs with features encodings for neural networks remains a promising direction for further research to overcome limitations of neural networks such as the spectral bias.

Perspectives

In the following, we discuss some possible future research directions in the continuation of this thesis.

A Refined Theoretical Analysis of the Influence of the Number of Parameters. In Part II, in our theoretical study of the influence of network size and l_2-regularization on the performance of TD learning algorithms, we have assumed that transitions are collected in the on-policy setting and we have considered the regularized LSTD algorithm with random features to approximate the behavior of neural TD learning algorithms in the lazy training regime. Directions for future work include a study of the off-policy setting and of gradient-based methods, which may lead to more complex behaviors. Furthermore, one could also relax the Gaussian assumption and consider the dynamics of learning within the hidden layers in more realistic regimes than the lazy training regime. Finally, one could go beyond policy evaluation to investigate the effects on the policy learned and on other RL algorithms, such as actor-critic methods.

"Proving" the Double Descent Phenomenon. In Chapter 7, we have shown that the asymptotic deterministic error functions in the double asymptotic regime feature corrections terms due to the constant ratio between the number of parameters and the number of distinct visited states. We have experimentally associated these corrections terms with the double descent phenomenon observed in Chapter 9. However, we did not mathematically prove the double descent phenomenon with corrections terms and why it occurs. Furthermore, a theoretical and empirical analysis can be performed to study whether, when, and how the double-descent phenomenon impacts TD learning algorithms using a gradient-based approach.

Generalization in TD Learning Algorithms. A theoretical understanding of generalization remains an open problem for many machine learning models. Using the double asymptotic regime considered in Part II, one can investigate the generalization error of RL aglorithms through a study of regularized kernel TD learning algorithms, as started in Chapter 8. By decomposing the generalization error into different spectral components in the Mercer feature space, one could

highlight which specific components of the value function are prioritized by TD learning algorithms with respect to the number of parameters. In particular, one could examine how the learning of these spectral components changes as the number of transitions collected or the number of distinct visited states collected grows. From a practical perspective, as the spectral components in the Mercer feature space depend on the dynamics of the environments, such understanding may be useful for the design of a reward function to improve the learning of value functions. In addition, this study could also improve our understanding of regularization in RL and propose new regularization penalities to improve generalization properties.

Appendices

Appendix A

Mathematical Proofs: Double Descent in LSTD

A.1 Proof of Theorem 7.2.3

Under Assumptions 1 and 2, this section is dedicated to prove the asymptotic equivalence between $\mathbb{E}[\boldsymbol{Q}_m(\lambda)]$ and

$$\bar{\boldsymbol{Q}}_m(\lambda) = \left[\frac{N}{m}\frac{1}{1+\delta}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n + \lambda\boldsymbol{I}_n\right]^{-1}$$

defined in Theorem 7.2.3, when $N, m \to \infty$. In order to prove Theorem 7.2.3, we shall proceed by introducing an intermediary resolvent $\tilde{\boldsymbol{Q}}_m(\lambda)$ (defined in equation A.1), and show subsequently under Assumptions 1 and 2 that

$$\|\mathbb{E}[\boldsymbol{Q}_m(\lambda)] - \tilde{\boldsymbol{Q}}_m(\lambda)\| \to 0 \quad \text{and} \quad \|\tilde{\boldsymbol{Q}}_m(\lambda) - \bar{\boldsymbol{Q}}_m(\lambda)\| \to 0,$$

as $N, m \to \infty$.

We denote the resolvent $\boldsymbol{Q}_m(\lambda)$ by $\boldsymbol{Q}_m$ to simplify the notations. The first half of the proof is dedicated to Lemma A.1.1, which proposes a first characterization of $\mathbb{E}[\boldsymbol{Q}_m]$ by $\tilde{\boldsymbol{Q}}_m$ as $N, m \to \infty$ under Assumptions 1 and 2. This preliminary step is classical in studying resolvents in the Random Matrix literature (Louart et al., 2018; Liao et al., 2020) as the direct comparison of $\mathbb{E}[\boldsymbol{Q}_m]$ to $\bar{\boldsymbol{Q}}_m$ with the implicit δ (equation 7.8) may be cumbersome.

Lemma A.1.1. *Under Assumptions 1 and 2, let $\lambda > 0$ and let $\tilde{\boldsymbol{Q}}_m(\lambda) \in \mathbb{R}^{n\times n}$ be the resolvent defined as*

$$\tilde{\boldsymbol{Q}}_m(\lambda) = \left[\frac{N}{m}\frac{1}{1+\alpha}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n + \lambda\boldsymbol{I}_n\right]^{-1}; \tag{A.1}$$

for the deterministic Gram feature matrix

$$\boldsymbol{\Phi}_{\hat{\mathcal{S}}} = \mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_d)}\left[\sigma(\boldsymbol{w}^T\hat{\boldsymbol{S}})^T\sigma(\boldsymbol{w}^T\hat{\boldsymbol{S}})\right],$$

and

$$\alpha = \frac{1}{m}\operatorname{Tr}\left((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\mathbb{E}[\boldsymbol{Q}_-(\lambda)]\right), \tag{A.2}$$

where

$$\boldsymbol{Q}_{-}(\lambda) = \left[\frac{1}{m}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\sigma(\boldsymbol{W}_{-}\hat{\boldsymbol{S}})^T\sigma(\boldsymbol{W}_{-}\hat{\boldsymbol{S}})\hat{\boldsymbol{U}}_n + \lambda\boldsymbol{I}_n\right]^{-1}, \tag{A.3}$$

for which $\boldsymbol{W}_{-} \in \mathbb{R}^{(N-1)\times d}$ *depicts the submatrix of the weight matrix* $\boldsymbol{W}$ *(defined in equation 6.5) without the first row. Then,*

$$\lim_{m\to\infty} \|\mathbb{E}_{\boldsymbol{W}}[\boldsymbol{Q}_m(\lambda)] - \tilde{\boldsymbol{Q}}_m(\lambda)\| = 0.$$

Remark 40. *Firstly, we can note that* α *is uniformly bounded. Since* $\frac{1}{m}\operatorname{Tr}(\boldsymbol{\Phi}_{\hat{\mathcal{S}}}) = \mathbb{E}\left[\frac{1}{m}\|\sigma(\boldsymbol{w}^T\hat{\boldsymbol{S}})\|^2\right]$ *and from Lemma A.7.2, we have*

$$\frac{1}{m}\operatorname{Tr}(\boldsymbol{\Phi}_{\hat{\mathcal{S}}}) = \int_0^\infty \Pr\left(\frac{1}{m}\|\sigma(\boldsymbol{w}^T\hat{\boldsymbol{S}})\|^2 > t\right)dt = \int_0^\infty 2t\Pr\left(\frac{1}{m}\|\sigma(\boldsymbol{w}^T\hat{\boldsymbol{S}})\| > t\right)dt = \mathcal{O}(1). \tag{A.4}$$

We deduce that

$$\alpha = \frac{1}{m}\operatorname{Tr}\big((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\mathbb{E}[\boldsymbol{Q}_{-}]\big) \leq \|\hat{\boldsymbol{U}}_n\mathbb{E}[\boldsymbol{Q}_{-}](\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\|\frac{1}{m}\operatorname{Tr}(\boldsymbol{\Phi}_{\hat{\mathcal{S}}}) = \mathcal{O}(1), \tag{A.5}$$

where we used $|\operatorname{Tr}(\boldsymbol{AB})| \leq \|\boldsymbol{A}\|\operatorname{Tr}(\boldsymbol{B})$ *for non-negative definite matrix* $\boldsymbol{B}$ *together with Lemma A.4.1 which asserts the operator norm of the resolvent* $\boldsymbol{Q}_{-}$ *is uniformly bounded. Furthermore, both* $\|\hat{\boldsymbol{U}}_n\|$ *and* $\|\hat{\boldsymbol{V}}_n\|$ *are upper bounded by* 1.

Proof. We decompose the matrix $\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}$ as

$$\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} = \sum_{i=1}^N \boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T, \tag{A.6}$$

where $\boldsymbol{\sigma}_i = \sigma(\hat{\boldsymbol{S}}^T\boldsymbol{w}_i) \in \mathbb{R}^m$ for which $\boldsymbol{w}_i \in \mathbb{R}^d$ denotes the i-th row of $\boldsymbol{W}$ (defined in equation 6.5). Using the resolvent identity (Lemma A.8.1), we write

$$\begin{aligned}
&\mathbb{E}[\boldsymbol{Q}_m] - \tilde{\boldsymbol{Q}}_m \\
&= \mathbb{E}\left[\boldsymbol{Q}_m\left[\tilde{\boldsymbol{Q}}_m^{-1} - \lambda\boldsymbol{I}_n - \frac{1}{m}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\right]\tilde{\boldsymbol{Q}}_m\right] \\
&= \frac{N}{m}\frac{1}{1+\alpha}\mathbb{E}[\boldsymbol{Q}_m](\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\tilde{\boldsymbol{Q}}_m - \frac{1}{m}\sum_{i=1}^N\mathbb{E}\left[\boldsymbol{Q}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\right]\tilde{\boldsymbol{Q}}_m \\
&= \frac{N}{m}\frac{1}{1+\alpha}\mathbb{E}[\boldsymbol{Q}_m](\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\tilde{\boldsymbol{Q}}_m - \frac{1}{m}\sum_{i=1}^N\mathbb{E}\left[\boldsymbol{Q}_{-i}\frac{(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n}{1 + \frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i}\right]\tilde{\boldsymbol{Q}}_m,
\end{aligned}$$

where the last equality is obtained with the Sherman identity (Lemma A.8.3) for

$$\boldsymbol{Q}_{-i} = \left[\frac{1}{m}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n - \frac{1}{m}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n + \lambda\boldsymbol{I}_n\right]^{-1} \tag{A.7}$$

independent of $\boldsymbol{\sigma}_i$ and thus $\boldsymbol{w}_i$. Exploiting this independence, we decompose

$$\mathbb{E}[\boldsymbol{Q}_m] - \tilde{\boldsymbol{Q}}_m \tag{A.8}$$

$$= \frac{N}{m}\frac{1}{1+\alpha}\mathbb{E}[\boldsymbol{Q}_m](\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\tilde{\boldsymbol{Q}}_m - \frac{1}{1+\alpha}\frac{1}{m}\sum_{i=1}^{N}\mathbb{E}\left[\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\right]\tilde{\boldsymbol{Q}}_m$$
$$+ \frac{1}{m}\frac{1}{1+\alpha}\sum_{i=1}^{N}\mathbb{E}\left[\boldsymbol{Q}_{-i}\frac{(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n(\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i - \alpha)}{1+\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i}\right]\tilde{\boldsymbol{Q}}_m \tag{A.9}$$

$$= \underbrace{\frac{1}{m}\frac{1}{1+\alpha}\sum_{i=1}^{N}\mathbb{E}[\boldsymbol{Q}_m - \boldsymbol{Q}_{-i}](\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\tilde{\boldsymbol{Q}}_m}_{=\boldsymbol{Z}_1}$$
$$+ \underbrace{\frac{1}{m}\frac{1}{1+\alpha}\sum_{i=1}^{N}\mathbb{E}\left[\boldsymbol{Q}_m\left(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n\right)^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\tilde{\boldsymbol{Q}}_m\left(\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i - \alpha\right)\right]}_{=\boldsymbol{Z}_2}. \tag{A.10}$$

The last equality is obtained by exploiting the Sherman identity (Lemma A.8.3) in reverse on the rightmost term and from the independence of $\boldsymbol{Q}_{-i}$ and $\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T$ for the second right-hand term. We want to prove that both $\boldsymbol{Z}_1$ and $\boldsymbol{Z}_2$ have a vanishing spectral norm under Assumptions 1 and 2. With both the resolvent identity (Lemma A.8.1) and the Sherman identity (Lemma A.8.3), we rewrite $\boldsymbol{Z}_1$ as

$$\begin{aligned}
\boldsymbol{Z}_1 &= \frac{1}{m}\frac{1}{1+\alpha}\sum_{i=1}^{N}\mathbb{E}[\boldsymbol{Q}_m - \boldsymbol{Q}_{-i}](\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\tilde{\boldsymbol{Q}}_m \\
&= -\frac{1}{m^2}\frac{1}{1+\alpha}\sum_{i=1}^{N}\mathbb{E}\left[\boldsymbol{Q}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}\right](\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\tilde{\boldsymbol{Q}}_m \\
&= -\frac{1}{m^2}\frac{1}{1+\alpha}\sum_{i=1}^{N}\mathbb{E}\left[\boldsymbol{Q}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\boldsymbol{D}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m\right](\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\tilde{\boldsymbol{Q}}_m \\
&= -\frac{1}{m^2}\frac{1}{1+\alpha}\mathbb{E}\left[\boldsymbol{Q}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{D}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m\right](\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\tilde{\boldsymbol{Q}}_m,
\end{aligned}$$

where $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ is a diagonal matrix for which, for all $i \in [N]$, we have

$$\boldsymbol{D}_i = \left(1 + \frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\right). \tag{A.11}$$

With a similar proof than for Lemma A.4.1, we can show there exists a $K_{\tilde{\boldsymbol{Q}}_m}$ such that, for all m, we have $\|\tilde{\boldsymbol{Q}}_m\| \leq K_{\tilde{\boldsymbol{Q}}_m}$ and then

$$\left\|\frac{1}{1+\alpha}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\tilde{\boldsymbol{Q}}_m\right\| = \left\|\frac{m}{N}(\boldsymbol{I}_n - \lambda\tilde{\boldsymbol{Q}}_m)\right\| \leq \frac{m}{N}(1 + \lambda K_{\tilde{\boldsymbol{Q}}_m}). \tag{A.12}$$

Furthermore, from Lemma A.1.4, we have

$$\left\|\frac{1}{m^2}\mathbb{E}\left[\boldsymbol{Q}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{D}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m\right]\right\| = \mathcal{O}\left(\frac{1}{m}\right). \tag{A.13}$$

Therefore, by combining both equation A.12 and equation A.13, we conclude that $\boldsymbol{Z}_1$ has a van-

ishing spectral norm, i.e.,

$$\|\boldsymbol{Z}_1\| = \left\| \frac{1}{m}\frac{1}{1+\alpha}\sum_{i=1}^{N} \mathbb{E}[\boldsymbol{Q}_m - \boldsymbol{Q}_{-i}](\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \boldsymbol{\Phi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \tilde{\boldsymbol{Q}}_m \right\| = \mathcal{O}\left(\frac{1}{m}\right). \tag{A.14}$$

We want to show now that $\boldsymbol{Z}_2$ also has a vanishing operator norm. For $i \in [N]$, by setting

$$\boldsymbol{B}_i = m^{\frac{1}{4}} \boldsymbol{Q}_m \left(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n\right)^T \boldsymbol{\sigma}_i \left(\frac{1}{m}\boldsymbol{\sigma}_i^T \hat{\boldsymbol{U}}_n \boldsymbol{Q}_{-i} (\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \boldsymbol{\sigma}_i - \alpha\right)$$

and

$$\boldsymbol{C}_i = m^{-\frac{1}{4}} \tilde{\boldsymbol{Q}}_m^T \hat{\boldsymbol{U}}_n^T \boldsymbol{\sigma}_i,$$

we decompose $\boldsymbol{Z}_2$ with its symmetric and its skew-symmetric part as

$$\begin{aligned}
\boldsymbol{Z}_2 &= \frac{1}{1+\alpha}\frac{1}{m}\sum_{i=1}^{N} \mathbb{E}\left[\boldsymbol{Q}_m \left(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n\right)^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \hat{\boldsymbol{U}}_n \tilde{\boldsymbol{Q}}_m \left(\frac{1}{m}\boldsymbol{\sigma}_i^T \hat{\boldsymbol{U}}_n \boldsymbol{Q}_{-i} (\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \boldsymbol{\sigma}_i - \alpha\right)\right] \\
&= \frac{1}{1+\alpha}\frac{1}{m}\sum_{i=1}^{N} \mathbb{E}\left[\boldsymbol{B}_i \boldsymbol{C}_i^T\right] \\
&= \frac{1}{1+\alpha}\frac{1}{m}\sum_{i=1}^{N} \mathbb{E}\left[\frac{\boldsymbol{B}_i\boldsymbol{C}_i^T + \boldsymbol{C}_i\boldsymbol{B}_i^T}{2}\right] + \frac{1}{1+\alpha}\frac{1}{m}\sum_{i=1}^{N} \mathbb{E}\left[\frac{\boldsymbol{B}_i\boldsymbol{C}_i^T - \boldsymbol{C}_i\boldsymbol{B}_i^T}{2}\right].
\end{aligned}$$

For the symmetric part, we use the relations $(\boldsymbol{B}_i - \boldsymbol{C}_i)(\boldsymbol{B}_i - \boldsymbol{C}_i)^T \succeq 0$ and $(\boldsymbol{B}_i + \boldsymbol{C}_i)(\boldsymbol{B}_i + \boldsymbol{C}_i)^T \succeq 0$ to deduce that

$$-\boldsymbol{B}_i\boldsymbol{B}_i^T - \boldsymbol{C}_i\boldsymbol{C}_i^T \preceq \boldsymbol{B}_i\boldsymbol{C}_i^T + \boldsymbol{C}_i\boldsymbol{B}_i^T \preceq \boldsymbol{B}_i\boldsymbol{B}_i^T + \boldsymbol{C}_i\boldsymbol{C}_i^T,$$

where $\preceq$ is the Loewner order for semi-positive-definite matrices. For the skew-symmetric part, we observe that $\|\mathbb{E}\left[\boldsymbol{B}_i\boldsymbol{C}_i^T - \boldsymbol{C}_i\boldsymbol{B}_i^T\right]\| = \|i\,\mathbb{E}\left[\boldsymbol{B}_i\boldsymbol{C}_i^T - \boldsymbol{C}_i\boldsymbol{B}_i^T\right]\|$ for $i^2 = -1$. With a similar reasoning than above, using the relations $(\boldsymbol{B}_i + i\boldsymbol{C}_i)(\boldsymbol{B}_i + i\boldsymbol{C}_i)^* \succeq 0$ and $-(\boldsymbol{B}_i - i\boldsymbol{C}_i)(\boldsymbol{B}_i - i\boldsymbol{C}_i)^* \preceq 0$, we deduce the relation

$$-\boldsymbol{B}_i\boldsymbol{B}_i^T - \boldsymbol{C}_i\boldsymbol{C}_i^T \preceq i(\boldsymbol{B}_i\boldsymbol{C}_i^T - \boldsymbol{C}_i\boldsymbol{B}_i^T) \preceq \boldsymbol{B}_i\boldsymbol{B}_i^T + \boldsymbol{C}_i\boldsymbol{C}_i^T.$$

From those relations, for both the symmetric and skew-symmetric parts, we have

$$\begin{aligned}
\|\boldsymbol{Z}_2\| &= \left\| \frac{1}{1+\alpha}\frac{1}{m}\sum_{i=1}^{N} \mathbb{E}\left[\boldsymbol{Q}\left(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n\right)^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \hat{\boldsymbol{U}}_n \tilde{\boldsymbol{Q}}_m \left(\frac{1}{m}\boldsymbol{\sigma}_i^T \hat{\boldsymbol{U}}_n \boldsymbol{Q}_{-i} (\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \boldsymbol{\sigma}_i - \alpha\right)\right] \right\| \\
&\leq \frac{1}{1+\alpha}\left(\left\| \sum_{i=1}^{N} \mathbb{E}\left[\frac{1}{m}\boldsymbol{B}_i\boldsymbol{B}_i^T\right] \right\| + \left\| \sum_{i=1}^{N} \mathbb{E}\left[\frac{1}{m}\boldsymbol{C}_i\boldsymbol{C}_i^T\right] \right\| \right).
\end{aligned} \tag{A.15}$$

From Lemma A.4.4, we know there exists a real $K'_{\boldsymbol{Q}_m} > 0$ such that, for all m, we have

$$\left\| \frac{1}{\sqrt{m}} \boldsymbol{Q}_m (\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \right\| \leq K'_{\boldsymbol{Q}_m}.$$

At this point,

$$\begin{aligned}
&\left\| \sum_{i=1}^{N} \mathbb{E}\left[\frac{1}{m} \boldsymbol{B}_i \boldsymbol{B}_i^T \right] \right\| \\
&= \left\| \sum_{i=1}^{N} \mathbb{E}\left[\frac{1}{\sqrt{m}} \boldsymbol{Q}_m \left(\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n \right)^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \left(\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n \right)^T \boldsymbol{Q}_m^T \left(\frac{1}{m} \boldsymbol{\sigma}_i^T \hat{\boldsymbol{U}}_n \boldsymbol{Q}_{-i} (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\sigma}_i - \alpha \right)^2 \right] \right\| \\
&= \left\| \sqrt{m}\, \mathbb{E}\left[\frac{1}{m} \boldsymbol{Q}_m \left(\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n \right)^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{D}_2^2 \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \left(\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n \right)^T \boldsymbol{Q}_m^T \right] \right\| \\
&\leq \sqrt{m} K_{\boldsymbol{Q}_m}'^2 \mathbb{E}[\|\boldsymbol{D}_2^2\|],
\end{aligned}$$

where $\boldsymbol{D}_2 \in \mathbb{R}^{N \times N}$ is a diagonal matrix for which, for all $i \in [N]$, we have

$$[\boldsymbol{D}_2]_i = \left(\frac{1}{m} \boldsymbol{\sigma}_i^T \hat{\boldsymbol{U}}_n \boldsymbol{Q}_{-i} (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\sigma}_i - \alpha \right).$$

From both Lemma A.7.4 and the union bound, we have

$$\Pr\left(\|\boldsymbol{D}_2\| > t \right) = \Pr\left(\max_{1 \leq i \leq N} [\boldsymbol{D}_2]_i > t \right) \leq CNe^{-cm \min(t, t^2)}$$

for some $c, C > 0$ independent of m and N. We have thus

$$\begin{aligned}
\mathbb{E}\left(\|\boldsymbol{D}_2\|^2 \right) = \mathbb{E}\left(\max_{1 \leq i \leq N} [\boldsymbol{D}_2^2]_i \right) &= \int_0^\infty \Pr\left(\max_{1 \leq i \leq N} [\boldsymbol{D}_2^2]_i > t \right) dt \\
&= \int_0^\infty 2t \Pr\left(\max_{1 \leq i \leq N} [\boldsymbol{D}_2]_i > t \right) dt \\
&\leq \int_0^\infty 2tCNe^{-cm \min(t, t^2)} dt \\
&= \int_0^1 2tCNe^{-cmt^2} dt + \int_1^\infty 2tCNe^{-cmt} dt \\
&\leq \int_0^\infty 2tCNe^{-cmt^2} dt + \int_0^\infty 2tCNe^{-cmt} dt \\
&= \frac{1}{m} \frac{2C}{c} \int_0^\infty tNe^{-t^2} dt + \frac{1}{m^2} \frac{2C}{c^2} \int_0^\infty tNe^{-t} dt \\
&= \mathcal{O}\left(\frac{1}{m} \right).
\end{aligned}$$

We deduce that

$$\left\| \sum_{i=1}^{N} \mathbb{E}\left[\frac{1}{m} \boldsymbol{B}_i \boldsymbol{B}_i^T \right] \right\| = \mathcal{O}\left(\frac{1}{\sqrt{m}} \right).$$

In addition, with a similar proof than for Lemma A.4.4, we can show there exists a real $K'_{\tilde{\boldsymbol{Q}}_m} > 0$ such that, for all m, we have

$$\left\| \sqrt{\frac{N}{m}} \sqrt{\frac{1}{1+\alpha}} \bar{\boldsymbol{Z}}^T \hat{\boldsymbol{U}}_n \tilde{\boldsymbol{Q}}_m \right\| \leq K'_{\tilde{\boldsymbol{Q}}_m},$$

where $\bar{\boldsymbol{Z}}\bar{\boldsymbol{Z}}^T$ is the Cholesky decomposition of $\boldsymbol{\Phi}_{\hat{\mathcal{S}}}$. Therefore,

$$\begin{aligned}\left\|\sum_{i=1}^{N}\mathbb{E}\left[\frac{1}{m}\boldsymbol{C}_i\boldsymbol{C}_i^T\right]\right\| &= \left\|\sum_{i=1}^{N}\mathbb{E}\left[\frac{1}{m\sqrt{m}}\tilde{\boldsymbol{Q}}_m^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\tilde{\boldsymbol{Q}}_m\right]\right\| \\ &= \left\|\frac{1}{\sqrt{m}}\frac{N}{m}\tilde{\boldsymbol{Q}}_m^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\tilde{\boldsymbol{Q}}_m\right\| \\ &= \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).\end{aligned}$$

From equation A.15 and above, we deduce that $\boldsymbol{Z}_2$ vanishes under the operator nom, i.e.,

$$\|\boldsymbol{Z}_2\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right). \tag{A.16}$$

Using both equation A.14 and equation A.16 into equation A.10, we conclude that

$$\|\mathbb{E}[\boldsymbol{Q}_m] - \tilde{\boldsymbol{Q}}_m\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right). \tag{A.17}$$

□

To get Theorem 7.2.3, we start from Lemma A.1.1 and we show that

$$\|\bar{\boldsymbol{Q}}_m(\lambda) - \tilde{\boldsymbol{Q}}_m(\lambda)\| \to 0,$$

as $N, m \to \infty$.

Theorem A.1.2 (Asymptotic Deterministic Resolvent)**.** *Under Assumptions 1 and 2, let $\lambda > 0$ and let $\bar{\boldsymbol{Q}}_m(\lambda) \in \mathbb{R}^{n\times n}$ be the resolvent defined as*

$$\bar{\boldsymbol{Q}}_m(\lambda) = \left[\frac{N}{m}\frac{1}{1+\delta}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n + \lambda\boldsymbol{I}_n\right]^{-1},$$

where δ is the correction factor defined as the unique positive solution to

$$\delta = \frac{1}{m}\operatorname{Tr}\left((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\left[\frac{N}{m}\frac{1}{1+\delta}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n + \lambda\boldsymbol{I}_n\right]^{-1}\right).$$

Then,

$$\lim_{m\to\infty}\left\|\mathbb{E}_{\boldsymbol{W}}\left[\boldsymbol{Q}_m(\lambda)\right] - \bar{\boldsymbol{Q}}_m(\lambda)\right\| = 0.$$

Proof. From Lemma A.6.1 in Appendix A.6, we know that δ exists and is the unique positive solution of equation 7.8 under Assumptions 1 and 2. From Lemma A.1.1 we have a first asymptotic equivalent of $\mathbb{E}_{\boldsymbol{W}}[\boldsymbol{Q}_m]$ given by

$$\tilde{\boldsymbol{Q}}_m = \left[\frac{N}{m}\frac{1}{1+\alpha}\left(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n\right)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n + \lambda\boldsymbol{I}_n\right]^{-1},$$

where

$$\alpha = \frac{1}{m}\operatorname{Tr}\big((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\mathbb{E}[\boldsymbol{Q}_-]\big),$$

since

$$\lim_{m\to\infty} \|\mathbb{E}_{\boldsymbol{W}}[\boldsymbol{Q}_m] - \tilde{\boldsymbol{Q}}_m\| = 0.$$

To finish the proof of the Theorem, we want to show that

$$\lim_{m\to\infty} \|\tilde{\boldsymbol{Q}}_m - \bar{\boldsymbol{Q}}_m\| = 0. \tag{A.18}$$

From the resolvent identity (Lemma A.8.1), we have

$$\|\tilde{\boldsymbol{Q}}_m - \bar{\boldsymbol{Q}}_m\| = \frac{N}{m}\frac{|\alpha-\delta|}{(1+\delta)(1+\alpha)} \left\| \tilde{\boldsymbol{Q}}_m \left(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n\right)^T \boldsymbol{\Phi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \bar{\boldsymbol{Q}}_m \right\|. \tag{A.19}$$

Let $\bar{\boldsymbol{Z}}\bar{\boldsymbol{Z}}^T$ be the Cholesky decomposition of $\boldsymbol{\Phi}_{\hat{\mathcal{S}}}$. With a similar proof than for Lemma A.4.4, we can show there exists a real $K'_{\tilde{\boldsymbol{Q}}} > 0$ such that, for all m, we have

$$\left\| \sqrt{\frac{1}{1+\alpha}}\sqrt{\frac{N}{m}} \tilde{\boldsymbol{Q}}_m \left(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n\right)^T \bar{\boldsymbol{Z}} \right\| \leq K'_{\tilde{\boldsymbol{Q}}}.$$

Similarly, we can show there exists a real $K'_{\bar{\boldsymbol{Q}}} > 0$ such that, for all m, we have

$$\left\| \sqrt{\frac{1}{1+\delta}}\sqrt{\frac{N}{m}} \bar{\boldsymbol{Z}}^T \hat{\boldsymbol{U}}_n \bar{\boldsymbol{Q}}_m \right\| \leq K'_{\bar{\boldsymbol{Q}}}.$$

Therefore,

$$\left\| \tilde{\boldsymbol{Q}}_m \left(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n\right)^T \boldsymbol{\Phi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \bar{\boldsymbol{Q}}_m \right\| \leq \sqrt{(1+\delta)(1+\alpha)}\frac{m}{N} K'_{\tilde{\boldsymbol{Q}}} K'_{\bar{\boldsymbol{Q}}}.$$

As a consequence, in order to prove equation A.18, it remains to prove that

$$\lim_{m\to\infty} |\alpha - \delta| = 0.$$

We decompose $|\alpha - \delta|$ as

$$|\alpha-\delta| = \left| \frac{1}{m} \operatorname{Tr}\big((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \boldsymbol{\Phi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \big[\mathbb{E}[\boldsymbol{Q}_-] - \bar{\boldsymbol{Q}}_m\big]\big) \right| \tag{A.20}$$

$$\leq \underbrace{\left| \frac{1}{m} \operatorname{Tr}\big((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \boldsymbol{\Phi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \big[\mathbb{E}[\boldsymbol{Q}_-] - \tilde{\boldsymbol{Q}}_m\big]\big) \right|}_{=Z_1} + \underbrace{\left| \frac{1}{m} \operatorname{Tr}\big((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \boldsymbol{\Phi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \big[\tilde{\boldsymbol{Q}}_m - \bar{\boldsymbol{Q}}_m\big]\big) \right|}_{=Z_2}. \tag{A.21}$$

To show Z_1 vanishes, we write α as

$$\begin{aligned}\alpha &= \frac{1}{m} \operatorname{Tr}\big((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \boldsymbol{\Phi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \mathbb{E}[\boldsymbol{Q}_-]\big) \\ &= \frac{1}{m} \operatorname{Tr}\big((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \boldsymbol{\Phi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \mathbb{E}[\boldsymbol{Q}_m]\big) + \frac{1}{m} \operatorname{Tr}\big((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \boldsymbol{\Phi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \big[\mathbb{E}[\boldsymbol{Q}_-] - \mathbb{E}[\boldsymbol{Q}_m]\big]\big).\end{aligned}$$

There exists a real $K > 0$ such that

$$\frac{1}{m} \operatorname{Tr}\big((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \boldsymbol{\Phi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \big[\mathbb{E}[\boldsymbol{Q}_-] - \mathbb{E}[\boldsymbol{Q}_m]\big]\big) \leq K \left\| \big[\mathbb{E}[\boldsymbol{Q}_-] - \mathbb{E}[\boldsymbol{Q}_m]\big] \right\|;$$

since both $\|\hat{\boldsymbol{U}}_n\|$ and $\|\hat{\boldsymbol{V}}_n\|$ are upper bounded by 1, $|\operatorname{Tr}(\boldsymbol{A}\boldsymbol{B})| \leq \|\boldsymbol{A}\| \operatorname{Tr}(\boldsymbol{B})$ for non-negative

definite matrix $\boldsymbol{B}$, and from equation A.4 that uniformly bounds $\frac{1}{m}\operatorname{Tr}(\boldsymbol{\Phi}_{\hat{\mathcal{S}}})$. From Lemma A.1.4, we have

$$\begin{aligned}\|\mathbb{E}\left[\boldsymbol{Q}_m - \boldsymbol{Q}_-\right]\| &= \left\| \frac{1}{m}\frac{m}{N}\sum_{i=1}^{N} \mathbb{E}\left[\boldsymbol{Q}_m - \boldsymbol{Q}_{-i}\right] \right\| \\ &= \left\| \frac{1}{m^2}\frac{m}{N}\mathbb{E}\left[\boldsymbol{Q}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{D}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m\right] \right\| \\ &= \mathcal{O}\left(\frac{1}{m}\right),\end{aligned}$$

where $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ is a diagonal matrix for which, for all $i \in [N]$, we have

$$\boldsymbol{D}_i = \left(1 + \frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\right).$$

As a consequence, by combining the results above and from Lemma A.1.1, we conclude for Z_1 that

$$|Z_1| = \left|\alpha - \frac{1}{m}\operatorname{Tr}\big((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\tilde{\boldsymbol{Q}}_m\big)\right| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

Using the vanishing result of Z_1 into equation A.21 and applying the resolvent identity (Lemma A.8.1) on Z_2, we get

$$|\alpha-\delta| \le \frac{N}{m}\frac{|\alpha-\delta|}{(1+\delta)(1+\alpha)}\left|\frac{1}{m}\operatorname{Tr}\big((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\tilde{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\bar{\boldsymbol{Q}}_m\big)\right| + \mathcal{O}\left(\frac{1}{\sqrt{m}}\right),$$

which implies that

$$|\alpha-\delta|\left(1 - \frac{N}{m}\frac{1}{(1+\delta)(1+\alpha)}\frac{1}{m}\operatorname{Tr}\big((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\tilde{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\bar{\boldsymbol{Q}}_m\big)\right) = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

It remains to show

$$\lim_{m\to\infty}\sup_m \frac{1}{m}\frac{N}{m}\frac{1}{(1+\delta)(1+\alpha)}\operatorname{Tr}\big((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\tilde{\boldsymbol{Q}}_m\big(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n\big)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\bar{\boldsymbol{Q}}_m\big) < 1.$$

Let the matrices $\boldsymbol{B}_n = (\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n$, $\boldsymbol{B}'_n = \bar{\boldsymbol{Z}}^T\hat{\boldsymbol{A}}_m\boldsymbol{Z}$, $\bar{\boldsymbol{Q}}'_m = \left[\frac{N}{m}\frac{1}{1+\delta}\boldsymbol{B}'_n + \lambda\boldsymbol{I}_m\right]^{-1}$, and $\tilde{\boldsymbol{Q}}'_m = \left[\frac{N}{m}\frac{1}{1+\alpha}\boldsymbol{B}'_n + \lambda\boldsymbol{I}_m\right]^{-1}$; where $\hat{\boldsymbol{A}}_m = \hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T$ is the empirical transition model matrix defined in equation 6.14. Using the Cauchy–Schwarz inequality, we write

$$\begin{aligned}&\frac{1}{m}\frac{N}{m}\frac{1}{(1+\delta)(1+\alpha)}\operatorname{Tr}\big(\boldsymbol{B}_n\tilde{\boldsymbol{Q}}_m\boldsymbol{B}_n\bar{\boldsymbol{Q}}_m\big) \\ &= \frac{1}{m}\frac{N}{m}\frac{1}{(1+\delta)(1+\alpha)}\operatorname{Tr}\big(\boldsymbol{B}'_n\tilde{\boldsymbol{Q}}'_m\boldsymbol{B}'_n\bar{\boldsymbol{Q}}'_m\big) \\ &\le \sqrt{\underbrace{\frac{N}{m}\frac{1}{m}\frac{1}{(1+\delta)^2}\operatorname{Tr}\big(\boldsymbol{B}'_n\bar{\boldsymbol{Q}}'_m\bar{\boldsymbol{Q}}'^T_m\boldsymbol{B}'^T_n\big)}_{Z'_1}\underbrace{\frac{N}{m}\frac{1}{m}\frac{1}{(1+\alpha)^2}\operatorname{Tr}\big(\boldsymbol{B}'_n\tilde{\boldsymbol{Q}}'_m\tilde{\boldsymbol{Q}}'^T_m\boldsymbol{B}'^T_n\big)}_{Z'_2}}.\end{aligned}$$

We observe that

$$\delta = \frac{1}{m}\operatorname{Tr}\big(\boldsymbol{B}_n\bar{\boldsymbol{Q}}_m\big) = \frac{1}{m}\operatorname{Tr}\big(\boldsymbol{B}'_n\bar{\boldsymbol{Q}}'_m\big) = \frac{1}{m}\operatorname{Tr}\big(\boldsymbol{B}'_n\bar{\boldsymbol{Q}}'_m\boldsymbol{Q}'^T_m\boldsymbol{Q}_m^{-1T}\big)$$

$$= \frac{1}{m}\frac{N}{m}\frac{1+\delta}{(1+\delta)^2}\operatorname{Tr}\big(\boldsymbol{B}'_n\bar{\boldsymbol{Q}}'_m\bar{\boldsymbol{Q}}'^T_m\boldsymbol{B}'^T_n\big) + \frac{\lambda}{m}\operatorname{Tr}\big(\boldsymbol{B}'_n\bar{\boldsymbol{Q}}'_m\bar{\boldsymbol{Q}}'^T_m\big).$$

Since $H(\boldsymbol{B}'_n)$ is at least semi-positive-definite under Assumption 2, we have

$$\operatorname{Tr}\big(\boldsymbol{B}'_n\bar{\boldsymbol{Q}}'_m\bar{\boldsymbol{Q}}'^T_m\big) = \operatorname{Tr}\big(\bar{\boldsymbol{Q}}'^T_m\boldsymbol{B}'_n\bar{\boldsymbol{Q}}'_m\big) = \operatorname{Tr}\big(\bar{\boldsymbol{Q}}'^T_m H(\boldsymbol{B}'_n)\bar{\boldsymbol{Q}}'_m\big) \geq 0.$$

As a consequence, we have

$$\frac{1}{m}\frac{N}{m}\frac{1}{(1+\delta)^2}\operatorname{Tr}\left(\boldsymbol{B}'_n\bar{\boldsymbol{Q}}'_m\bar{\boldsymbol{Q}}'^T_m\boldsymbol{B}'^T_n\right) \leq \frac{\delta - \frac{\lambda}{m}\operatorname{Tr}\big(\boldsymbol{B}'_n\bar{\boldsymbol{Q}}'_m\bar{\boldsymbol{Q}}'^T_m\big)}{1+\delta} \leq \frac{\delta}{1+\delta}.$$

To prove $\frac{\delta}{1+\delta} < 1$, it remains to show that $\delta < \infty$. With a similar proof than for Lemma A.4.1, we can show there exists a real $K_{\bar{\boldsymbol{Q}}} > 0$ such that, for all m, we have $\|\bar{\boldsymbol{Q}}_m\| \leq K_{\bar{\boldsymbol{Q}}}$, and thus

$$\delta = \frac{1}{m}\operatorname{Tr}\big(\boldsymbol{B}_n\bar{\boldsymbol{Q}}_m\big) = \frac{1}{m}\operatorname{Tr}\big((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\bar{\boldsymbol{Q}}_m\big) \leq \frac{2}{m}\operatorname{Tr}(\boldsymbol{\Phi}_{\hat{\mathcal{S}}})\|\bar{\boldsymbol{Q}}_m(\delta)\| \leq \frac{2}{m}\operatorname{Tr}(\boldsymbol{\Phi}_{\hat{\mathcal{S}}})K_{\bar{\boldsymbol{Q}}} < \infty$$

where we used for the first inequality the relation $|\operatorname{Tr}(\boldsymbol{AB})| \leq \|\boldsymbol{A}\|\operatorname{Tr}(\boldsymbol{B})$ for non-negative definite matrix $\boldsymbol{B}$. Furthermore, from equation A.4, $\frac{1}{m}\operatorname{Tr}(\boldsymbol{\Phi}_{\hat{\mathcal{S}}})$ is bounded under Assumptions 1 and 2, and both $\|\hat{\boldsymbol{U}}_n\|$ and $\|\hat{\boldsymbol{V}}_n\|$ are upper bounded by 1. We thus conclude for Z'_1 that

$$\limsup_m \frac{1}{m}\frac{N}{m}\frac{1}{(1+\delta)^2}\operatorname{Tr}\big(\boldsymbol{B}'_n\bar{\boldsymbol{Q}}'_m\bar{\boldsymbol{Q}}'^T_m\boldsymbol{B}'^T_n\big) < 1. \tag{A.22}$$

With similar arguments, we can show for Z'_2 that

$$\limsup_m \frac{1}{m}\frac{N}{m}\frac{1}{(1+\alpha)^2}\operatorname{Tr}\big(\boldsymbol{B}'_n\tilde{\boldsymbol{Q}}'_m\tilde{\boldsymbol{Q}}'^T_m\boldsymbol{B}'^T_n\big) < 1,$$

which concludes the proof that

$$|\alpha - \delta| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right). \tag{A.23}$$

Using the result above with equation A.19, we get

$$\begin{aligned}\|\tilde{\boldsymbol{Q}}_m - \bar{\boldsymbol{Q}}_m\| &= |\alpha - \delta|\left\|\frac{N}{m}\frac{1}{(1+\delta)(1+\alpha)}\tilde{\boldsymbol{Q}}_m\left(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n\right)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\bar{\boldsymbol{Q}}_m\right\| \\ &= \mathcal{O}\left(\frac{1}{\sqrt{m}}\right),\end{aligned}$$

which concludes the proof. □

Lemma A.1.3. *Under Assumptions 1 and 2, let $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ be the diagonal matrix defined in equation A.11 for which, for all $i \in [N]$, we have*

$$\boldsymbol{D}_i = 1 + \frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i. \tag{A.24}$$

Then

$$\mathbb{E}\left[\|\boldsymbol{D}\|\right] = \mathcal{O}(1).$$

Proof. Let $\alpha = \frac{1}{m}\operatorname{Tr}\big((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\mathbb{E}[\boldsymbol{Q}_-(\lambda)]\big)$ defined in equation A.2. From equation A.5, α is uniformly bounded, i.e., there exists a real $K_\alpha > 0$ such that $\alpha \leq K_\alpha$. From both Lemma A.7.4

and the union bound, we have

$$\Pr\left(\|\boldsymbol{D}\| > 1+\alpha+t\right) = \Pr\left(\max_{1\leq i\leq N} \boldsymbol{D}_i > 1+\alpha+t\right) \leq CNe^{-cm\min(t,t^2)},$$

for some $c, C > 0$ independent of m and N. Therefore,

$$\begin{aligned}
\mathbb{E}\left[\|\boldsymbol{D}\|\right] = \mathbb{E}\left[\max_{1\leq i\leq N} \boldsymbol{D}_i\right] &= \int_0^\infty \Pr\left(\max_{1\leq i\leq N} \boldsymbol{D}_i > t\right) dt \\
&= \int_0^{2(1+K_\alpha)} \Pr\left(\max_{1\leq i\leq N} \boldsymbol{D}_i > t\right) dt + \int_{2(1+K_\alpha)}^\infty \Pr\left(\max_{1\leq i\leq N} \boldsymbol{D}_i > t\right) dt \\
&\leq 2(1+K_\alpha) + \int_{2(1+K_\alpha)}^\infty CNe^{-cm\min\left((t-(1+K_\alpha))^2, t-(1+K_\alpha)\right)} dt \\
&= 2(1+K_\alpha) + \int_{1+K_\alpha}^\infty CNe^{-cmt} dt \\
&= 2(1+K_\alpha) + \frac{CN}{cm} e^{-Cm(1+K_\alpha)} \\
&= \mathcal{O}(1).
\end{aligned}$$

□

Lemma A.1.4. *Under Assumptions 1 and 2, let $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ be the diagonal matrix defined in equation A.11 for which, for all $i \in [N]$, we have*

$$\boldsymbol{D}_i = 1 + \frac{1}{m}\boldsymbol{\sigma}_i^T \hat{\boldsymbol{U}}_n \boldsymbol{Q}_{-i} (\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \boldsymbol{\sigma}_i.$$

Then

$$\left\| \mathbb{E}\left[\frac{1}{m}\boldsymbol{Q}_m (\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{D} \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m\right] \right\| = \mathcal{O}(1).$$

Proof. From Lemma A.4.4, there exists $K'_{\boldsymbol{Q}_m} > 0$ such that, for all m, we have $\left\|\frac{1}{\sqrt{m}}\boldsymbol{Q}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\right\| \leq 2K'_{\boldsymbol{Q}_m}$ and $\left\|\frac{1}{\sqrt{m}}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m\right\| \leq K'_{\boldsymbol{Q}_m}$. Therefore,

$$\left\| \mathbb{E}\left[\frac{1}{m}\boldsymbol{Q}_m (\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{D} \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m\right] \right\| \leq 2K'^2_{\boldsymbol{Q}_m} \mathbb{E}\left[\|\boldsymbol{D}\|\right].$$

From Lemma A.1.3, we have

$$\mathbb{E}\left[\|\boldsymbol{D}\|\right] = \mathcal{O}(1).$$

As a consequence, we deduce that

$$\left\| \mathbb{E}\left[\frac{1}{m}\boldsymbol{Q}_m (\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{D} \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m\right] \right\| = \mathcal{O}(1). \tag{A.25}$$

□

A.2 Proof of Theorem 7.3.2

This section is dedicated to finding an asymptotic deterministic limit of the empirical $\widehat{\text{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda)$ (equation 6.8) under Assumptions 1 and 2. We determine in Theorem 7.3.2 a deterministic limit of $\widehat{\text{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda)$ by combining Theorem 7.2.3, which provides an asymptotically more tractable approximation of $\mathbb{E}_{\boldsymbol{W}}\left[\boldsymbol{Q}_m(\lambda)\right]$ under the form of a fixed-point equation, with concentration arguments. Theorem 7.3.2 is corollary of Lemma A.2.2 and of the concentration result of Lemma A.7.2 found in Section A.7. Both Lemma A.2.4 and Lemma A.2.5 are key Lemma used in the proof of Theorem 7.3.2 and Theorem 7.4.2.

To simplify the notations, we denote the matrix $\boldsymbol{Q}_m$ as the resolvent $\boldsymbol{Q}_m(\lambda)$ (defined in equation 6.13). We define the matrix $\boldsymbol{\Psi}_{\hat{\mathcal{S}}} \in \mathbb{R}^{m\times m}$ as

$$\boldsymbol{\Psi}_{\hat{\mathcal{S}}} = \frac{N}{m}\frac{1}{1+\delta}\boldsymbol{\Phi}_{\hat{\mathcal{S}}}.$$

Furthermore, the notation $\boldsymbol{A} = \boldsymbol{B} + \mathcal{O}_{\|\cdot\|}\left(\frac{1}{\sqrt{m}}\right)$ means that $\|\boldsymbol{A}-\boldsymbol{B}\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$.

Theorem A.2.1 (Asymptotic Empirical MSBE)**.** *Under the conditions of Theorem 7.2.3, the deterministic asymptotic empirical MSBE is*

$$\widehat{\overline{\text{MSBE}}}(\hat{\boldsymbol{\theta}}_n^\lambda) = \tfrac{\lambda^2}{n}\|\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{r}\|^2 + \hat{\Delta},$$

with second-order correction factor

$$\hat{\Delta} = \tfrac{\lambda^2}{n}\frac{\frac{1}{N}\operatorname{Tr}\left(\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m(\lambda)^T\right)}{1-\frac{1}{N}\operatorname{Tr}\left(\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m(\lambda)^T\boldsymbol{\Psi}_1\right)}\|\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2,$$

where

$$\boldsymbol{\Psi}_1 = \hat{\boldsymbol{U}}_n^T\boldsymbol{\Psi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n, \quad \textit{and} \quad \boldsymbol{\Psi}_2 = (\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Psi}_{\hat{\mathcal{S}}}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n).$$

As $N, m, d \to \infty$ with asymptotic constant ratio N/m,

$$\widehat{\text{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda) - \widehat{\overline{\text{MSBE}}}(\hat{\boldsymbol{\theta}}_n^\lambda) \xrightarrow{a.s} 0.$$

Proof. We have

$$\widehat{\text{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda) = \frac{\lambda^2}{n}\|\boldsymbol{Q}_m\,\boldsymbol{r}\|^2 = \frac{\lambda^2}{n}\boldsymbol{r}^T\boldsymbol{Q}_m^T\boldsymbol{Q}_m\boldsymbol{r}.$$

From Lemma 7.3.1, we have

$$\Pr\left(\left|\frac{\lambda^2}{n}\boldsymbol{r}^T\boldsymbol{Q}_m^T\boldsymbol{Q}_m\boldsymbol{r} - \frac{\lambda^2}{n}\boldsymbol{r}^T\mathbb{E}[\boldsymbol{Q}_m^T\boldsymbol{Q}_m]\boldsymbol{r}\right| > t\right) \le Ce^{-cn^2mt^2},$$

for some $C, c > 0$ independent of m, n and N. Furthermore, from Lemma A.2.2, we have

$$\left\|\mathbb{E}[\boldsymbol{Q}_m^T\boldsymbol{Q}_m] - \bar{\boldsymbol{Q}}_m^T\bar{\boldsymbol{Q}}_m - \frac{\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\bar{\boldsymbol{Q}}_m\right)}{1-\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m\right)}\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m\right\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

As a consequence, we have

$$\widehat{\text{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda) - \widehat{\overline{\text{MSBE}}}(\hat{\boldsymbol{\theta}}_n^\lambda) \xrightarrow{a.s} 0,$$

as $m \to \infty$. □

Lemma A.2.2. *Under Assumptions 1 and 2, let $\boldsymbol{Q}_m \in \mathbb{R}^{n\times n}$ be the resolvent defined in equation 6.13, let $\bar{\boldsymbol{Q}}_m \in \mathbb{R}^{n\times n}$ be the deterministic resolvent defined in equation 7.7, and let $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ be any matrix with a bounded operator norm. Then,*

$$\left\| \mathbb{E}\left[\boldsymbol{Q}_m^T \boldsymbol{M} \boldsymbol{Q}_m\right] - \bar{\boldsymbol{Q}}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m - \frac{\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2 \bar{\boldsymbol{Q}}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m\right)}{1 - \frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2 \bar{\boldsymbol{Q}}_m^T \boldsymbol{\Psi}_1 \bar{\boldsymbol{Q}}_m\right)} \bar{\boldsymbol{Q}}_m^T \boldsymbol{\Psi}_1 \bar{\boldsymbol{Q}}_m \right\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right),$$

for $\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2 \in \mathbb{R}^{n\times n}$ defined in equation 7.12.

Proof. From Lemma A.2.6, we have

$$\begin{aligned}\mathbb{E}\left[\boldsymbol{Q}_m^T \boldsymbol{M} \boldsymbol{Q}_m\right] = \bar{\boldsymbol{Q}}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m &+ \mathbb{E}\left[\boldsymbol{Q}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\Psi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m\right] \\ &- \mathbb{E}\left[\boldsymbol{Q}_-^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\Psi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \boldsymbol{Q}_-\right] + \frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2 \bar{\boldsymbol{Q}}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m\right) \mathbb{E}\left[\boldsymbol{Q}_-^T \boldsymbol{\Psi}_1 \boldsymbol{Q}_-\right] \\ &+ \mathcal{O}_{\|\cdot\|}\left(\frac{1}{\sqrt{m}}\right).\end{aligned}$$

Let

$$\boldsymbol{M}' = \boldsymbol{M} \bar{\boldsymbol{Q}}_m (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\Psi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n = \boldsymbol{M}\left[\boldsymbol{I}_n - \lambda \bar{\boldsymbol{Q}}_m\right]$$

With a similar proof than for Lemma A.4.1, we can show that there exists a real $K_{\bar{\boldsymbol{Q}}}$ such that, for all m, we have $\|\bar{\boldsymbol{Q}}_m\| \leq K_{\bar{\boldsymbol{Q}}}$. We deduce thus that $\boldsymbol{M}'$ is a matrix with a bounded operator norm since $\|\boldsymbol{M}'\| \leq (1 + \lambda K_{\bar{\boldsymbol{Q}}})\|\boldsymbol{M}\|$. From Lemma A.2.3, we have

$$\left\| \mathbb{E}\left[\boldsymbol{Q}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\Psi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m\right] - \mathbb{E}\left[\boldsymbol{Q}_-^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\Psi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \boldsymbol{Q}_-\right] \right\| = \mathcal{O}\left(\frac{1}{m}\right).$$

Therefore,

$$\mathbb{E}\left[\boldsymbol{Q}_m^T \boldsymbol{M} \boldsymbol{Q}_m\right] = \bar{\boldsymbol{Q}}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m + \frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2 \bar{\boldsymbol{Q}}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m\right) \mathbb{E}\left[\boldsymbol{Q}_-^T \boldsymbol{\Psi}_1 \boldsymbol{Q}_-\right] + \mathcal{O}_{\|\cdot\|}\left(\frac{1}{\sqrt{m}}\right).$$

Furthermore, from Lemma A.2.4, we have

$$\left\| \mathbb{E}\left[\boldsymbol{Q}_m^T \boldsymbol{\Psi}_1 \boldsymbol{Q}_m\right] - \mathbb{E}\left[\boldsymbol{Q}_-^T \boldsymbol{\Psi}_1 \boldsymbol{Q}_-\right] \right\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right),$$

and from Lemma A.2.5 we have

$$\mathbb{E}\left[\boldsymbol{Q}_m^T \boldsymbol{\Psi}_1 \boldsymbol{Q}_m\right] = \bar{\boldsymbol{Q}}_m^T \boldsymbol{\Psi}_1 \bar{\boldsymbol{Q}}_m + \frac{1}{1 - \frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2 \bar{\boldsymbol{Q}}_m^T \boldsymbol{\Psi}_1 \bar{\boldsymbol{Q}}_m\right)} \bar{\boldsymbol{Q}}_m^T \boldsymbol{\Psi}_1 \bar{\boldsymbol{Q}}_m + \mathcal{O}_{\|\cdot\|}\left(\frac{1}{\sqrt{m}}\right).$$

We conclude thus

$$\mathbb{E}\left[\boldsymbol{Q}_m^T \boldsymbol{M} \boldsymbol{Q}_m\right] = \bar{\boldsymbol{Q}}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m + \frac{\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2 \bar{\boldsymbol{Q}}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m\right)}{1 - \frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2 \bar{\boldsymbol{Q}}_m^T \boldsymbol{\Psi}_1 \bar{\boldsymbol{Q}}_m\right)} \bar{\boldsymbol{Q}}_m^T \boldsymbol{\Psi}_1 \bar{\boldsymbol{Q}}_m + \mathcal{O}_{\|\cdot\|}\left(\frac{1}{\sqrt{m}}\right).$$

□

Lemma A.2.3. *Under Assumptions 1 and 2, let $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ be any matrix with a bounded operator norm, let $\boldsymbol{Q}_m \in \mathbb{R}^{n\times n}$ be the resolvent defined in equation 6.13, and let $\boldsymbol{Q}_- \in \mathbb{R}^{n\times n}$ be the resolvent*

defined in equation A.3. Then,

$$\left\|\mathbb{E}\left[\boldsymbol{Q}_m^T\boldsymbol{M}\boldsymbol{Q}_m\right]-\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{M}\boldsymbol{Q}_-\right]\right\|=\mathcal{O}\left(\frac{1}{m}\right).$$

Proof. We observe that

$$\begin{aligned}\left\|\mathbb{E}\left[\boldsymbol{Q}_m^T\boldsymbol{M}\boldsymbol{Q}_m\right]-\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{M}\boldsymbol{Q}_-\right]\right\| \leq& \left\|\mathbb{E}\left[\boldsymbol{Q}_m^T\boldsymbol{M}\boldsymbol{Q}_m\right]-\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{M}\boldsymbol{Q}_m\right]\right\| \\ &+\left\|\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{M}\boldsymbol{Q}_m\right]-\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{M}\boldsymbol{Q}_-\right]\right\|.\end{aligned}$$

The objective is to show that both terms vanish. By exchangeability arguments, we have

$$\begin{aligned}&\left\|\mathbb{E}\left[\boldsymbol{Q}_m^T\boldsymbol{M}\boldsymbol{Q}_m\right]-\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{M}\boldsymbol{Q}_m\right]\right\| \\ &=\left\|\frac{1}{N}\mathbb{E}\left[\sum_{i=1}^{N}[\boldsymbol{Q}_m-\boldsymbol{Q}_{-i}]^T\boldsymbol{M}\boldsymbol{Q}_m\right]\right\| \\ &=\left\|\frac{1}{N}\mathbb{E}\left[\sum_{i=1}^{N}\frac{1}{m}\boldsymbol{Q}_m^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)\boldsymbol{Q}_{-i}^T\boldsymbol{M}\boldsymbol{Q}_m\right]\right\| \quad \text{(Lemma A.8.1)} \\ &=\left\|\frac{1}{N}\mathbb{E}\left[\sum_{i=1}^{N}\frac{1}{m}\boldsymbol{Q}_m^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)\boldsymbol{Q}_m^T\boldsymbol{M}\boldsymbol{Q}_m\left(1+\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\right)\right]\right\| \\ &=\left\|\frac{1}{N}\mathbb{E}\left[\frac{1}{m}\boldsymbol{Q}_m^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{D}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)\boldsymbol{Q}_m^T\boldsymbol{M}\boldsymbol{Q}_m\right]\right\|,\end{aligned}$$

where $\boldsymbol{D}\in\mathbb{R}^{N\times N}$ is a diagonal matrix for which, for all $i\in[N]$, we have

$$\boldsymbol{D}_i=1+\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i.$$

From Lemma A.1.3, we know

$$\mathbb{E}\left[\|\boldsymbol{D}\|\right]=\mathcal{O}(1).$$

Furthermore, from Lemma A.4.4, we know there exists a real $K'_{\boldsymbol{Q}}>0$ such that, for all m, we have

$$\left\|\frac{1}{\sqrt{m}}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m\right\|\leq K'_{\boldsymbol{Q}}$$

and

$$\left\|\frac{1}{\sqrt{m}}\boldsymbol{Q}_m(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\right\|\leq 2K'_{\boldsymbol{Q}}.$$

We deduce thus

$$\begin{aligned}\left\|\mathbb{E}\left[\boldsymbol{Q}_m^T\boldsymbol{M}\boldsymbol{Q}_m\right]-\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{M}\boldsymbol{Q}_m\right]\right\|&=\left\|\frac{1}{m}\frac{m}{N}\mathbb{E}\left[\frac{1}{m}\boldsymbol{Q}_m^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{D}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)\boldsymbol{Q}_m^T\boldsymbol{M}\boldsymbol{Q}_m\right]\right\| \\ &=\mathcal{O}\left(\frac{1}{m}\right).\end{aligned}$$

With a similar reasoning, we can show that

$$\left\|\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{M}\boldsymbol{Q}_m\right]-\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{M}\boldsymbol{Q}_-\right]\right\|=\mathcal{O}\left(\frac{1}{m}\right),$$

and we conclude thus

$$\left\|\mathbb{E}\left[\boldsymbol{Q}_m^T\boldsymbol{M}\boldsymbol{Q}_m\right]-\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{M}\boldsymbol{Q}_-\right]\right\|=\mathcal{O}\left(\frac{1}{m}\right).$$

□

Lemma A.2.4. *Under Assumptions 1 and 2, let $\boldsymbol{Q}_m\in\mathbb{R}^{n\times n}$ be the resolvent defined in equation 6.13, let $\boldsymbol{Q}_-\in\mathbb{R}^{n\times n}$ be the resolvent defined in equation A.3, and let $\boldsymbol{\Psi}_1\in\mathbb{R}^{n\times n}$ be the matrix defined in equation 7.12. Then,*

$$\left\|\mathbb{E}\left[\boldsymbol{Q}_m^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\right]-\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{\Psi}_1\boldsymbol{Q}_-\right]\right\|=\mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

Proof. We observe that

$$\begin{aligned}\left\|\mathbb{E}\left[\boldsymbol{Q}_m^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\right]-\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{\Psi}_1\boldsymbol{Q}_-\right]\right\|\leq&\left\|\mathbb{E}\left[\boldsymbol{Q}_m^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\right]-\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\right]\right\|\\&+\left\|\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\right]-\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{\Psi}_1\boldsymbol{Q}_-\right]\right\|.\end{aligned}$$

The objective is to show that both terms vanish. By exchangeability arguments, we have

$$\begin{aligned}&\left\|\mathbb{E}\left[\boldsymbol{Q}_m^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\right]-\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\right]\right\|\\&=\left\|\frac{1}{N}\sum_{i=1}^N\mathbb{E}\left[\left[\boldsymbol{Q}_m-\boldsymbol{Q}_{-i}\right]^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\right]\right\|\\&=\left\|\frac{1}{N}\sum_{i=1}^N\mathbb{E}\left[\frac{1}{m}\boldsymbol{Q}_m^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)\boldsymbol{Q}_{-i}^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\right]\right\|\qquad\text{(Lemma A.8.1)}\\&=\left\|\frac{1}{N}\sum_{i=1}^N\mathbb{E}\left[\frac{1}{m}\boldsymbol{Q}_m^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)\boldsymbol{Q}_m^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\left(1+\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\right)\right]\right\|\\&=\left\|\underbrace{\frac{1}{N}\mathbb{E}\left[\frac{1}{m}\boldsymbol{Q}_m^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{D}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)\boldsymbol{Q}_m^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\right]}_{=\boldsymbol{Z}}\right\|,\end{aligned}$$

where $\boldsymbol{D}\in\mathbb{R}^{N\times N}$ is a diagonal matrix for which, for all $i\in[N]$, we have

$$\boldsymbol{D}_i=1+\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i.$$

Let the matrices

$$\boldsymbol{B}=\frac{1}{N}\frac{1}{m^{\frac{1}{4}}}\boldsymbol{Q}_m^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{D}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)\boldsymbol{Q}_m^T$$

and

$$\boldsymbol{C}^T=\frac{1}{m^{\frac{3}{4}}}\boldsymbol{\Psi}_1\boldsymbol{Q}_m.$$

We decompose $\boldsymbol{Z}$ with its symmetric and its skew-symmetric parts as

$$\boldsymbol{Z}=\mathbb{E}\left[\boldsymbol{B}\boldsymbol{C}^T\right]=\mathbb{E}\left[\frac{\boldsymbol{B}\boldsymbol{C}^T+\boldsymbol{C}\boldsymbol{B}^T}{2}\right]+\mathbb{E}\left[\frac{\boldsymbol{B}\boldsymbol{C}^T-\boldsymbol{C}\boldsymbol{B}^T}{2}\right].$$

With the same reasoning on the symmetric part and the skew-symmetric part than for equa-

tion A.15, we get for the operator norm

$$\|\boldsymbol{Z}\| \leq \|\mathbb{E}\left[\boldsymbol{B}\boldsymbol{B}^T\right]\| + \|\mathbb{E}\left[\boldsymbol{C}\boldsymbol{C}^T\right]\|.$$

We want to show that both $\|\mathbb{E}\left[\boldsymbol{B}\boldsymbol{B}^T\right]\|$ and $\|\mathbb{E}\left[\boldsymbol{C}\boldsymbol{C}^T\right]\|$ vanish. We have

$$\mathbb{E}\left[\boldsymbol{B}\boldsymbol{B}^T\right] = \mathbb{E}\left[\frac{m^2}{N^2}\frac{1}{m^2\sqrt{m}}\boldsymbol{Q}_m^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{D}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\boldsymbol{Q}_m^T\boldsymbol{Q}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{D}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m\right].$$

From Lemma A.1.3, we know

$$\mathbb{E}\left[\|\boldsymbol{D}\|\right] = \mathcal{O}(1).$$

Furthermore, from Lemma A.4.4, we know there exists a real $K'_{\boldsymbol{Q}} > 0$ such that, for all m, we have

$$\left\|\frac{1}{\sqrt{m}}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m\right\| \leq K'_{\boldsymbol{Q}}$$

and

$$\left\|\frac{1}{\sqrt{m}}\boldsymbol{Q}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\right\| \leq 2K'_{\boldsymbol{Q}}.$$

We have therefore

$$\|\mathbb{E}\left[\boldsymbol{B}\boldsymbol{B}^T\right]\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

For $\mathbb{E}\left[\boldsymbol{C}\boldsymbol{C}^T\right]$, we have

$$\mathbb{E}\left[\boldsymbol{C}\boldsymbol{C}^T\right] = \mathbb{E}\left[\frac{1}{m\sqrt{m}}\boldsymbol{Q}_m^T\boldsymbol{\Psi}_1^2\boldsymbol{Q}_m\right].$$

Let $\boldsymbol{\sigma}_{N+1}$ and $\boldsymbol{\sigma}_{N+2}$ be independent vectors with the same law as $\boldsymbol{\sigma}_i$, we have

$$\mathbb{E}\left[\frac{1}{m\sqrt{m}}\boldsymbol{Q}_m^T\boldsymbol{\Psi}_1\boldsymbol{\Psi}_1\boldsymbol{Q}_m\right] = \mathbb{E}\left[\frac{1}{m\sqrt{m}}\frac{N^2}{m^2}\frac{1}{(1+\delta)^2}\boldsymbol{Q}_m^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_{N+1}\boldsymbol{\sigma}_{N+1}^T\hat{\boldsymbol{U}}_n\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_{N+2}\boldsymbol{\sigma}_{N+2}^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m\right].$$

Let

$$\boldsymbol{B}' = \frac{1}{m^{\frac{3}{4}}}\frac{N}{m}\frac{1}{1+\delta}\boldsymbol{Q}_m^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_{N+1}\boldsymbol{\sigma}_{N+1}^T\hat{\boldsymbol{U}}_n$$

and

$$\boldsymbol{C}'^T = \frac{1}{m^{\frac{3}{4}}}\frac{N}{m}\frac{1}{1+\delta}\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_{N+2}\boldsymbol{\sigma}_{N+2}^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m.$$

We decompose $\mathbb{E}\left[\boldsymbol{C}\boldsymbol{C}^T\right]$ with its symmetric and its skew-symmetric parts as

$$\mathbb{E}\left[\boldsymbol{C}\boldsymbol{C}^T\right] = \mathbb{E}[\boldsymbol{B}'\boldsymbol{C}'^T] = \mathbb{E}\left[\frac{\boldsymbol{B}'\boldsymbol{C}'^T + \boldsymbol{C}'\boldsymbol{B}'^T}{2}\right] + \mathbb{E}\left[\frac{\boldsymbol{B}'\boldsymbol{C}'^T - \boldsymbol{C}'\boldsymbol{B}'^T}{2}\right],$$

and we get for the operator norm

$$\|\mathbb{E}\left[\boldsymbol{C}\boldsymbol{C}^T\right]\| \leq \|\mathbb{E}\left[\boldsymbol{B}'\boldsymbol{B}'^T\right]\| + \|\mathbb{E}\left[\boldsymbol{C}'\boldsymbol{C}'^T\right]\|.$$

To prove $\|\mathbb{E}\left[\boldsymbol{C}\boldsymbol{C}^T\right]\|$ vanish, we prove both $\|\mathbb{E}\left[\boldsymbol{B}'\boldsymbol{B}'^T\right]\|$ and $\|\mathbb{E}\left[\boldsymbol{C}'\boldsymbol{C}'^T\right]\|$ vanish. Let $K = \frac{1}{(1+\delta)^2}\frac{N}{N+1}\frac{N}{m}$, we write $\mathbb{E}\left[\boldsymbol{B}'\boldsymbol{B}'^T\right]$ as

$$\mathbb{E}\left[\boldsymbol{B}'\boldsymbol{B}'^T\right] = \mathbb{E}\left[\frac{1}{m\sqrt{m}}\frac{N^2}{m^2}\frac{1}{(1+\delta)^2}\boldsymbol{Q}_m^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_{N+1}\boldsymbol{\sigma}_{N+1}^T\hat{\boldsymbol{U}}_n\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_{N+1}\boldsymbol{\sigma}_{N+1}^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m\right]$$

$$
\begin{aligned}
&= \mathbb{E}\left[\frac{1}{m\sqrt{m}}\frac{N^2}{m^2}\frac{1}{(1+\delta)^2}\boldsymbol{Q}_{-N-1}^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_{N+1}\boldsymbol{\sigma}_{N+1}^T\hat{\boldsymbol{U}}_n\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_{N+1}\boldsymbol{\sigma}_{N+1}^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-N-1}\right] \\
&= \mathbb{E}\left[K\frac{1}{m\sqrt{m}}\sum_{i=1}^{N+1}\boldsymbol{Q}_{-i}^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}\left(\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_i\right)\right] \\
&= \mathbb{E}\left[K\frac{1}{m\sqrt{m}}\sum_{i=1}^{N+1}\boldsymbol{Q}_{-i}^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}\frac{1}{m}\operatorname{Tr}\big(\hat{\boldsymbol{U}}_n\hat{\boldsymbol{U}}_n^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\big)\right] \\
&+ \mathbb{E}\left[K\frac{1}{m\sqrt{m}}\sum_{i=1}^{N+1}\boldsymbol{Q}_{-i}^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}\left(\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_i - \frac{1}{m}\operatorname{Tr}\big(\hat{\boldsymbol{U}}_n\hat{\boldsymbol{U}}_n^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\big)\right)\right] \\
&= \underbrace{\mathbb{E}\left[K\frac{1}{m}\operatorname{Tr}\big(\hat{\boldsymbol{U}}_n\hat{\boldsymbol{U}}_n^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\big)\frac{1}{m\sqrt{m}}\boldsymbol{Q}_m^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{D}^2\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m\right]}_{=\boldsymbol{Z}_1} \\
&+ \underbrace{\mathbb{E}\left[K\frac{1}{m}\operatorname{Tr}\big(\hat{\boldsymbol{U}}_n\hat{\boldsymbol{U}}_n^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\big)\frac{1}{m\sqrt{m}}\boldsymbol{Q}_m^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{D}^2\boldsymbol{D}'\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m\right]}_{=\boldsymbol{Z}_2},
\end{aligned}
$$

where $\boldsymbol{D}' \in \mathbb{R}^{N\times N}$ is a diagonal matrices for which, for all $i \in [N]$, we have

$$
\boldsymbol{D}'_i = \frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_i - \frac{1}{m}\operatorname{Tr}\big(\hat{\boldsymbol{U}}_n\hat{\boldsymbol{U}}_n^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\big).
$$

From Lemma A.1.3, from Lemma A.4.4, and from equation A.4, we have

$$
\|\boldsymbol{Z}_1\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).
$$

From Lemma A.7.2, we have

$$
\mathbb{E}\big[\|\boldsymbol{D}'\|\big] = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).
$$

and thus

$$
\|\boldsymbol{Z}_2\| = \mathcal{O}\left(\frac{1}{m}\right).
$$

We conclude that

$$
\left\|\mathbb{E}\left[\boldsymbol{B}'\boldsymbol{B}'^T\right]\right\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right)
$$

and

$$
\left\|\mathbb{E}\left[\boldsymbol{C}'\boldsymbol{C}'^T\right]\right\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).
$$

Therefore,

$$
\left\|\mathbb{E}\left[\boldsymbol{C}\boldsymbol{C}^T\right]\right\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right)
$$

and

$$
\left\|\mathbb{E}\left[\boldsymbol{Q}_m^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m\right] - \mathbb{E}\left[\boldsymbol{Q}_-^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m\right]\right\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).
$$

With a similar reasoning, we can show

$$
\left\|\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\right] - \mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{\Psi}_1\boldsymbol{Q}_-\right]\right\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).
$$

We conclude thus

$$\left\|\mathbb{E}\left[\boldsymbol{Q}_m^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\right]-\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{\Psi}_1\boldsymbol{Q}_-\right]\right\|=\mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

□

Lemma A.2.5. *Under Assumptions 1 and 2, let $\boldsymbol{Q}_m\in\mathbb{R}^{n\times n}$ be the resolvent defined in equation 6.13, let $\bar{\boldsymbol{Q}}_m\in\mathbb{R}^{n\times n}$ be the deterministic resolvent defined in equation 7.7, let $\boldsymbol{\Psi}_1,\boldsymbol{\Psi}_2\in\mathbb{R}^{n\times n}$ be the matrices defined in equation 7.12. Then,*

$$\left\|\mathbb{E}\big[\boldsymbol{Q}_m^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\big]-\frac{1}{1-\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m\right)}\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m\right\|=\mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

Proof. From Lemma A.2.6, we know that

$$\begin{aligned}\mathbb{E}\big[\boldsymbol{Q}_m^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\big]&=\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m+\mathbb{E}\left[\boldsymbol{Q}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Psi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m\right]\\&\quad-\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Psi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_-\right]+\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m\right)\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{\Psi}_1\boldsymbol{Q}_-\right]\\&\quad+\mathcal{O}_{\|\cdot\|}\left(\frac{1}{\sqrt{m}}\right).\end{aligned}$$

Exploiting $\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Psi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n=\boldsymbol{I}_n-\lambda\bar{\boldsymbol{Q}}_n$ in the above equation, and from Lemma A.2.4, we obtain the simplification

$$\mathbb{E}\big[\boldsymbol{Q}_m^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\big]=\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m+\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m\right)\mathbb{E}\left[\boldsymbol{Q}_m^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\right]+\mathcal{O}_{\|\cdot\|}\left(\frac{1}{\sqrt{m}}\right).$$

or equivalently

$$\mathbb{E}\big[\boldsymbol{Q}_m^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\big]\left(1-\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m\right)\mathbb{E}\left[\boldsymbol{Q}_m^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\right]\right)=\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m+\mathcal{O}_{\|\cdot\|}\left(\frac{1}{\sqrt{m}}\right).$$

Let $\boldsymbol{B}_n'=\bar{\boldsymbol{Z}}^T\hat{\boldsymbol{A}}_m\boldsymbol{Z}$ and $\bar{\boldsymbol{Q}}_m'=\left[\frac{N}{m}\frac{1}{1+\delta}\boldsymbol{B}_n'+\lambda\boldsymbol{I}_m\right]^{-1}$, for which $\hat{\boldsymbol{A}}_m=\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T$ is the empirical transition model matrix (equation 6.14) and $\bar{\boldsymbol{Z}}\bar{\boldsymbol{Z}}^T=\boldsymbol{\Phi}_{\hat{\mathcal{S}}}$ is the Cholesky decompositon of $\boldsymbol{\Phi}_{\hat{\mathcal{S}}}$. We have from the cyclic properties of the trace

$$\begin{aligned}&\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m\right)\\&=\frac{1}{m}\frac{N}{m}\frac{1}{(1+\delta)^2}\operatorname{Tr}\big((\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)\bar{\boldsymbol{Q}}_m^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\bar{\boldsymbol{Q}}_m\big)\\&=\frac{1}{m}\frac{N}{m}\frac{1}{(1+\delta)^2}\operatorname{Tr}\big(\boldsymbol{B}_n'\boldsymbol{Q}_m'\boldsymbol{Q}_m'^T\boldsymbol{B}_n'^T\big).\end{aligned}$$

From equation A.22, we have

$$\limsup_m\frac{1}{m}\frac{N}{m}\frac{1}{(1+\delta)^2}\operatorname{Tr}\big(\boldsymbol{B}_n'\bar{\boldsymbol{Q}}_m'\bar{\boldsymbol{Q}}_m'^T\boldsymbol{B}_n'^T\big)<1.$$

Therefore,

$$\mathbb{E}\big[\boldsymbol{Q}_m^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\big]=\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m+\frac{1}{1-\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m\right)}\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m+\mathcal{O}_{\|\cdot\|}\left(\frac{1}{\sqrt{m}}\right).$$

□

Lemma A.2.6. *Under Assumptions 1 and 2, let $\boldsymbol{Q}_m \in \mathbb{R}^{n\times n}$ be the resolvent defined in equation 6.13, let $\boldsymbol{Q}_- \in \mathbb{R}^{n\times n}$ be the resolvent defined in equation A.3, let $\bar{\boldsymbol{Q}}_m \in \mathbb{R}^{n\times n}$ be the deterministic resolvent defined in equation 7.7, let $\hat{\boldsymbol{U}}_n, \hat{\boldsymbol{V}}_n \in \mathbb{R}^{m\times n}$ be the shift matrices defined in equation 6.7, and let $\boldsymbol{M}$ be either any matrix with a bounded operator norm or $\boldsymbol{M} = \boldsymbol{\Psi}_1$. Then,*

$$\begin{aligned}
&\Big\| \mathbb{E}\big[\boldsymbol{Q}_m^T \boldsymbol{M} \boldsymbol{Q}_m\big] - \bar{\boldsymbol{Q}}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m - \mathbb{E}\left[\boldsymbol{Q}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\Psi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m\right] \\
&\quad + \mathbb{E}\left[\boldsymbol{Q}_-^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\Psi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \boldsymbol{Q}_-\right] - \frac{1}{N} \operatorname{Tr}\left(\boldsymbol{\Psi}_2 \bar{\boldsymbol{Q}}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m\right) \mathbb{E}\left[\boldsymbol{Q}_-^T \boldsymbol{\Psi}_1 \boldsymbol{Q}_-\right] \Big\| \\
&= \mathcal{O}\left(\frac{1}{\sqrt{m}}\right),
\end{aligned}$$

for $\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2 \in \mathbb{R}^{n\times n}$ defined in equation 7.12.

Proof. With the resolvent identity (Lemma A.8.1), we decompose $\mathbb{E}\big[\boldsymbol{Q}_m^T \boldsymbol{M} \boldsymbol{Q}_m\big]$ as

$$\begin{aligned}
\mathbb{E}\big[\boldsymbol{Q}_m^T \boldsymbol{M} \boldsymbol{Q}_m\big] &= \mathbb{E}\big[\boldsymbol{Q}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m\big] - \mathbb{E}\big[\boldsymbol{Q}_m^T \boldsymbol{M} [\bar{\boldsymbol{Q}}_m - \boldsymbol{Q}_m]\big] && \text{(A.26)} \\
&= \mathbb{E}\big[\boldsymbol{Q}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m\big] \\
&\quad - \mathbb{E}\left[\boldsymbol{Q}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m \left[\frac{1}{m} (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n - (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\Psi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n\right] \boldsymbol{Q}_m\right] \\
&= \underbrace{\mathbb{E}\big[\boldsymbol{Q}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m\big]}_{=\boldsymbol{Z}_1} + \mathbb{E}\left[\boldsymbol{Q}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\Psi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m\right] && \text{(A.27)} \\
&\quad \underbrace{- \frac{1}{m} \sum_{i=1}^{N} \mathbb{E}\left[\boldsymbol{Q}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m\right]}_{=\boldsymbol{Z}_2},
\end{aligned}$$

where $\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} = \sum_{i=1}^N \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T$ is the same decompositon of $\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}$ than the one used in equation A.6. From Theorem 7.2.3, we have

$$\big\| \mathbb{E}[\boldsymbol{Q}_m] - \bar{\boldsymbol{Q}}_m \big\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

Therefore, from above and from Lemma A.2.9 which upper bounds $\|\boldsymbol{M} \bar{\boldsymbol{Q}}_m\|$, we deduce for $\boldsymbol{Z}_1$ that

$$\begin{aligned}
\|\boldsymbol{Z}_1\| - \big\|\bar{\boldsymbol{Q}}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m\big\| &= \big\|\mathbb{E}\big[\boldsymbol{Q}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m\big]\big\| - \big\|\bar{\boldsymbol{Q}}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m\big\| \\
&\leq \big\|\mathbb{E}[\boldsymbol{Q}_m] - \bar{\boldsymbol{Q}}_m\big\| \, \|\boldsymbol{M} \bar{\boldsymbol{Q}}_m\| \\
&= \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).
\end{aligned}$$

We want to find now a deterministic approximation for $\boldsymbol{Z}_2$ in equation A.27. From the Sherman identity (Lemma A.8.3) and with the resolvent $\boldsymbol{Q}_{-i}$ defined in equation A.7 as

$$\boldsymbol{Q}_{-i} = \left[\frac{1}{m} (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n - \frac{1}{m} (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \hat{\boldsymbol{U}}_n + \lambda \boldsymbol{I}_n\right]^{-1},$$

we obtain the following relation

$$\boldsymbol{Q}_m = \boldsymbol{Q}_{-i} - \frac{\frac{1}{m}\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}}{1+\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i}.$$

By remarking that for all $i \in [N]$, we have

$$\begin{aligned}
&\boldsymbol{Q}_m^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m \\
&= \boldsymbol{Q}_m^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}\frac{1}{1+\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i} \\
&= \frac{1}{1+\delta}\boldsymbol{Q}_m^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i} \\
&+ \frac{1}{1+\delta}\boldsymbol{Q}_m^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}\frac{\delta - \frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i}{1+\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i},
\end{aligned}$$

we decompose $\boldsymbol{Z}_2$ as

$$\boldsymbol{Z}_2 = \frac{1}{m}\mathbb{E}\left[\sum_{i=1}^N \boldsymbol{Q}_m^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m\right] \tag{A.28}$$

$$= \underbrace{\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Psi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_-\right]}_{=\boldsymbol{Z}_{21}} \tag{A.29}$$

$$- \underbrace{\frac{1}{m}\frac{1}{1+\delta}\sum_{i=1}^N \mathbb{E}\left[\boldsymbol{Q}_{-i}^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}\frac{\frac{1}{m}\boldsymbol{\sigma}_i^T(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\boldsymbol{Q}_{-i}^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i}{1+\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i}\right]}_{=\boldsymbol{Z}_{22}} \tag{A.30}$$

$$+ \underbrace{\frac{1}{m}\frac{1}{1+\delta}\sum_{i=1}^N \mathbb{E}\left[\boldsymbol{Q}_{-i}^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}\frac{\delta - \frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i}{1+\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i}\right]}_{=\boldsymbol{Z}_{23}} \tag{A.31}$$

$$- \underbrace{\frac{1}{m}\frac{1}{1+\delta}\sum_{i=1}^N \mathbb{E}\Bigg[\boldsymbol{Q}_{-i}^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i} \frac{\left(\frac{1}{m}\boldsymbol{\sigma}_i^T(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\boldsymbol{Q}_{-i}^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\right)\left(\delta - \frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\right)}{\left(1+\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\right)^2}\Bigg]}_{=\boldsymbol{Z}_{24}} \tag{A.32}$$

$$= \boldsymbol{Z}_{21} - \boldsymbol{Z}_{22} + \boldsymbol{Z}_{23} - \boldsymbol{Z}_{24}. \tag{A.33}$$

From Lemma A.2.7, we have

$$\left\|\boldsymbol{Z}_{22} - \frac{1}{N}\operatorname{Tr}(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m)\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{\Psi}_1\boldsymbol{Q}_-\right]\right\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

With a similar proof than for $\boldsymbol{Z}_{22}$, we can show for $\boldsymbol{Z}_{24}$ that

$$\|\boldsymbol{Z}_{24}\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

From Lemma A.2.8, we have

$$\|\boldsymbol{Z}_{23}\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

As a consequence, we conclude that

$$\begin{aligned}\mathbb{E}\big[\boldsymbol{Q}_m^T\boldsymbol{M}\boldsymbol{Q}_m\big] &= \bar{\boldsymbol{Q}}_m^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m + \mathbb{E}\left[\boldsymbol{Q}_m^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Psi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m\right] \\ &\quad - \mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Psi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_-\right] + \frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m\right)\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{\Psi}_1\boldsymbol{Q}_-\right] \\ &\quad + \mathcal{O}_{\|\cdot\|}\left(\frac{1}{\sqrt{m}}\right).\end{aligned}$$

□

Lemma A.2.7. *Under Assumptions 1 and 2, let $\boldsymbol{Z}_{22} \in \mathbb{R}^{n\times n}$ be the matrix defined in equation A.30 as*

$$\boldsymbol{Z}_{22} = \frac{1}{m}\frac{1}{1+\delta}\sum_{i=1}^{N}\mathbb{E}\left[\boldsymbol{Q}_{-i}^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}\frac{\frac{1}{m}\boldsymbol{\sigma}_i^T(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\boldsymbol{Q}_{-i}^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i}{1 + \frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i}\right].$$

Then,

$$\left\|\boldsymbol{Z}_{22} - \frac{1}{N}\operatorname{Tr}\big(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m\big)\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{\Psi}_1\boldsymbol{Q}_-\right]\right\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right),$$

where $\bar{\boldsymbol{Q}}_m \in \mathbb{R}^{n\times n}$ is the deterministic resolvent defined in equation 7.7, $\boldsymbol{Q}_- \in \mathbb{R}^{n\times n}$ is the resolvent defined in equation A.3, and $\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2 \in \mathbb{R}^{n\times n}$ defined in equation 7.12.

Proof. Let $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ be a diagonal matrix for which, for all $i \in [N]$, we have

$$\boldsymbol{D}_i = 1 + \frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i,$$

and $\boldsymbol{D}_2 \in \mathbb{R}^{N\times N}$ be another diagonal matrix for which, for all $i \in [N]$, we have

$$\begin{aligned}[\boldsymbol{D}_2]_i &= \frac{\frac{1}{m}\boldsymbol{\sigma}_i^T(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\boldsymbol{Q}_{-i}^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i}{1 + \frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i} - \frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m\right) \\ &= \frac{\frac{1}{m}\boldsymbol{\sigma}_i^T(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\boldsymbol{Q}_{-i}^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i}{1 + \frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i} \\ &\quad - \frac{\frac{1}{m}\operatorname{Tr}\left((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\bar{\boldsymbol{Q}}_m^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m\right)}{1+\delta} \\ &= \frac{\frac{1}{m}\boldsymbol{\sigma}_i^T(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\boldsymbol{Q}_{-i}^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i}{1 + \frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i} \\ &\quad - \frac{\frac{1}{m}\operatorname{Tr}\left((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\bar{\boldsymbol{Q}}_m^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\right)}{1+\delta}.\end{aligned}$$

We have

$$\left\|\boldsymbol{Z}_{22} - \frac{1}{N}\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{\Psi}_1\boldsymbol{Q}_-\right]\operatorname{Tr}\big(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m\big)\right\|$$

$$
\begin{aligned}
&= \left\| \boldsymbol{Z}_{22} - \frac{1}{m}\frac{1}{1+\delta}\mathbb{E}\left[\sum_{i=1}^{N} \boldsymbol{Q}_{-i}^T \hat{\boldsymbol{U}}_n^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \hat{\boldsymbol{U}}_n \boldsymbol{Q}_{-i}\right] \frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2 \bar{\boldsymbol{Q}}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m\right) \right\| \\
&= \left\| \frac{1}{m}\frac{1}{1+\delta}\mathbb{E}\left[\sum_{i=1}^{N} \boldsymbol{Q}_{-i}^T \hat{\boldsymbol{U}}_n^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \hat{\boldsymbol{U}}_n \boldsymbol{Q}_{-i} \left(\frac{\frac{1}{m}\boldsymbol{\sigma}_i^T (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n) \boldsymbol{Q}_{-i}^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\sigma}_i}{1 + \frac{1}{m}\boldsymbol{\sigma}_i^T \hat{\boldsymbol{U}}_n \boldsymbol{Q}_{-i} (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\sigma}_i} \right.\right.\right. \\
&\quad \left.\left.\left. - \frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2 \bar{\boldsymbol{Q}}_m^T \boldsymbol{M} \bar{\boldsymbol{Q}}_m\right) \right) \right] \right\| \\
&= \Bigg\| \underbrace{\frac{1}{m}\frac{1}{1+\delta}\mathbb{E}\left[\boldsymbol{Q}_m^T \hat{\boldsymbol{U}}_n^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{D}^2 \boldsymbol{D}_2 \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m\right]}_{=\boldsymbol{Z}_{221}} \Bigg\|.
\end{aligned}
$$

Let the matrices

$$\boldsymbol{B} = m^{-\frac{1}{4}} \frac{1}{\sqrt{m}} \boldsymbol{Q}_m^T \hat{\boldsymbol{U}}_n^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{D}^2,$$

and

$$\boldsymbol{C}^T = m^{\frac{1}{4}} \frac{1}{\sqrt{m}} \boldsymbol{D}_2 \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m.$$

Using the matrices above, we have

$$\boldsymbol{Z}_{221} = \frac{1}{m}\frac{1}{1+\delta}\mathbb{E}\left[\boldsymbol{Q}_m^T \hat{\boldsymbol{U}}_n^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{D}^2 \boldsymbol{D}_2 \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m\right] = \frac{1}{1+\delta}\mathbb{E}\left[\boldsymbol{B}\boldsymbol{C}^T\right] = \frac{1}{1+\delta}\mathbb{E}\left[\frac{\boldsymbol{B}\boldsymbol{C}^T + \boldsymbol{C}\boldsymbol{B}^T}{2}\right],$$

since $\boldsymbol{Z}_{221}$ is symmetric. We use the relations $(\boldsymbol{B}-\boldsymbol{C})(\boldsymbol{B}-\boldsymbol{C})^T \succeq 0$ and $(\boldsymbol{B}+\boldsymbol{C})(\boldsymbol{B}+\boldsymbol{C})^T \succeq 0$ to deduce the following relation

$$-\boldsymbol{B}\boldsymbol{B}^T - \boldsymbol{C}\boldsymbol{C}^T \preceq \boldsymbol{B}\boldsymbol{C}^T + \boldsymbol{C}\boldsymbol{B}^T \preceq \boldsymbol{B}\boldsymbol{B}^T + \boldsymbol{C}\boldsymbol{C}^T.$$

From this relation, we obtain

$$\|\boldsymbol{Z}_{221}\| \leq \frac{1}{2(1+\delta)}\left(\mathbb{E}\left[\left\|\boldsymbol{B}\boldsymbol{B}^T\right\|\right] + \mathbb{E}\left[\left\|\boldsymbol{C}\boldsymbol{C}^T\right\|\right]\right),$$

where

$$\boldsymbol{B}\boldsymbol{B}^T = \frac{1}{m\sqrt{m}} \boldsymbol{Q}_m^T \hat{\boldsymbol{U}}_n^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{D}^4 \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m$$

and

$$\boldsymbol{C}\boldsymbol{C}^T = \frac{1}{\sqrt{m}} \boldsymbol{Q}_m^T \hat{\boldsymbol{U}}_n^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{D}_2^2 \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m.$$

To get the Lemma, we prove that both $\mathbb{E}\left[\left\|\boldsymbol{B}\boldsymbol{B}^T\right\|\right]$ and $\mathbb{E}\left[\left\|\boldsymbol{C}\boldsymbol{C}^T\right\|\right]$ vanish. From Lemma A.4.4, we know there exists a real $K'_{\boldsymbol{Q}} > 0$ such that, for all m, we have

$$\left\| \frac{1}{\sqrt{m}} \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m \right\| \leq K'_{\boldsymbol{Q}}$$

and

$$\left\| \frac{1}{\sqrt{m}} \boldsymbol{Q}_m (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \right\| \leq 2K'_{\boldsymbol{Q}}.$$

Furthermore, from Lemma A.1.3, we know

$$\mathbb{E}\left[\|\boldsymbol{D}^4\|\right] = \mathcal{O}(1).$$

We conclude that

$$\mathbb{E}\big[\|\boldsymbol{B}\boldsymbol{B}^T\|\big] = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

For $\mathbb{E}\big[\|\boldsymbol{C}\boldsymbol{C}^T\|\big]$, we remark that

$$\begin{aligned}\Pr\big([\boldsymbol{D}_2]_i \geq t\big) \leq \Pr\Bigg(& \frac{\frac{1}{m}\boldsymbol{\sigma}_i^T(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\boldsymbol{Q}_{-i}^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i}{1 + \frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i} \\ & - \frac{\frac{1}{m}\operatorname{Tr}\big(\boldsymbol{\Phi}_{\hat{\mathcal{S}}}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\bar{\boldsymbol{Q}}_m^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\big)}{1 + \frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i} \geq \frac{t}{2}\Bigg) \\ & + \Pr\Bigg(\frac{\frac{1}{m}\operatorname{Tr}\big(\boldsymbol{\Phi}_{\hat{\mathcal{S}}}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\bar{\boldsymbol{Q}}_m^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\big)}{1 + \frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i} \\ & - \frac{\frac{1}{m}\operatorname{Tr}\big(\boldsymbol{\Phi}_{\hat{\mathcal{S}}}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\bar{\boldsymbol{Q}}_m^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\big)}{1 + \delta} \geq \frac{t}{2}\Bigg).\end{aligned}$$

Since $\|\boldsymbol{M}\bar{\boldsymbol{Q}}_m\|$ is bounded from Lemma A.2.9, with a similar proof than for Lemma A.7.4, we can prove that

$$\begin{aligned}\Pr\Bigg(& \bigg|\frac{1}{m}\boldsymbol{\sigma}^T(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\boldsymbol{Q}_{-i}^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma} \\ & - \frac{1}{m}\operatorname{Tr}\big((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\mathbb{E}[\boldsymbol{Q}_{-i}^T]\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\big)\bigg| > t\Bigg) \leq Ce^{-cm\max(t,t^2)},\end{aligned}$$

for some C, c independent of N, m. Besides, from the proof of Theorem 7.2.3, we also have

$$\begin{aligned}\bigg|\frac{1}{m}\operatorname{Tr}\big((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\mathbb{E}[\boldsymbol{Q}_{-i}^T]\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\big) \\ - \frac{1}{m}\operatorname{Tr}\big((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\bar{\boldsymbol{Q}}_m^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\big)\bigg| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right),\end{aligned}$$

as both $\|\hat{\boldsymbol{U}}_n\|$ and $\|\hat{\boldsymbol{V}}_n\|$ are upper bounded by 1, $|\operatorname{Tr}(\boldsymbol{A}\boldsymbol{B})| \leq \|\boldsymbol{A}\|\operatorname{Tr}(\boldsymbol{B})$ for non-negative definite matrix $\boldsymbol{B}$, and from equation A.4 that bounds $\frac{1}{m}\operatorname{Tr}(\boldsymbol{\Phi}_{\hat{\mathcal{S}}})$. From Lemma A.7.4, we have

$$\Pr\left(\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i - \alpha > t\right) \leq C'e^{-mc'\max(t,t^2)},$$

for some C', c' independent of N, m. From equation A.23 in the proof of Theorem 7.2.3, we have

$$|\alpha - \delta| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

Combining all the results above, we deduce that

$$\mathbb{E}\left(\|\boldsymbol{D}_2\|^2\right) = \mathbb{E}\left(\max_{1\leq i\leq N}[\boldsymbol{D}_2^2]_i\right) = \int_0^\infty \Pr\left(\max_{1\leq i\leq N}[\boldsymbol{D}_2^2]_i > t\right)dt = \mathcal{O}\left(\frac{1}{m}\right),$$

and therefore

$$\mathbb{E}\big[\|\boldsymbol{C}\boldsymbol{C}^T\|\big] = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

□

Lemma A.2.8. *Under Assumptions 1 and 2, let $\boldsymbol{Z}_{23} \in \mathbb{R}^{n\times n}$ be the matrix defined in equa-*

tion A.31 as

$$\boldsymbol{Z}_{23} = \frac{1}{m}\frac{1}{1+\delta}\sum_{i=1}^{N}\mathbb{E}\left[\boldsymbol{Q}_{-i}^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}\frac{\delta-\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i}{1+\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i}\right].$$

Then,

$$\|\boldsymbol{Z}_{23}\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

Proof. Let the matrices

$$\boldsymbol{B}_i = m^{-\frac{1}{4}}\frac{1}{\sqrt{m}}\boldsymbol{Q}_{-i}^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i$$

and

$$\boldsymbol{C}_i^T = m^{\frac{1}{4}}\frac{1}{\sqrt{m}}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}\frac{\delta-\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i}{1+\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i}.$$

We decompose $\boldsymbol{Z}_{23}$ with its symmetric and skew-symmetric parts as

$$\begin{aligned}\boldsymbol{Z}_{23} &= \frac{1}{1+\delta}\sum_{i=1}^{N}\mathbb{E}\left[\boldsymbol{B}_i\boldsymbol{C}_i^T\right]\\ &= \frac{1}{1+\delta}\sum_{i=1}^{N}\mathbb{E}\left[\frac{\boldsymbol{B}_i\boldsymbol{C}_i^T+\boldsymbol{C}_i\boldsymbol{B}_i^T}{2}\right]+\frac{1}{1+\delta}\mathbb{E}\left[\sum_{i=1}^{N}\frac{\boldsymbol{B}_i\boldsymbol{C}_i^T-\boldsymbol{C}_i\boldsymbol{B}_i^T}{2}\right].\end{aligned}$$

With the same reasoning on the symmetric part and the skew-symmetric part than for equation A.15, we get the operator norm

$$\|\boldsymbol{Z}_{23}\| \leq \frac{1}{1+\delta}\left\|\mathbb{E}\left[\sum_{i=1}^{N}\boldsymbol{B}_i\boldsymbol{B}_i^T\right]\right\|+\frac{1}{1+\delta}\left\|\mathbb{E}\left[\sum_{i=1}^{N}\boldsymbol{C}_i\boldsymbol{C}_i^T\right]\right\|.$$

We want to show that both $\left\|\mathbb{E}\left[\sum_{i=1}^{N}\boldsymbol{B}_i\boldsymbol{B}_i^T\right]\right\|$ and $\left\|\mathbb{E}\left[\sum_{i=1}^{N}\boldsymbol{C}_i\boldsymbol{C}_i^T\right]\right\|$ vanish. We write $\mathbb{E}\left[\sum_{i=1}^{N}\boldsymbol{C}_i\boldsymbol{C}_i^T\right]$ as

$$\begin{aligned}\mathbb{E}\left[\sum_{i=1}^{N}\boldsymbol{C}_i\boldsymbol{C}_i^T\right] &= \mathbb{E}\left[\sum_{i=1}^{N}\frac{1}{\sqrt{m}}\boldsymbol{Q}_{-i}^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}\frac{\left(\delta-\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\right)^2}{\left(1+\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\right)^2}\right]\\ &= \mathbb{E}\left[\sum_{i=1}^{N}\frac{1}{\sqrt{m}}\boldsymbol{Q}_m^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m\left(\delta-\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\right)^2\right]\\ &= \mathbb{E}\left[\frac{1}{\sqrt{m}}\boldsymbol{Q}_m^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{D}_3^2\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m\right],\end{aligned}$$

where $\boldsymbol{D}_3 \in \mathbb{R}^{N\times N}$ is a diagonal matrix for which, for all $i \in [N]$, we have

$$[\boldsymbol{D}_3]_i = \delta-\frac{1}{m}\boldsymbol{\sigma}_i^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i.$$

With a similar proof than for Lemma A.1.3, and from Lemma A.1.1 and Theorem 7.2.3, we find

that

$$\mathbb{E}\left(\|\boldsymbol{D}_3\|^2\right) = \mathcal{O}\left(\frac{1}{m}\right).$$

From Lemma A.4.4, we know there exists a real $K'_{\boldsymbol{Q}} > 0$ such that, for all m, we have

$$\left\|\frac{1}{\sqrt{m}}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m\right\| \leq K'_{\boldsymbol{Q}}$$

and

$$\left\|\frac{1}{\sqrt{m}}\boldsymbol{Q}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\right\| \leq 2K'_{\boldsymbol{Q}}.$$

We deduce thus

$$\left\|\mathbb{E}\left[\sum_{i=1}^N \boldsymbol{C}_i\boldsymbol{C}_i^T\right]\right\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

We write $\mathbb{E}\left[\sum_{i=1}^N \boldsymbol{B}_i\boldsymbol{B}_i^T\right]$ as

$$\begin{aligned}\mathbb{E}\left[\sum_{i=1}^N \boldsymbol{B}_i\boldsymbol{B}_i^T\right] &= \mathbb{E}\left[\sum_{i=1}^N \frac{1}{m\sqrt{m}}\boldsymbol{Q}_{-i}^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\bar{\boldsymbol{Q}}_m^T\boldsymbol{M}^T\boldsymbol{Q}_{-i}\right] \\ &= \frac{1}{\sqrt{m}}\frac{N}{m}\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{M}\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\bar{\boldsymbol{Q}}_m^T\boldsymbol{M}^T\boldsymbol{Q}_-\right],\end{aligned}$$

With a similar proof than for Lemma A.4.1, we can show there exists a real $K_{\bar{\boldsymbol{Q}}} > 0$ such that, for all m, we have

$$\|\bar{\boldsymbol{Q}}_m\| \leq K_{\bar{\boldsymbol{Q}}}.$$

Let $\hat{\boldsymbol{A}}_m = \hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T$ be the empirical transition model matrix defined in equation 6.14. Under Assumption 2, $\hat{\boldsymbol{A}}_m$ is invertible. From Lemma A.4.3, we have

$$\begin{aligned}\left\|\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\right\| &= \left\|\bar{\boldsymbol{Q}}_m(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\hat{\boldsymbol{A}}_m^{-1}\right\| \\ &= \left\|\frac{m}{N}(1+\delta)\left[\boldsymbol{I}_n - \lambda\bar{\boldsymbol{Q}}_m\right](\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\hat{\boldsymbol{A}}_m^{-1}\right\| \\ &\leq 2\frac{m}{N}\frac{1+\delta}{\xi_{\min}}(1 + K_{\bar{\boldsymbol{Q}}}).\end{aligned}$$

From above and from Lemma A.2.9 that upper bounds $\|\boldsymbol{M}\bar{\boldsymbol{Q}}_m\|$, we conclude that

$$\left\|\mathbb{E}\left[\sum_{i=1}^N \boldsymbol{B}_i\boldsymbol{B}_i^T\right]\right\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

□

Lemma A.2.9. *Under Assumptions 1 and 2, let $\bar{\boldsymbol{Q}}_m \in \mathbb{R}^{n\times n}$ be the deterministic resolvent defined in equation 7.7, and let $\boldsymbol{M}$ be either any matrix with a bounded operator norm or $\boldsymbol{M} = \hat{\boldsymbol{U}}_n^T\boldsymbol{\Psi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n$. Then there exists a real $K > 0$ such that, for all m, we have*

$$\|\boldsymbol{M}\bar{\boldsymbol{Q}}_m\| \leq K.$$

Proof. With a similar proof than for Lemma A.4.1, we can show there exists a real $K_{\bar{\boldsymbol{Q}}} > 0$ such

that, for all m, we have

$$\|\bar{\boldsymbol{Q}}_m\| \leq K_{\bar{\boldsymbol{Q}}}.$$

In the case where $\boldsymbol{M}$ is a matrix with a bounded operator norm, i.e., $\|\boldsymbol{M}\| \leq K_{\boldsymbol{M}}$ we have

$$\|\boldsymbol{M}\bar{\boldsymbol{Q}}_m\| \leq K_{\boldsymbol{M}} K_{\bar{\boldsymbol{Q}}}.$$

Otherwise, when $\boldsymbol{M} = \hat{\boldsymbol{U}}_n^T \boldsymbol{\Psi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n$, we consider $\hat{\boldsymbol{A}}_m = \hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T$ the empirical transition model matrix defined in equation 6.14. Under Assumption 2, $\hat{\boldsymbol{A}}_m$ is invertible. From Lemma A.4.3, we have

$$\begin{aligned}
\|\boldsymbol{M}\bar{\boldsymbol{Q}}_m\| &= \left\| \frac{N}{m} \frac{1}{1+\delta} \hat{\boldsymbol{U}}_n^T \boldsymbol{\Phi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \bar{\boldsymbol{Q}}_m \right\| \\
&= \left\| \frac{N}{m} \frac{1}{1+\delta} \hat{\boldsymbol{U}}_n^T \hat{\boldsymbol{A}}_m^{-1} \hat{\boldsymbol{U}}_n (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\Phi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n \bar{\boldsymbol{Q}}_m \right\| \\
&= \left\| \hat{\boldsymbol{U}}_n^T \hat{\boldsymbol{A}}_m^{-1} \hat{\boldsymbol{U}}_n \left[\boldsymbol{I}_n - \lambda \bar{\boldsymbol{Q}}_m \right] \right\| \\
&\leq \frac{1}{\xi_{\min}} (1 + \lambda K_{\bar{\boldsymbol{Q}}}).
\end{aligned}$$

□

A.3 Proof of Theorem 7.4.2

To simplify the notations, we denote the matrix $\boldsymbol{Q}_m$ as the resolvent $\boldsymbol{Q}_m(\lambda)$ (defined in equation 6.13), and we set $p = |\mathcal{S}|$. We define the matrices $\boldsymbol{\Psi}_{\hat{\mathcal{S}}} \in \mathbb{R}^{m \times m}$ and $\boldsymbol{\Psi}_{\mathcal{S}} \in \mathbb{R}^{p \times p}$ as

$$\boldsymbol{\Psi}_{\hat{\mathcal{S}}} = \frac{N}{m} \frac{1}{1+\delta} \boldsymbol{\Phi}_{\hat{\mathcal{S}}} \quad \text{and} \quad \boldsymbol{\Psi}_{\mathcal{S}} = \frac{N}{m} \frac{1}{1+\delta} \boldsymbol{\Phi}_{\mathcal{S}}.$$

We also add the notation $\boldsymbol{A} = \boldsymbol{B} + \mathcal{O}_{\|\cdot\|}\left(\frac{1}{\sqrt{m}}\right)$ which means that $\|\boldsymbol{A} - \boldsymbol{B}\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$.

Under Assumptions 1, 2 and 3, this section is dedicated to finding an asymptotic deterministic version of the true $\mathrm{MSBE}(\hat{\boldsymbol{\theta}}_n^\lambda)$ defined in equation 3.2 with a similar approach than the one used in Appendix A.2 for $\widehat{\mathrm{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda)$.

Theorem A.3.1 (Asymptotic MSBE)**.** *Under Assumptions 1, 2, and 3, the determinsitic asymptotic MSBE is*

$$\overline{\mathrm{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda) = \left\| \bar{\boldsymbol{r}} + \gamma \tfrac{1}{\sqrt{n}} \boldsymbol{P}^\pi \boldsymbol{\Psi}_{\mathcal{S}} \boldsymbol{U}_n \bar{\boldsymbol{Q}}_m(\lambda) \boldsymbol{r} - \tfrac{1}{\sqrt{n}} \boldsymbol{\Psi}_{\mathcal{S}} \boldsymbol{U}_n \bar{\boldsymbol{Q}}_m(\lambda) \boldsymbol{r} \right\|_{\boldsymbol{D}_{\boldsymbol{\mu}^\pi}}^2 + \Delta,$$

with second-order correction factor

$$\Delta = \tfrac{1}{n} \frac{\frac{1}{N} \mathrm{Tr}\left(\boldsymbol{\Lambda}_{\boldsymbol{P}} \left[\boldsymbol{\Theta}_{\mathcal{S}} \boldsymbol{\Psi}_2 \boldsymbol{\Theta}_{\mathcal{S}}^T - 2 \boldsymbol{\Theta}_{\mathcal{S}} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\Psi}_{\mathcal{S}} + \boldsymbol{\Psi}_{\mathcal{S}} \right] \right)}{1 - \frac{1}{N} \mathrm{Tr}\left(\boldsymbol{\Psi}_2 \bar{\boldsymbol{Q}}_m(\lambda)^T \boldsymbol{\Psi}_1 \bar{\boldsymbol{Q}}_m(\lambda) \right)} \|\bar{\boldsymbol{Q}}_m(\lambda) \boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2,$$

where

$$\begin{aligned}
\boldsymbol{\Lambda}_{\boldsymbol{P}} &= [\boldsymbol{I}_{|\mathcal{S}|} - \gamma \boldsymbol{P}^\pi]^T \boldsymbol{D}_{\boldsymbol{\mu}^\pi} [\boldsymbol{I}_{|\mathcal{S}|} - \gamma \boldsymbol{P}^\pi], \\
\boldsymbol{\Theta}_{\mathcal{S}} &= \boldsymbol{\Psi}_{\mathcal{S}} \boldsymbol{U}_n \bar{\boldsymbol{Q}}_m(\lambda).
\end{aligned}$$

As $N, m, d \to \infty$ with asymptotic constant ratio N/m, $\mathrm{MSBE}(\hat{\boldsymbol{\theta}}_n^\lambda) - \overline{\mathrm{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda) \xrightarrow{a.s} 0$.

Proof. We decompose $\mathrm{MSBE}(\hat{\boldsymbol{\theta}}_n^\lambda)$ as

$$\begin{aligned}
\mathrm{MSBE}(\hat{\boldsymbol{\theta}}_n^\lambda) = \|\bar{\boldsymbol{r}} + \gamma \boldsymbol{P}^\pi \boldsymbol{\Sigma}_{\mathcal{S}}^T \hat{\boldsymbol{\theta}}_n^\lambda - \boldsymbol{\Sigma}_{\mathcal{S}}^T \hat{\boldsymbol{\theta}}_n^\lambda\|_{\boldsymbol{D}_{\boldsymbol{\mu}^\pi}}^2 &= \|\bar{\boldsymbol{r}} + \left[\gamma \boldsymbol{P}^\pi - \boldsymbol{I}_p\right] \boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\theta}_n^\lambda\|_{\boldsymbol{D}_{\boldsymbol{\mu}^\pi}}^2 && \text{(A.34)} \\
&= \left\| \bar{\boldsymbol{r}} - \frac{1}{m\sqrt{n}} \left[\boldsymbol{I}_p - \gamma \boldsymbol{P}^\pi\right] \boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\Sigma}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_m \boldsymbol{r} \right\|_{\boldsymbol{D}_{\boldsymbol{\mu}^\pi}}^2 && \text{(A.35)} \\
&= \|\bar{\boldsymbol{r}}\|_{\boldsymbol{D}_{\boldsymbol{\mu}^\pi}}^2 && \text{(A.36)} \\
&\quad \underbrace{- \frac{2}{m\sqrt{n}} \bar{\boldsymbol{r}}^T \boldsymbol{D}_{\boldsymbol{\mu}^\pi} \left[\boldsymbol{I}_p - \gamma \boldsymbol{P}^\pi\right] \boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\Sigma}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_m \boldsymbol{r}}_{=Z_2} && \text{(A.37)} \\
&\quad \underbrace{+ \left\| \frac{1}{m\sqrt{n}} \left[\boldsymbol{I}_p - \gamma \boldsymbol{P}^\pi\right] \boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\Sigma}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_m \boldsymbol{r} \right\|_{\boldsymbol{D}_{\boldsymbol{\mu}^\pi}}^2}_{=Z_3} . && \text{(A.38)}
\end{aligned}$$

We want to find an asymptotic equivalent for both Z_2 and Z_3. From Lemma A.3.2, we have

$$\mathbb{E}\left[Z_2\right] = \frac{2}{\sqrt{n}} \bar{\boldsymbol{r}}^T \boldsymbol{D}_{\boldsymbol{\mu}^\pi} \left[\boldsymbol{I}_p - \gamma \boldsymbol{P}^\pi\right] \boldsymbol{\Psi}_{\mathcal{S}} \boldsymbol{U}_n \bar{\boldsymbol{Q}}_m \boldsymbol{r} + \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

For Z_3, we have

$$Z_3 = \left\| \frac{1}{m\sqrt{n}} \left[\boldsymbol{I}_p - \gamma \boldsymbol{P}^\pi\right] \boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\Sigma}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_m \boldsymbol{r} \right\|_{\boldsymbol{D}_{\boldsymbol{\mu}^\pi}}^2 = \frac{1}{nm^2} \boldsymbol{r}^T \boldsymbol{Q}_m^T \boldsymbol{U}_n^T \boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\Sigma}_{\mathcal{S}} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\Sigma}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_m \boldsymbol{r}.$$

From Lemma A.3.3, we have

$$\begin{aligned}
\mathbb{E}\left[Z_3\right] = &\frac{1}{n} \boldsymbol{r}^T \bar{\boldsymbol{Q}}_m^T \boldsymbol{U}_n^T \boldsymbol{\Psi}_{\mathcal{S}} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Psi}_{\mathcal{S}} \boldsymbol{U}_n \bar{\boldsymbol{Q}}_m \boldsymbol{r} \\
&+ \frac{1}{n} \frac{\frac{1}{N} \operatorname{Tr}\left(\boldsymbol{\Lambda}_{\boldsymbol{P}} \left[\boldsymbol{\Theta}_{\mathcal{S}} \boldsymbol{\Psi}_2 \boldsymbol{\Theta}_{\mathcal{S}}^T - 2 \boldsymbol{\Theta}_{\mathcal{S}} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\Psi}_{\mathcal{S}} + \boldsymbol{\Psi}_{\mathcal{S}}\right]\right)}{1 - \frac{1}{N} \operatorname{Tr}\left(\boldsymbol{\Psi}_2 \bar{\boldsymbol{Q}}_m^T \boldsymbol{\Psi}_1 \bar{\boldsymbol{Q}}_m\right)} \|\bar{\boldsymbol{Q}}_m \boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2 \\
&+ \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).
\end{aligned}$$

With a similar proof than for Lemma 7.3.1 we can deduce that

$$\mathrm{MSBE}(\hat{\boldsymbol{\theta}}_n^\lambda) - \overline{\mathrm{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda) \xrightarrow{a.s} 0,$$

as $m \to \infty$. □

Lemma A.3.2. *Under Assumptions 1, 2 and 3, let $Z_2 \in \mathbb{R}$ defined in equation A.37 as*

$$Z_2 = \frac{1}{m\sqrt{n}} \mathbb{E}\left[\bar{\boldsymbol{r}}^T \boldsymbol{D}_{\boldsymbol{\mu}^\pi} \left[\boldsymbol{I}_p - \gamma \boldsymbol{P}^\pi\right] \boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\Sigma}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_m \boldsymbol{r}\right].$$

Then

$$\left| Z_2 - \frac{1}{\sqrt{n}} \bar{\boldsymbol{r}}^T \boldsymbol{D}_{\boldsymbol{\mu}^\pi} \left[\boldsymbol{I}_p - \gamma \boldsymbol{P}^\pi\right] \boldsymbol{\Psi}_{\mathcal{S}} \boldsymbol{U}_n \bar{\boldsymbol{Q}}_m \boldsymbol{r} \right| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right),$$

for $\bar{\boldsymbol{Q}}_m$ the deterministic resolvent defined in equation 7.7, and $\boldsymbol{\Psi}_{\mathcal{S}} \in \mathbb{R}^{p \times p}$ defined in equation 7.15.

Proof. As in equation A.6, we decompose the matrix $\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}$ as

$$\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}} = \sum_{i=1}^{N}\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T,$$

where $\boldsymbol{\sigma}_i = \sigma(\boldsymbol{S}^T\boldsymbol{w}_i) \in \mathbb{R}^m$ for which $\boldsymbol{w}_i \in \mathbb{R}^d$ denotes the i-th row of $\boldsymbol{W}$ defined in equation 6.5. Let $\boldsymbol{Q}_{-i} \in \mathbb{R}^{n\times n}$ be the following resolvent

$$\boldsymbol{Q}_{-i} = \left[\frac{1}{m}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{U}_n - \frac{1}{m}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{U}_n + \lambda\boldsymbol{I}_n\right]^{-1},$$

independent of $\boldsymbol{\sigma}_i$ and thus $\boldsymbol{w}_i$. From the Sherman identity (Lemma A.8.3), we have

$$\begin{aligned}
Z_2 &= \frac{1}{m\sqrt{n}}\mathbb{E}\left[\bar{\boldsymbol{r}}^T\boldsymbol{D}_{\boldsymbol{\mu}^\pi}\left[\boldsymbol{I}_p - \gamma\boldsymbol{P}^\pi\right]\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_m\boldsymbol{r}\right] \\
&= \frac{1}{m\sqrt{n}}\mathbb{E}\left[\sum_{i=1}^{N}\bar{\boldsymbol{r}}^T\boldsymbol{D}_{\boldsymbol{\mu}^\pi}\left[\boldsymbol{I}_p - \gamma\boldsymbol{P}^\pi\right]\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_m\boldsymbol{r}\right] \\
&= \frac{1}{m\sqrt{n}}\mathbb{E}\left[\sum_{i=1}^{N}\frac{\bar{\boldsymbol{r}}^T\boldsymbol{D}_{\boldsymbol{\mu}^\pi}\left[\boldsymbol{I}_p - \gamma\boldsymbol{P}^\pi\right]\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-i}\boldsymbol{r}}{1 + \frac{1}{m}\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-i}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_i}\right].
\end{aligned}$$

Let $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ be a diagonal matrix for which, for all $i \in [N]$, we have

$$\boldsymbol{D}_i = \delta - \frac{1}{m}\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-i}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_i.$$

We replace $1 + \frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j$ by $1 + \delta$ as following

$$\begin{aligned}
Z_2 = &\underbrace{\frac{1}{m\sqrt{n}}\frac{1}{1+\delta}\mathbb{E}\left[\sum_{i=1}^{N}\bar{\boldsymbol{r}}^T\boldsymbol{D}_{\boldsymbol{\mu}^\pi}\left[\boldsymbol{I}_p - \gamma\boldsymbol{P}^\pi\right]\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-i}\boldsymbol{r}\right]}_{Z_{21}} \\
&+ \underbrace{\frac{1}{m\sqrt{n}}\frac{1}{1+\delta}\mathbb{E}\left[\sum_{i=1}^{N}\frac{\bar{\boldsymbol{r}}^T\boldsymbol{D}_{\boldsymbol{\mu}^\pi}\left[\boldsymbol{I}_p - \gamma\boldsymbol{P}^\pi\right]\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-i}\boldsymbol{D}_i\boldsymbol{r}}{1 + \frac{1}{m}\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-i}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_i}\right]}_{Z_{22}}.
\end{aligned}$$

We have Z_{22} vanishing since $\mathbb{E}\big[\|\boldsymbol{D}\|\big] = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$ and from Lemma 7.4.1. From Theorem 7.2.3, we have thus

$$\begin{aligned}
Z_2 &= \frac{1}{m\sqrt{n}}\frac{1}{1+\delta}\mathbb{E}\left[\sum_{i=1}^{N}\bar{\boldsymbol{r}}^T\boldsymbol{D}_{\boldsymbol{\mu}^\pi}\left[\boldsymbol{I}_p - \gamma\boldsymbol{P}^\pi\right]\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-i}\boldsymbol{r}\right] + \mathcal{O}\left(\frac{1}{\sqrt{m}}\right) \\
&= \frac{1}{\sqrt{n}}\frac{N}{m}\frac{1}{1+\delta}\bar{\boldsymbol{r}}^T\boldsymbol{D}_{\boldsymbol{\mu}^\pi}\left[\boldsymbol{I}_p - \gamma\boldsymbol{P}^\pi\right]\boldsymbol{\Phi}_{\mathcal{S}}\boldsymbol{U}_n\mathbb{E}\big[\boldsymbol{Q}_-\big]\boldsymbol{r} + \mathcal{O}\left(\frac{1}{\sqrt{m}}\right) \\
&= \frac{1}{\sqrt{n}}\bar{\boldsymbol{r}}^T\boldsymbol{D}_{\boldsymbol{\mu}^\pi}\left[\boldsymbol{I}_p - \gamma\boldsymbol{P}^\pi\right]\boldsymbol{\Psi}_{\mathcal{S}}\boldsymbol{U}_n\bar{\boldsymbol{Q}}_m\boldsymbol{r} + \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).
\end{aligned}$$

□

Lemma A.3.3. *Under Assumptions 1, 2 and 3, let* $\boldsymbol{\Lambda}_{\boldsymbol{P}} \in \mathbb{R}^{p\times p}$ *be the matrix defined in equa-*

tion 7.15, and let $Z_3 \in \mathbb{R}$ be defined in equation A.38 as

$$Z_3 = \mathbb{E}\left[\frac{1}{nm^2}\boldsymbol{r}^T\boldsymbol{Q}_m^T\boldsymbol{U}_n^T\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_m\boldsymbol{r}\right].$$

Then

$$\left|Z_3 - \frac{1}{n}\boldsymbol{r}^T\bar{\boldsymbol{Q}}_m^T\boldsymbol{U}_n^T\boldsymbol{\Psi}_{\mathcal{S}}\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Psi}_{\mathcal{S}}\boldsymbol{U}_n\bar{\boldsymbol{Q}}_m\boldsymbol{r} - \frac{1}{n}\frac{\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Lambda}_{\boldsymbol{P}}\left[\boldsymbol{\Theta}_{\mathcal{S}}\boldsymbol{\Psi}_2\boldsymbol{\Theta}_{\mathcal{S}}^T - 2\boldsymbol{\Theta}_{\mathcal{S}}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\Psi}_{\mathcal{S}} + \boldsymbol{\Psi}_{\mathcal{S}}\right]\right)}{1 - \frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m\right)}\|\bar{\boldsymbol{Q}}_m\boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2\right| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right),$$

where $\bar{\boldsymbol{Q}}_m$ is the deterministic resolvent defined in equation 7.7, $\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2 \in \mathbb{R}^{n\times n}$ are defined in equation 7.12, $\boldsymbol{\Psi}_{\mathcal{S}} \in \mathbb{R}^{p\times p}$ and $\boldsymbol{\Theta}_{\mathcal{S}} \in \mathbb{R}^{p\times n}$ are defined in equation 7.15.

Proof. As in equation A.6, we decompose the matrix $\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}$ as

$$\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}} = \sum_{i=1}^N \boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T,$$

where $\boldsymbol{\sigma}_i = \sigma(\boldsymbol{S}^T\boldsymbol{w}_i) \in \mathbb{R}^m$ for which $\boldsymbol{w}_i \in \mathbb{R}^d$ denotes the i-th row of $\boldsymbol{W}$ defined in equation 6.5. Let $\boldsymbol{Q}_{-i} \in \mathbb{R}^{n\times n}$ be the following resolvent

$$\boldsymbol{Q}_{-i} = \left[\frac{1}{m}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{U}_n - \frac{1}{m}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{U}_n + \lambda\boldsymbol{I}_n\right]^{-1}$$

independent of $\boldsymbol{\sigma}_i$ and thus $\boldsymbol{w}_i$. From the Sherman identity (Lemma A.8.3), we decompose Z_3 as

$$Z_3 = \mathbb{E}\left[\frac{1}{nm^2}\boldsymbol{r}^T\boldsymbol{Q}_m^T\boldsymbol{U}_n^T\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_m\boldsymbol{r}\right] \tag{A.39}$$

$$= \sum_{i,j=1}^N \mathbb{E}\left[\frac{1}{nm^2}\boldsymbol{r}^T\boldsymbol{Q}_m^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_m\boldsymbol{r}\right] \tag{A.40}$$

$$= \sum_{i,j=1}^N \mathbb{E}\left[\frac{1}{nm^2}\boldsymbol{r}^T\frac{\boldsymbol{Q}_{-i}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T}{1 + \frac{1}{m}\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-i}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_i}\boldsymbol{\Lambda}_{\boldsymbol{P}}\frac{\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}}{1 + \frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j}\boldsymbol{r}\right] \tag{A.41}$$

$$= \underbrace{\sum_{i=1}^N\sum_{j\neq i}\mathbb{E}\left[\frac{1}{nm^2}\boldsymbol{r}^T\frac{\boldsymbol{Q}_{-i}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T}{1 + \frac{1}{m}\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-i}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_i}\boldsymbol{\Lambda}_{\boldsymbol{P}}\frac{\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}}{1 + \frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j}\boldsymbol{r}\right]}_{=Z_{31}}$$

$$+ \underbrace{\sum_{i=1}^N\mathbb{E}\left[\frac{1}{nm^2}\boldsymbol{r}^T\frac{\boldsymbol{Q}_{-i}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-i}}{\left(1 + \frac{1}{m}\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-i}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_i\right)^2}\boldsymbol{r}\right]}_{=Z_{32}}. \tag{A.42}$$

From Lemma A.3.4, we have

$$Z_{31} = \frac{1}{n}\boldsymbol{r}^T\bar{\boldsymbol{Q}}_m^T\boldsymbol{U}_n^T\boldsymbol{\Psi}_{\mathcal{S}}\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Psi}_{\mathcal{S}}\boldsymbol{U}_n\bar{\boldsymbol{Q}}_m\boldsymbol{r}$$

$$+\frac{1}{n}\frac{\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Lambda}_{\boldsymbol{P}}\left[\boldsymbol{\Theta}_{\mathcal{S}}\boldsymbol{\Psi}_2\boldsymbol{\Theta}_{\mathcal{S}}^T-2\boldsymbol{\Theta}_{\mathcal{S}}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\Psi}_{\mathcal{S}}\right]\right)}{1-\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m\right)}\|\bar{\boldsymbol{Q}}_m\boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2$$
$$+\mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

For the second term Z_{32}, we have from Lemma A.3.5,

$$Z_{32}=\frac{1}{n}\frac{\frac{1}{N}\operatorname{Tr}(\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Psi}_{\mathcal{S}})}{1-\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m\right)}\|\bar{\boldsymbol{Q}}_m\boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2+\mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

We conclude that

$$\begin{aligned}Z_3&=Z_{32}+Z_{32}\\&=\frac{1}{n}\boldsymbol{r}^T\bar{\boldsymbol{Q}}_m^T\boldsymbol{U}_n^T\boldsymbol{\Psi}_{\mathcal{S}}\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Psi}_{\mathcal{S}}\boldsymbol{U}_n\bar{\boldsymbol{Q}}_m\boldsymbol{r}\\&+\frac{1}{n}\frac{\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Lambda}_{\boldsymbol{P}}\left[\boldsymbol{\Theta}_{\mathcal{S}}\boldsymbol{\Psi}_2\boldsymbol{\Theta}_{\mathcal{S}}^T-2\boldsymbol{\Theta}_{\mathcal{S}}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\Psi}_{\mathcal{S}}+\boldsymbol{\Psi}_{\mathcal{S}}\right]\right)}{1-\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m\right)}\|\bar{\boldsymbol{Q}}_m\boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2\\&+\mathcal{O}\left(\frac{1}{\sqrt{m}}\right).\end{aligned}$$

□

Lemma A.3.4. *Under Assumptions 1, 2 and 3, let $Z_{31}\in\mathbb{R}$ defined in equation A.42 as*

$$Z_{31}=\frac{1}{nm^2}\sum_{i=1}^N\sum_{j\neq i}\mathbb{E}\left[\frac{\boldsymbol{r}^T\boldsymbol{Q}_{-i}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}\boldsymbol{r}}{\left(1+\frac{1}{m}\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-i}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_i\right)\left(1+\frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j\right)}\right],$$

Then

$$\begin{aligned}&\Bigg|Z_{31}-\frac{1}{n}\boldsymbol{r}^T\bar{\boldsymbol{Q}}_m^T\boldsymbol{U}_n^T\boldsymbol{\Psi}_{\mathcal{S}}\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Psi}_{\mathcal{S}}\boldsymbol{U}_n\bar{\boldsymbol{Q}}_m\boldsymbol{r}\\&\quad-\frac{1}{n}\frac{\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Lambda}_{\boldsymbol{P}}\left[\boldsymbol{\Theta}_{\mathcal{S}}\boldsymbol{\Psi}_2\boldsymbol{\Theta}_{\mathcal{S}}^T-2\boldsymbol{\Theta}_{\mathcal{S}}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\Psi}_{\mathcal{S}}\right]\right)}{1-\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m\right)}\|\bar{\boldsymbol{Q}}_m\boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2\Bigg|=\mathcal{O}\left(\frac{1}{\sqrt{m}}\right),\end{aligned}$$

where $\bar{\boldsymbol{Q}}_m$ is the deterministic resolvent defined in equation 7.7, $\boldsymbol{\Psi}_1,\boldsymbol{\Psi}_2\in\mathbb{R}^{n\times n}$ are defined in equation 7.12, $\boldsymbol{\Psi}_{\mathcal{S}}\in\mathbb{R}^{p\times p}$ and $\boldsymbol{\Theta}_{\mathcal{S}}\in\mathbb{R}^{p\times n}$ are defined in equation 7.15.

Proof. Using the Sherman identity (Lemma A.8.3), we decompose Z_{31} as

$$\begin{aligned}Z_{31}&=\frac{1}{nm^2}\sum_{i=1}^N\sum_{j\neq i}\mathbb{E}\left[\frac{\boldsymbol{r}^T\boldsymbol{Q}_{-i}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}\boldsymbol{r}}{\left(1+\frac{1}{m}\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-i}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_i\right)\left(1+\frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j\right)}\right]\\&=\frac{1}{nm^2}\sum_{i=1}^N\sum_{j\neq i}\mathbb{E}\left[\boldsymbol{r}^T\frac{\boldsymbol{Q}_m^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}}{1+\frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j}\boldsymbol{r}\right]\\&=\underbrace{\frac{1}{nm^2}\sum_{i=1}^N\sum_{j\neq i}\mathbb{E}\left[\boldsymbol{r}^T\frac{\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}}{1+\frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j}\boldsymbol{r}\right]}_{=Z_{311}}\end{aligned}$$

$$\underbrace{-\frac{1}{nm^3}\sum_{i=1}^{N}\sum_{j\neq i}\mathbb{E}\left[\boldsymbol{r}^T\frac{\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}}{\left(1+\frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j\right)^2}\boldsymbol{r}\right]}_{=Z_{312}}.$$

We want to find an asymptotic equivalent for both Z_{311} and Z_{312}. For Z_{312}, we have

$$\begin{aligned}Z_{312}&=\frac{1}{nm^3}\sum_{i=1}^{N}\sum_{j\neq i}\mathbb{E}\left[\boldsymbol{r}^T\frac{\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}}{\left(1+\frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j\right)^2}\boldsymbol{r}\right]\\&=\frac{1}{nm}\sum_{j=1}^{N}\mathbb{E}\left[\boldsymbol{r}^T\frac{\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}\left(\frac{1}{m^2}\boldsymbol{\sigma}_j^T(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\Sigma}_{\mathcal{S}}^{-jT}\boldsymbol{\Sigma}_{\mathcal{S}}^{-j}\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\sigma}_j\right)}{\left(1+\frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j\right)^2}\boldsymbol{r}\right],\end{aligned}$$

where $\boldsymbol{\Sigma}_{\mathcal{S}}^{-j}=\sigma(\boldsymbol{W}^{-j}\boldsymbol{S})\in\mathbb{R}^{(N-1)\times n}$ for which $\boldsymbol{W}^{-j}\in\mathbb{R}^{(N-1)\times d}$ depicts the same matrix than the weight matrix $\boldsymbol{W}$ defined in equation 6.5 without the j^{th} row. Let $\boldsymbol{D}_{312}\in\mathbb{R}^{N\times N}$ be a diagonal matrix for which, for all $j\in[N]$, we have

$$[\boldsymbol{D}_{312}]_j=\frac{1}{m^2}\boldsymbol{\sigma}_j^T(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\Sigma}_{\mathcal{S}}^{-jT}\boldsymbol{\Sigma}_{\mathcal{S}}^{-j}\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\sigma}_j-\frac{1}{m^2}\operatorname{Tr}\left((\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\Sigma}_{\mathcal{S}}^{-jT}\boldsymbol{\Sigma}_{\mathcal{S}}^{-j}\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Phi}_{\mathcal{S}}\right).$$

From Lemma 7.4.1, we know there exists a real $K_1>0$ such that, for all m, we have $\|\boldsymbol{D}_{\boldsymbol{\mu}^\pi}[\boldsymbol{I}_p-\gamma\boldsymbol{P}^\pi]\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_m\|\leq K_1$. Therefore, we deduce that

$$\left\|\frac{1}{m}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\Sigma}_{\mathcal{S}}^{-jT}\boldsymbol{\Sigma}_{\mathcal{S}}^{-j}\boldsymbol{\Lambda}_{\boldsymbol{P}}\right\|=\mathcal{O}(1).$$

From Lemma A.7.2, we deduce that $\mathbb{E}\big[\|\boldsymbol{D}_{312}\|\big]=\mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$. Therefore, we get

$$\begin{aligned}Z_{312}&=\frac{1}{nm}\sum_{j=1}^{N}\mathbb{E}\left[\boldsymbol{r}^T\frac{\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}\frac{1}{m^2}\operatorname{Tr}\big((\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\Sigma}_{\mathcal{S}}^{-jT}\boldsymbol{\Sigma}_{\mathcal{S}}^{-j}\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Phi}_{\mathcal{S}}\big)}{\left(1+\frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j\right)^2}\boldsymbol{r}\right]\\&+\frac{1}{nm}\sum_{j=1}^{N}\mathbb{E}\left[\boldsymbol{r}^T\frac{\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}[\boldsymbol{D}_{312}]_j}{\left(1+\frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j\right)^2}\boldsymbol{r}\right]\\&=\frac{1}{nm}\sum_{j=1}^{N}\mathbb{E}\left[\boldsymbol{r}^T\frac{\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}\frac{1}{m^2}\operatorname{Tr}\big((\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\Sigma}_{\mathcal{S}}^{-jT}\boldsymbol{\Sigma}_{\mathcal{S}}^{-j}\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Phi}_{\mathcal{S}}\big)}{\left(1+\frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j\right)^2}\boldsymbol{r}\right]\\&+\frac{1}{nm}\mathbb{E}\left[\boldsymbol{r}^T\boldsymbol{Q}_m^T\boldsymbol{U}_n^T\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{D}_{312}\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_m\boldsymbol{r}\right]\\&=\frac{1}{nm}\sum_{j=1}^{N}\mathbb{E}\left[\boldsymbol{r}^T\frac{\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}\frac{1}{m^2}\operatorname{Tr}\big((\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\Sigma}_{\mathcal{S}}^{-jT}\boldsymbol{\Sigma}_{\mathcal{S}}^{-j}\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Phi}_{\mathcal{S}}\big)}{\left(1+\frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j\right)^2}\boldsymbol{r}\right]\\&+\mathcal{O}\left(\frac{1}{\sqrt{m}}\right),\end{aligned}\tag{A.43}$$

where the last equality is obtained since $\mathbb{E}\big[\|\boldsymbol{D}_{312}\|\big]=\mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$, and since we know there exists a real $K'_{\boldsymbol{Q}}>0$ such that, for all m, we have

$$\left\|\frac{1}{\sqrt{m}}\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_m\right\|\leq K'_{\boldsymbol{Q}}$$

and

$$\left\| \frac{1}{\sqrt{m}} \boldsymbol{Q}_m (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\Sigma}_{\mathcal{S}}^T \right\| \leq 2K'_{\boldsymbol{Q}}.$$

from Lemma A.4.4. We replace $1 + \frac{1}{m} \boldsymbol{\sigma}_j^T \boldsymbol{U}_n \boldsymbol{Q}_{-j} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_j$ by $1 + \delta$ in Z_{312} as following

$$\begin{aligned}
Z_{312} &= \frac{1}{nm^3} \frac{1}{(1+\delta)^2} \sum_{j=1}^{N} \mathbb{E}\left[\boldsymbol{r}^T \boldsymbol{Q}_{-j}^T \boldsymbol{U}_n^T \boldsymbol{\sigma}_j \boldsymbol{\sigma}_j^T \boldsymbol{U}_n \boldsymbol{Q}_{-j} \operatorname{Tr}\big((\boldsymbol{U}_n - \gamma \boldsymbol{V}_n) \boldsymbol{Q}_{-j}^T \boldsymbol{U}_n^T \boldsymbol{\Sigma}_{\mathcal{S}}^{-jT} \boldsymbol{\Sigma}_{\mathcal{S}}^{-j} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}}\big) \boldsymbol{r} \right] \\
&\quad + \frac{1}{nm^3} \frac{1}{(1+\delta)^2} \sum_{j=1}^{N} \mathbb{E}\left[\boldsymbol{r}^T \frac{\boldsymbol{Q}_{-j}^T \boldsymbol{U}_n^T \boldsymbol{\sigma}_j \boldsymbol{D}'_j \boldsymbol{\sigma}_j^T \boldsymbol{U}_n \boldsymbol{Q}_{-j} \operatorname{Tr}\big((\boldsymbol{U}_n - \gamma \boldsymbol{V}_n) \boldsymbol{Q}_{-j}^T \boldsymbol{U}_n^T \boldsymbol{\Sigma}_{\mathcal{S}}^{-jT} \boldsymbol{\Sigma}_{\mathcal{S}}^{-j} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}}\big)}{\big(1 + \frac{1}{m} \boldsymbol{\sigma}_j^T \boldsymbol{U}_n \boldsymbol{Q}_{-j} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_j\big)^2} \boldsymbol{r} \right] \\
&\quad + \mathcal{O}\left(\frac{1}{\sqrt{m}}\right) \\
&= \underbrace{\frac{N}{nm^3} \frac{1}{(1+\delta)^2} \mathbb{E}\left[\boldsymbol{r}^T \boldsymbol{Q}_-^T \boldsymbol{U}_n^T \boldsymbol{\Phi}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_- \boldsymbol{r} \operatorname{Tr}\big((\boldsymbol{U}_n - \gamma \boldsymbol{V}_n) \boldsymbol{Q}_-^T \boldsymbol{U}_n^T \boldsymbol{\Sigma}_{\mathcal{S}}^{-T} \boldsymbol{\Sigma}_{\mathcal{S}}^{-} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}}\big) \right]}_{= Z_{3121}} \\
&\quad + \underbrace{\frac{1}{nm^3} \frac{1}{(1+\delta)^2} \mathbb{E}\left[\boldsymbol{r}^T \boldsymbol{Q}_m^T \boldsymbol{U}_n^T \boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{D}' \boldsymbol{\Sigma}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_m \boldsymbol{r} \operatorname{Tr}\big((\boldsymbol{U}_n - \gamma \boldsymbol{V}_n) \boldsymbol{Q}_-^T \boldsymbol{U}_n^T \boldsymbol{\Sigma}_{\mathcal{S}}^{-T} \boldsymbol{\Sigma}_{\mathcal{S}}^{-} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}}\big) \right]}_{= Z_{3122}} \\
&\quad + \mathcal{O}\left(\frac{1}{\sqrt{m}}\right),
\end{aligned}$$

where $\boldsymbol{D}' \in \mathbb{R}^{N \times N}$ is a diagonal matrix for which, for all $j \in [N]$, we have

$$\boldsymbol{D}'_j = (1+\delta)^2 - \left(1 + \frac{1}{m} \boldsymbol{\sigma}_j^T \boldsymbol{U}_n \boldsymbol{Q}_{-j} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_j\right)^2.$$

With a similar proof than for equation A.4, we can show $\frac{1}{m} \operatorname{Tr}\big(\boldsymbol{\Phi}_{\mathcal{S}}\big) = \frac{p}{m} \frac{1}{p} \operatorname{Tr}\big(\boldsymbol{\Phi}_{\mathcal{S}}\big)$ is uniformly bounded under Assumption 3. Combining $|\operatorname{Tr}(\boldsymbol{AB})| \leq \|\boldsymbol{A}\| \operatorname{Tr}(\boldsymbol{B})$ for non-negative definite matrix $\boldsymbol{B}$ and Lemma 7.4.1, we have $\frac{1}{m^2} \operatorname{Tr}\big((\boldsymbol{U}_n - \gamma \boldsymbol{V}_n) \boldsymbol{Q}_-^T \boldsymbol{U}_n^T \boldsymbol{\Sigma}_{\mathcal{S}}^{-T} \boldsymbol{\Sigma}_{\mathcal{S}}^{-} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}}\big) = \mathcal{O}(1)$. From all these upper bounds, and since it can be shown that $\mathbb{E}\big[\|\boldsymbol{D}'\|\big] = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$, we deduce the second term, Z_{3122}, vanishes and thus

$$\begin{aligned}
Z_{312} &= \frac{1}{nm^2} \frac{1}{1+\delta} \mathbb{E}\left[\boldsymbol{r}^T \boldsymbol{Q}_-^T \boldsymbol{\Psi}_1 \boldsymbol{Q}_- \boldsymbol{r} \operatorname{Tr}\big((\boldsymbol{U}_n - \gamma \boldsymbol{V}_n) \boldsymbol{Q}_-^T \boldsymbol{U}_n^T \boldsymbol{\Sigma}_{\mathcal{S}}^{-T} \boldsymbol{\Sigma}_{\mathcal{S}}^{-} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}}\big) \right] \\
&\quad + \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).
\end{aligned}$$

Let $\boldsymbol{Q}_{-ij} \in \mathbb{R}^{n \times n}$ be the resolvent defined as

$$\boldsymbol{Q}_{-ij} = \left[\frac{1}{m} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\Sigma}_{\mathcal{S}}^{-ijT} \boldsymbol{\Sigma}_{\mathcal{S}}^{-ij} \boldsymbol{U}_n + \lambda \boldsymbol{I}_n \right]^{-1}, \tag{A.44}$$

where $\boldsymbol{\Sigma}_{\mathcal{S}}^{-ij} = \sigma(\boldsymbol{W}^{-ij} \boldsymbol{S}) \in \mathbb{R}^{(N-2) \times n}$ for which $\boldsymbol{W}^{-ij} \in \mathbb{R}^{(N-2) \times d}$ depicts the same matrix than the weight matrix $\boldsymbol{W}$ defined in equation 6.5 without the i^{th} and j^{th} row. Using the Sherman identity (Lemma A.8.3), the term $\frac{1}{m^2} \operatorname{Tr}\big((\boldsymbol{U}_n - \gamma \boldsymbol{V}_n) \boldsymbol{Q}_-^T \boldsymbol{U}_n^T \boldsymbol{\Sigma}_{\mathcal{S}}^{-T} \boldsymbol{\Sigma}_{\mathcal{S}}^{-} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}}\big)$ in Z_{312} can be rewritten as

$$\frac{1}{m^2} \operatorname{Tr}\big((\boldsymbol{U}_n - \gamma \boldsymbol{V}_n) \boldsymbol{Q}_-^T \boldsymbol{U}_n^T \boldsymbol{\Sigma}_{\mathcal{S}}^{-T} \boldsymbol{\Sigma}_{\mathcal{S}}^{-} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}}\big)$$

$$
\begin{aligned}
&= \frac{1}{m^2} \operatorname{Tr}\left(\sum_{i\neq j} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n) \boldsymbol{Q}_{-j}^T \boldsymbol{U}_n^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}} \right) \\
&= \frac{1}{m^2} \operatorname{Tr}\left(\sum_{i\neq j} \frac{(\boldsymbol{U}_n - \gamma \boldsymbol{V}_n) \boldsymbol{Q}_{-ij}^T \boldsymbol{U}_n^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}}}{1 + \frac{1}{m} \boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-ij} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_i} \right) \\
&= \frac{1}{m^2} \frac{1}{1+\delta} \operatorname{Tr}\left(\sum_{i\neq j} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n) \boldsymbol{Q}_{-ij}^T \boldsymbol{U}_n^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}} \right) \\
&+ \frac{1}{m^2} \frac{1}{1+\delta} \operatorname{Tr}\left(\sum_{i\neq j} \frac{(\boldsymbol{U}_n - \gamma \boldsymbol{V}_n) \boldsymbol{Q}_{-ij}^T \boldsymbol{U}_n^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}} \left(\delta - \frac{1}{m} \boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-ij} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_i \right)}{1 + \frac{1}{m} \boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-ij} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_i} \right) \\
&= \frac{N}{m^2} \frac{1}{1+\delta} \operatorname{Tr}\left((\boldsymbol{U}_n - \gamma \boldsymbol{V}_n) \boldsymbol{Q}_{--}^T \boldsymbol{U}_n^T \boldsymbol{\Phi}_{\mathcal{S}} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}} \right) \\
&+ \frac{1}{m^2} \frac{1}{1+\delta} \operatorname{Tr}\left((\boldsymbol{U}_n - \gamma \boldsymbol{V}_n) \boldsymbol{Q}_{-j}^T \boldsymbol{U}_n^T \boldsymbol{\Sigma}_{\mathcal{S}}^{-jT} \boldsymbol{D} \boldsymbol{\Sigma}_{\mathcal{S}}^{-j} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}} \right),
\end{aligned}
$$

where $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ is a diagonal matrix for which, for all $i \in [N]$, we have

$$
\boldsymbol{D}_i = \delta - \frac{1}{m} \boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-ij} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_i.
$$

From the uniform boundness of $\frac{1}{m} \operatorname{Tr}\big(\boldsymbol{\Phi}_{\mathcal{S}}\big) = \frac{1}{K_r} \frac{1}{p} \operatorname{Tr}\big(\boldsymbol{\Phi}_{\mathcal{S}}\big)$, from Lemma 7.4.1, we have $\frac{1}{m^2} \operatorname{Tr}\big((\boldsymbol{U}_n - \gamma \boldsymbol{V}_n) \boldsymbol{Q}_{-}^T \boldsymbol{U}_n^T \boldsymbol{\Sigma}_{\mathcal{S}}^{-T} \boldsymbol{\Sigma}_{\mathcal{S}}^{-} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}}\big) = \mathcal{O}(1)$. Since the operator norm of $\mathbb{E}\big[\|\boldsymbol{D}\|\big]$ is of order $\mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$, we deduce the second term vanishes, and thus

$$
\begin{aligned}
&\frac{1}{m^2} \operatorname{Tr}\big((\boldsymbol{U}_n - \gamma \boldsymbol{V}_n) \boldsymbol{Q}_{-}^T \boldsymbol{U}_n^T \boldsymbol{\Sigma}_{\mathcal{S}}^{-T} \boldsymbol{\Sigma}_{\mathcal{S}}^{-} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}}\big) \\
&= \frac{N}{m^2} \frac{1}{1+\delta} \operatorname{Tr}\left((\boldsymbol{U}_n - \gamma \boldsymbol{V}_n) \boldsymbol{Q}_{--}^T \boldsymbol{U}_n^T \boldsymbol{\Phi}_{\mathcal{S}} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}} \right) + \mathcal{O}\left(\frac{1}{\sqrt{m}} \right).
\end{aligned}
$$

Applying Lemma A.2.4 and Lemma A.2.5, we deduce for Z_{312} that

$$
\begin{aligned}
Z_{312} &= \frac{1}{n} \frac{\frac{1}{N} \operatorname{Tr}\big((\boldsymbol{U}_n - \gamma \boldsymbol{V}_n) \bar{\boldsymbol{Q}}_m^T \boldsymbol{U}_n^T \boldsymbol{\Psi}_{\mathcal{S}} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Psi}_{\mathcal{S}}\big)}{1 - \frac{1}{N} \operatorname{Tr}\left(\boldsymbol{\Psi}_2 \bar{\boldsymbol{Q}}_m^T \boldsymbol{\Psi}_1 \bar{\boldsymbol{Q}}_m \right)} \|\bar{\boldsymbol{Q}}_m \boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2 + \mathcal{O}\left(\frac{1}{\sqrt{m}} \right) \\
&= \frac{1}{n} \frac{\frac{1}{N} \operatorname{Tr}\big(\boldsymbol{\Psi}_{\mathcal{S}} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n) \boldsymbol{\Theta}_{\mathcal{S}}^T \boldsymbol{\Lambda}_{\boldsymbol{P}}\big)}{1 - \frac{1}{N} \operatorname{Tr}\left(\boldsymbol{\Psi}_2 \bar{\boldsymbol{Q}}_m^T \boldsymbol{\Psi}_1 \bar{\boldsymbol{Q}}_m \right)} \|\bar{\boldsymbol{Q}}_m \boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2 + \mathcal{O}\left(\frac{1}{\sqrt{m}} \right).
\end{aligned}
$$

Now, we want to find an asymptotic equivalent for Z_{311}. We replace $1 + \frac{1}{m} \boldsymbol{\sigma}_j^T \boldsymbol{U}_n \boldsymbol{Q}_{-j} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_j$ by $1+\delta$ in Z_{311} as following

$$
\begin{aligned}
Z_{311} &= \frac{1}{nm^2} \sum_{i=1}^N \sum_{j\neq i} \mathbb{E}\left[\boldsymbol{r}^T \frac{\boldsymbol{Q}_{-j}^T \boldsymbol{U}_n^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\sigma}_j \boldsymbol{\sigma}_j^T \boldsymbol{U}_n \boldsymbol{Q}_{-j}}{1 + \frac{1}{m} \boldsymbol{\sigma}_j^T \boldsymbol{U}_n \boldsymbol{Q}_{-j} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_j} \boldsymbol{r} \right] \\
&= \frac{1}{nm^2} \sum_{j=1}^N \mathbb{E}\left[\boldsymbol{r}^T \frac{\boldsymbol{Q}_{-j}^T \boldsymbol{U}_n^T \boldsymbol{\Sigma}_{\mathcal{S}}^{-jT} \boldsymbol{\Sigma}_{\mathcal{S}}^{-j} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\sigma}_j \boldsymbol{\sigma}_j^T \boldsymbol{U}_n \boldsymbol{Q}_{-j}}{1 + \frac{1}{m} \boldsymbol{\sigma}_j^T \boldsymbol{U}_n \boldsymbol{Q}_{-j} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_j} \boldsymbol{r} \right] \\
&= \underbrace{\frac{1}{nm^2} \frac{1}{1+\delta} \sum_{j=1}^N \mathbb{E}\left[\boldsymbol{r}^T \boldsymbol{Q}_{-j}^T \boldsymbol{U}_n^T \boldsymbol{\Sigma}_{\mathcal{S}}^{-jT} \boldsymbol{\Sigma}_{\mathcal{S}}^{-j} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\sigma}_j \boldsymbol{\sigma}_j^T \boldsymbol{U}_n \boldsymbol{Q}_{-j} \boldsymbol{r} \right]}_{= Z_{3111}}
\end{aligned}
$$

$$+\underbrace{\frac{1}{nm^2}\frac{1}{1+\delta}\sum_{j=1}^{N}\mathbb{E}\left[\boldsymbol{r}^T\frac{\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\Sigma}_{\mathcal{S}}^{-jT}\boldsymbol{\Sigma}_{\mathcal{S}}^{-j}\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}\boldsymbol{D}_j}{1+\frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j}\boldsymbol{r}\right]}_{=Z_{3112}},$$

where $\boldsymbol{D}\in\mathbb{R}^{N\times N}$ is a diagonal matrix for which, for all $j\in[N]$, we have

$$\boldsymbol{D}_j=\delta-\frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j.$$

We observe that

$$\begin{aligned}
\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\frac{\boldsymbol{\Sigma}_{\mathcal{S}}^{-jT}\boldsymbol{\Sigma}_{\mathcal{S}}^{-j}}{m}&=\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\frac{\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}}{m}-\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\frac{\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T}{m}\\
&=\boldsymbol{Q}_m^T\boldsymbol{U}_n^T\frac{\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}}{m}+\frac{\frac{1}{m}\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)\boldsymbol{Q}_{-j}^T}{1+\frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n^T\boldsymbol{Q}_{-j}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j}\boldsymbol{U}_n^T\frac{\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}}{m}-\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\frac{\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T}{m}\\
&=\boldsymbol{Q}_m^T\boldsymbol{U}_n^T\frac{\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}}{m}\\
&\quad+\boldsymbol{Q}_m^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)\boldsymbol{Q}_m^T\left(1+\frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n^T\boldsymbol{Q}_{-j}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j\right)\boldsymbol{U}_n^T\frac{\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}}{m}\\
&\quad-\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\frac{\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T}{m}\\
&=\boldsymbol{Q}_m^T\boldsymbol{U}_n^T\frac{\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}}{m}+\frac{\boldsymbol{Q}_m^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)\boldsymbol{Q}_m^T}{1-\frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n^T\boldsymbol{Q}_m(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j}\boldsymbol{U}_n^T\frac{\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}}{m}-\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\frac{\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T}{m}.
\end{aligned}$$

From above, we expand Z_{3112} as

$$\begin{aligned}
&Z_{3112}\\
&=\frac{1}{nm^2}\frac{1}{1+\delta}\sum_{j=1}^{N}\mathbb{E}\left[\boldsymbol{r}^T\frac{\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\Sigma}_{\mathcal{S}}^{-jT}\boldsymbol{\Sigma}_{\mathcal{S}}^{-j}\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}\boldsymbol{D}_j}{1+\frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j}\boldsymbol{r}\right]\\
&=\underbrace{\frac{1}{nm^2}\frac{1}{1+\delta}\mathbb{E}\left[\boldsymbol{r}^T\boldsymbol{Q}_m^T\boldsymbol{U}_n^T\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{D}\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_m\boldsymbol{r}\right]}_{=Z_{31121}}\\
&+\underbrace{\sum_{j=1}^{N}\mathbb{E}\left[\boldsymbol{r}^T\frac{\boldsymbol{Q}_m^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)\boldsymbol{Q}_m^T\boldsymbol{U}_n^T\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_m\boldsymbol{D}_j}{nm^2(1+\delta)\left(1-\frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n^T\boldsymbol{Q}_m(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j\right)}\boldsymbol{r}\right]}_{=Z_{31122}}\\
&-\underbrace{\frac{1}{nm^2}\frac{1}{1+\delta}\sum_{j=1}^{N}\mathbb{E}\left[\boldsymbol{r}^T\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\sigma}_j\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_m\boldsymbol{D}_j\left(1+\frac{1}{m}\boldsymbol{\sigma}_j^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_j\right)\boldsymbol{r}\right]}_{=Z_{31123}}.
\end{aligned}$$

We have $Z_{31121}=\mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$ since $\mathbb{E}\left[\|\boldsymbol{D}\|\right]=\mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$ and from Lemma 7.4.1. Subsequently, we rewrite Z_{31122} as

$$Z_{31122}=\frac{1}{nm}\frac{1}{1+\delta}\mathbb{E}\left[\boldsymbol{r}^T\boldsymbol{Q}_m^T\boldsymbol{U}_n^T\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{D}_{31122}\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_m\boldsymbol{r}\right],$$

with $\boldsymbol{D}_{31122} \in \mathbb{R}^{N\times N}$ a diagonal matrix for which, for all $j \in [N]$, we have

$$[\boldsymbol{D}_{31122}]_j = \frac{1}{m}\frac{\boldsymbol{D}_j \boldsymbol{\sigma}_j^T (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)\boldsymbol{Q}_m^T \boldsymbol{U}_n^T \boldsymbol{\Sigma}_{\mathcal{S}}^T \boldsymbol{\Sigma}_{\mathcal{S}} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\sigma}_j}{1 - \frac{1}{m}\boldsymbol{\sigma}_j^T \boldsymbol{U}_n^T \boldsymbol{Q}_m (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_j}.$$

It can be shown that $\mathbb{E}\left[\|\boldsymbol{D}_{31122}\|\right] = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$, and we can deduce $Z_{31122} = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$. Similarly, we have $Z_{31123} = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$. Z_{3112} vanishes, and thus

$$\begin{aligned} Z_{311} &= Z_{3111} + \mathcal{O}\left(\frac{1}{\sqrt{m}}\right) \\ &= \frac{1}{nm^2}\frac{1}{1+\delta}\sum_{j=1}^N \mathbb{E}\left[\boldsymbol{r}^T \boldsymbol{Q}_{-j}^T \boldsymbol{U}_n^T \boldsymbol{\Sigma}_{\mathcal{S}}^{-jT} \boldsymbol{\Sigma}_{\mathcal{S}}^{-j} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\sigma}_j \boldsymbol{\sigma}_j^T \boldsymbol{U}_n \boldsymbol{Q}_{-j} \boldsymbol{r}\right] + \mathcal{O}\left(\frac{1}{\sqrt{m}}\right). \end{aligned}$$

It remains to handle Z_{3111} for which we have from the Sherman identity (Lemma A.8.3),

$$\begin{aligned} Z_{3111} &= \frac{1}{nm^2}\frac{1}{1+\delta}\sum_{j=1}^N \mathbb{E}\left[\boldsymbol{r}^T \boldsymbol{Q}_{-j}^T \boldsymbol{U}_n^T \boldsymbol{\Sigma}_{\mathcal{S}}^{-jT} \boldsymbol{\Sigma}_{\mathcal{S}}^{-j} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\sigma}_j \boldsymbol{\sigma}_j^T \boldsymbol{U}_n \boldsymbol{Q}_{-j} \boldsymbol{r}\right] \\ &= \frac{1}{nm^2}\frac{1}{1+\delta}\sum_{j=1}^N \sum_{i\neq j} \mathbb{E}\left[\boldsymbol{r}^T \boldsymbol{Q}_{-j}^T \boldsymbol{U}_n^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_{-j} \boldsymbol{r}\right] \\ &= \underbrace{\frac{1}{nm^2}\frac{1}{1+\delta}\sum_{j=1}^N \sum_{i\neq j} \mathbb{E}\left[\boldsymbol{r}^T \frac{\boldsymbol{Q}_{-ij}^T \boldsymbol{U}_n^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_{-ij}}{1 + \frac{1}{m}\boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-ij} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_i} \boldsymbol{r}\right]}_{=Z_{31111}} \\ &\quad - \underbrace{\frac{1}{nm^3}\frac{1}{1+\delta}\sum_{j=1}^N \sum_{i\neq j} \mathbb{E}\left[\boldsymbol{r}^T \frac{\boldsymbol{Q}_{-ij}^T \boldsymbol{U}_n^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_{-ij} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-ij}}{\left(1 + \frac{1}{m}\boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-ij} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_i\right)^2} \boldsymbol{r}\right]}_{=Z_{31112}}. \end{aligned}$$

Again, we replace $1 + \frac{1}{m}\boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-ij} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_i$ by $1+\delta$ in Z_{31111} as following

$$\begin{aligned} Z_{31111} &= \frac{1}{nm^2}\frac{1}{1+\delta}\sum_{j=1}^N \sum_{i\neq j} \mathbb{E}\left[\boldsymbol{r}^T \frac{\boldsymbol{Q}_{-ij}^T \boldsymbol{U}_n^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_{-ij}}{1 + \frac{1}{m}\boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-ij} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_i} \boldsymbol{r}\right] \\ &= \frac{1}{nm^2}\frac{1}{(1+\delta)^2}\sum_{j=1}^N \sum_{i\neq j} \mathbb{E}\left[\boldsymbol{r}^T \boldsymbol{Q}_{-ij}^T \boldsymbol{U}_n^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_{-ij} \boldsymbol{r}\right] \\ &\quad + \frac{1}{nm^2}\frac{1}{(1+\delta)^2}\sum_{j=1}^N \sum_{i\neq j} \mathbb{E}\left[\boldsymbol{r}^T \frac{\boldsymbol{Q}_{-ij}^T \boldsymbol{U}_n^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_{-ij} \boldsymbol{D}_i}{1 + \frac{1}{m}\boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-ij} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_i} \boldsymbol{r}\right] \\ &= \frac{1}{n}\frac{N^2}{m^2}\frac{1}{(1+\delta)^2}\mathbb{E}\left[\boldsymbol{r}^T \boldsymbol{Q}_{--}^T \boldsymbol{U}_n^T \boldsymbol{\Phi}_{\mathcal{S}} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_{--} \boldsymbol{r}\right] \\ &\quad + \frac{1}{nm^2}\frac{1}{(1+\delta)^2}\sum_{j=1}^N \sum_{i\neq j} \mathbb{E}\left[\boldsymbol{r}^T \boldsymbol{Q}_{-j}^T \boldsymbol{U}_n^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_{-j} \boldsymbol{D}_i \boldsymbol{r}\right] \\ &\quad + \frac{1}{nm^3}\frac{1}{(1+\delta)^2}\sum_{j=1}^N \sum_{i\neq j} \mathbb{E}\left[\boldsymbol{r}^T \frac{\boldsymbol{Q}_{-j}^T \boldsymbol{U}_n^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Phi}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_{-j} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-j}}{1 - \frac{1}{m}\boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-j} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_i} \boldsymbol{D}_i \boldsymbol{r}\right] \\ &= \frac{1}{n}\mathbb{E}\left[\boldsymbol{r}^T \boldsymbol{Q}_{--}^T \boldsymbol{U}_n^T \boldsymbol{\Psi}_{\mathcal{S}} \boldsymbol{\Lambda}_{\boldsymbol{P}} \boldsymbol{\Psi}_{\mathcal{S}} \boldsymbol{U}_n \boldsymbol{Q}_{--} \boldsymbol{r}\right] \end{aligned}$$

$$+\frac{1}{nm^2}\frac{1}{(1+\delta)^2}\sum_{j=1}^{N}\mathbb{E}\left[\boldsymbol{r}^T\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\Sigma}_{\mathcal{S}}^{-jT}\boldsymbol{D}\boldsymbol{\Sigma}_{\mathcal{S}}^{-j}\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Phi}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_{-j}\boldsymbol{r}\right]$$
$$+\frac{1}{nm}\frac{1}{(1+\delta)^2}\sum_{j=1}^{N}\mathbb{E}\left[\boldsymbol{r}^T\boldsymbol{Q}_{-j}^T\boldsymbol{U}_n^T\boldsymbol{\Sigma}_{\mathcal{S}}^{-jT}\boldsymbol{D}_{31111}\boldsymbol{\Sigma}_{\mathcal{S}}^{-j}\boldsymbol{U}_n\boldsymbol{Q}_{-j}\boldsymbol{r}\right],$$

where $\boldsymbol{D}_{31111}\in\mathbb{R}^{N\times N}$ is a diagonal matrix for which, for all $i\in[N]$, we have

$$[\boldsymbol{D}_{31111}]_i=\frac{1}{m}\frac{\boldsymbol{\sigma}_i^T\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Phi}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_{-j}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_i\left(\delta-\frac{1}{m}\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-ij}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_i\right)}{1-\frac{1}{m}\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-j}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_i},$$

and $\boldsymbol{Q}_{--}$ is a resolvent with the same law than $\boldsymbol{Q}_{-ij}$. With similar arguments that before, we can show that $\mathbb{E}\big[\|\boldsymbol{D}\|\big]$ and $\mathbb{E}\big[\|\boldsymbol{D}_{31111}\|\big]$ are of order $\mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$, and therefore

$$Z_{31111}=\frac{1}{n}\mathbb{E}\left[\boldsymbol{r}^T\boldsymbol{Q}_{--}^T\boldsymbol{U}_n^T\boldsymbol{\Psi}_{\mathcal{S}}\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Psi}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_{--}\boldsymbol{r}\right]+\mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

By extending Lemma A.2.2 to the matrix $\boldsymbol{\Lambda}'_{\boldsymbol{P}}=\boldsymbol{U}_n^T\boldsymbol{\Psi}_{\mathcal{S}}\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Psi}_{\mathcal{S}}\boldsymbol{U}_n$, and from Lemma A.2.4 we obtain

$$\begin{aligned}Z_{31111}&=\frac{1}{n}\boldsymbol{r}^T\bar{\boldsymbol{Q}}_m^T\boldsymbol{U}_n^T\boldsymbol{\Psi}_{\mathcal{S}}\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Psi}_{\mathcal{S}}\boldsymbol{U}_n\bar{\boldsymbol{Q}}_m\boldsymbol{r}\\&\quad+\frac{1}{n}\frac{\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Theta}_{\mathcal{S}}\boldsymbol{\Psi}_2\boldsymbol{\Theta}_{\mathcal{S}}^T\right)}{1-\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m\right)}\|\bar{\boldsymbol{Q}}_m\boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2\\&\quad+\mathcal{O}\left(\frac{1}{\sqrt{m}}\right).\end{aligned}$$

Following the same reasoning for Z_{31112}, and from Lemma A.7.2, we have

$$\begin{aligned}Z_{31112}&=\frac{1}{nm^3}\frac{1}{1+\delta}\sum_{j=1}^{N}\sum_{i\neq j}\mathbb{E}\left[\boldsymbol{r}^T\frac{\boldsymbol{Q}_{-ij}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Phi}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_{-ij}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-ij}}{\left(1+\frac{1}{m}\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-ij}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_i\right)^2}\boldsymbol{r}\right]\\&=\frac{1}{nm^3}\frac{1}{(1+\delta)^3}\sum_{j=1}^{N}\sum_{i\neq j}\mathbb{E}\left[\boldsymbol{r}^T\boldsymbol{Q}_{-ij}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Phi}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_{-ij}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-ij}\boldsymbol{r}\right]\\&\quad+\mathcal{O}\left(\frac{1}{\sqrt{m}}\right)\\&=\frac{1}{nm^2}\frac{1}{(1+\delta)^3}\sum_{j=1}^{N}\sum_{i\neq j}\mathbb{E}\left[\boldsymbol{r}^T\boldsymbol{Q}_{-ij}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-ij}\boldsymbol{r}\left(\frac{1}{m}\boldsymbol{\sigma}_i^T\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Phi}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_{-ij}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_i\right)\right]\\&\quad+\mathcal{O}\left(\frac{1}{\sqrt{m}}\right)\\&=\frac{1}{nm^2}\frac{1}{(1+\delta)^3}\sum_{j=1}^{N}\sum_{i\neq j}\mathbb{E}\left[\boldsymbol{r}^T\boldsymbol{Q}_{-ij}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-ij}\boldsymbol{r}\frac{1}{m}\operatorname{Tr}(\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Phi}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_{-ij}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\Phi}_{\mathcal{S}})\right]\\&\quad+\mathcal{O}\left(\frac{1}{\sqrt{m}}\right)\\&=\frac{1}{n}\frac{1}{N}\mathbb{E}\left[\boldsymbol{r}^T\boldsymbol{Q}_{--}^T\boldsymbol{\Psi}_1\boldsymbol{Q}_{--}\boldsymbol{r}\operatorname{Tr}(\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Psi}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_{--}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\Psi}_{\mathcal{S}})\right]\\&\quad+\mathcal{O}\left(\frac{1}{\sqrt{m}}\right),\end{aligned}$$

where the last equality is obtained with similar reasoning than for equation A.43. From Lemma A.2.4 and Lemma A.2.5, we have

$$Z_{31112} = \frac{1}{n} \frac{\frac{1}{N}\operatorname{Tr}(\boldsymbol{\Lambda_P}\boldsymbol{\Theta}_{\mathcal{S}}(\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\Psi}_{\mathcal{S}})}{1 - \frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2 \bar{\boldsymbol{Q}}_m^T \boldsymbol{\Psi}_1 \bar{\boldsymbol{Q}}_m\right)} \|\bar{\boldsymbol{Q}}_m \boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2 + \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

We conclude for Z_{311} that

$$\begin{aligned} Z_{311} = &\frac{1}{n} \boldsymbol{r}^T \bar{\boldsymbol{Q}}_m^T \boldsymbol{U}_n^T \boldsymbol{\Psi}_{\mathcal{S}} \boldsymbol{\Lambda_P} \boldsymbol{\Psi}_{\mathcal{S}} \boldsymbol{U}_n \bar{\boldsymbol{Q}}_m \boldsymbol{r} \\ &+ \frac{1}{n} \frac{\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Lambda_P}\left[\boldsymbol{\Theta}_{\mathcal{S}} \boldsymbol{\Psi}_2 \boldsymbol{\Theta}_{\mathcal{S}}^T - \boldsymbol{\Theta}_{\mathcal{S}}(\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\Psi}_{\mathcal{S}}\right]\right)}{1 - \frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2 \bar{\boldsymbol{Q}}_m^T \boldsymbol{\Psi}_1 \bar{\boldsymbol{Q}}_m\right)} \|\bar{\boldsymbol{Q}}_m \boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2 \\ &+ \mathcal{O}\left(\frac{1}{\sqrt{m}}\right), \end{aligned}$$

and for Z_{31} that

$$\begin{aligned} Z_{31} = &\frac{1}{n} \boldsymbol{r}^T \bar{\boldsymbol{Q}}_m^T \boldsymbol{U}_n^T \boldsymbol{\Psi}_{\mathcal{S}} \boldsymbol{\Lambda_P} \boldsymbol{\Psi}_{\mathcal{S}} \boldsymbol{U}_n \bar{\boldsymbol{Q}}_m \boldsymbol{r} \\ &+ \frac{1}{n} \frac{\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Lambda_P}\left[\boldsymbol{\Theta}_{\mathcal{S}} \boldsymbol{\Psi}_2 \boldsymbol{\Theta}_{\mathcal{S}}^T - 2\boldsymbol{\Theta}_{\mathcal{S}}(\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\Psi}_{\mathcal{S}}\right]\right)}{1 - \frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2 \bar{\boldsymbol{Q}}_m^T \boldsymbol{\Psi}_1 \bar{\boldsymbol{Q}}_m\right)} \|\bar{\boldsymbol{Q}}_m \boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2 \\ &+ \mathcal{O}\left(\frac{1}{\sqrt{m}}\right). \end{aligned}$$

□

Lemma A.3.5. *Under Assumptions 1, 2 and 3, let $Z_{32} \in \mathbb{R}$ defined in equation A.42 as*

$$Z_{32} = \sum_{i=1}^{N} \mathbb{E}\left[\frac{1}{nm^2} \boldsymbol{r}^T \frac{\boldsymbol{Q}_{-i}^T \boldsymbol{U}_n^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{\Lambda_P} \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-i}}{\left(1 + \frac{1}{m}\boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-i} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_i\right)^2} \boldsymbol{r}\right].$$

Then

$$\left| Z_{32} - \frac{1}{n} \frac{\frac{1}{N}\operatorname{Tr}(\boldsymbol{\Lambda_P}\boldsymbol{\Psi}_{\mathcal{S}})}{1 - \frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2 \bar{\boldsymbol{Q}}_m^T \boldsymbol{\Psi}_1 \bar{\boldsymbol{Q}}_m\right)} \|\bar{\boldsymbol{Q}}_m \boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2 \right| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right),$$

where $\bar{\boldsymbol{Q}}_m$ is the deterministic resolent defined in equation 7.7, $\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2 \in \mathbb{R}^{n \times n}$ are defined in equation 7.12, and $\boldsymbol{\Psi}_{\mathcal{S}} \in \mathbb{R}^{p \times p}$ is defined in equation 7.15.

Proof. We decompose Z_{32} as

$$\begin{aligned} Z_{32} &= \frac{1}{nm^2} \sum_{i=1}^{N} \mathbb{E}\left[\boldsymbol{r}^T \frac{\boldsymbol{Q}_{-i}^T \boldsymbol{U}_n^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{\Lambda_P} \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-i}}{\left(1 + \frac{1}{m}\boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-i} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_i\right)^2} \boldsymbol{r}\right] \\ &= \frac{1}{nm} \sum_{i=1}^{N} \mathbb{E}\left[\boldsymbol{r}^T \frac{\boldsymbol{Q}_{-i}^T \boldsymbol{U}_n^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-i} \frac{1}{m}\operatorname{Tr}\left(\boldsymbol{\Phi}_{\mathcal{S}} \boldsymbol{\Lambda_P}\right)}{\left(1 + \frac{1}{m}\boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-i} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_i\right)^2} \boldsymbol{r}\right] \\ &\quad + \frac{1}{nm} \sum_{i=1}^{N} \mathbb{E}\left[\boldsymbol{r}^T \frac{\boldsymbol{Q}_{-i}^T \boldsymbol{U}_n^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-i} \frac{1}{m}\left(\boldsymbol{\sigma}_i^T \boldsymbol{\Lambda_P} \boldsymbol{\sigma}_i - \operatorname{Tr}\left(\boldsymbol{\Phi}_{\mathcal{S}} \boldsymbol{\Lambda_P}\right)\right)}{\left(1 + \frac{1}{m}\boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-i} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_i\right)^2} \boldsymbol{r}\right] \\ &= \frac{1}{n} \frac{\operatorname{Tr}\left(\boldsymbol{\Phi}_{\mathcal{S}} \boldsymbol{\Lambda_P}\right)}{m} \boldsymbol{r}^T \underbrace{\sum_{i=1}^{N} \mathbb{E}\left[\frac{1}{m} \frac{\boldsymbol{Q}_{-i}^T \boldsymbol{U}_n^T \boldsymbol{\sigma}_i \boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-i}}{\left(1 + \frac{1}{m}\boldsymbol{\sigma}_i^T \boldsymbol{U}_n \boldsymbol{Q}_{-i} (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\sigma}_i\right)^2}\right]}_{= \boldsymbol{Z}_{321}} \boldsymbol{r} \end{aligned}$$

$$+\frac{1}{n}\boldsymbol{r}^T\underbrace{\sum_{i=1}^{N}\mathbb{E}\left[\frac{1}{m}\boldsymbol{Q}_m^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_m\frac{1}{m}\left(\boldsymbol{\sigma}_i^T\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\sigma}_i-\operatorname{Tr}\left(\boldsymbol{\Phi}_{\mathcal{S}}\boldsymbol{\Lambda}_{\boldsymbol{P}}\right)\right)\right]}_{\boldsymbol{Z}_{322}}\boldsymbol{r}.$$

We want to show $\boldsymbol{Z}_{322}$ vanishes and find an asymptotic equivalent for $\boldsymbol{Z}_{321}$. Let $\boldsymbol{D}_{322}\in\mathbb{R}^{N\times N}$ be a diagonal matrix for which, for all $i\in[N]$, we have

$$\left[\boldsymbol{D}_{322}\right]_i=\frac{1}{m}\boldsymbol{\sigma}_i^T\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\sigma}_i-\frac{1}{m}\operatorname{Tr}\left(\boldsymbol{\Phi}_{\mathcal{S}}\boldsymbol{\Lambda}_{\boldsymbol{P}}\right).$$

We rewrite $\boldsymbol{Z}_{322}$ as

$$\begin{aligned}\boldsymbol{Z}_{322}&=\sum_{i=1}^{N}\mathbb{E}\left[\frac{1}{m}\boldsymbol{Q}_m^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_m\frac{1}{m}\left(\boldsymbol{\sigma}_i^T\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\sigma}_i-\operatorname{Tr}\left(\boldsymbol{\Phi}_{\mathcal{S}}\boldsymbol{\Lambda}_{\boldsymbol{P}}\right)\right)\right]\\&=\mathbb{E}\left[\frac{1}{m}\boldsymbol{Q}_m^T\boldsymbol{U}_n^T\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{D}_{322}\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_m\right]\end{aligned}$$

From Lemma A.4.4, we know there exists a real $K'_{\boldsymbol{Q}}>0$ such that, for all m, we have

$$\left\|\frac{1}{\sqrt{m}}\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_m\right\|\leq K'_{\boldsymbol{Q}}$$

and

$$\left\|\frac{1}{\sqrt{m}}\boldsymbol{Q}_m(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\Sigma}_{\mathcal{S}}^T\right\|\leq 2K'_{\boldsymbol{Q}}.$$

Using Lemma A.7.2 we show that $\mathbb{E}\left[\|\boldsymbol{D}_{322}\|\right]=\mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$, and we deduce that

$$\|\boldsymbol{Z}_{322}\|=\mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

We want to find an asymptotic equivalent for $\boldsymbol{Z}_{321}$. Let $\boldsymbol{D}_{321}\in\mathbb{R}^{N\times N}$ be a diagonal matrix for which, for all $i\in[N]$, we have

$$\left[\boldsymbol{D}_{321}\right]_i=(1+\delta)^2-\left(1+\frac{1}{m}\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-i}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_i\right)^2.$$

We replace $1+\frac{1}{m}\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-i}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_i$ by $1+\delta$ as following

$$\begin{aligned}\boldsymbol{Z}_{321}&=\sum_{i=1}^{N}\mathbb{E}\left[\frac{1}{m}\frac{\boldsymbol{Q}_{-i}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-i}}{\left(1+\frac{1}{m}\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-i}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_i\right)^2}\right]\\&=\frac{1}{(1+\delta)^2}\sum_{i=1}^{N}\mathbb{E}\left[\frac{1}{m}\boldsymbol{Q}_{-i}^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-i}\right]\\&+\frac{1}{(1+\delta)^2}\sum_{i=1}^{N}\mathbb{E}\left[\frac{1}{m}\boldsymbol{Q}_m^T\boldsymbol{U}_n^T\boldsymbol{\sigma}_i\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_m\left((1+\delta)^2-\left(1+\frac{1}{m}\boldsymbol{\sigma}_i^T\boldsymbol{U}_n\boldsymbol{Q}_{-i}(\boldsymbol{U}_n-\gamma\boldsymbol{V}_n)^T\boldsymbol{\sigma}_i\right)^2\right)\right]\\&=\frac{N}{m}\frac{1}{(1+\delta)^2}\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{U}_n^T\boldsymbol{\Phi}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_-\right]+\frac{1}{(1+\delta)^2}\mathbb{E}\left[\frac{1}{m}\boldsymbol{Q}_m\boldsymbol{U}_n^T\boldsymbol{\Sigma}_{\mathcal{S}}^T\boldsymbol{D}_{321}\boldsymbol{\Sigma}_{\mathcal{S}}\boldsymbol{U}_n\boldsymbol{Q}_m\right]\\&=\frac{1}{1+\delta}\mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{\Psi}_1\boldsymbol{Q}_-\right]+\mathcal{O}_{\|\cdot\|}\left(\frac{1}{\sqrt{m}}\right).\end{aligned}$$

The last equality is obtained since we can show that $\mathbb{E}\big[\|\boldsymbol{D}_{321}\|\big] = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right)$. We have from Lemma A.2.4

$$\left\|\mathbb{E}\left[\boldsymbol{Q}_m^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\right] - \mathbb{E}\left[\boldsymbol{Q}_-^T\boldsymbol{\Psi}_1\boldsymbol{Q}_-\right]\right\| = \mathcal{O}\left(\frac{1}{\sqrt{m}}\right),$$

and from Lemma A.2.5

$$\boldsymbol{Z}_{321} = \frac{1}{1+\delta}\mathbb{E}\big[\boldsymbol{Q}_m^T\boldsymbol{\Psi}_1\boldsymbol{Q}_m\big] = \frac{1}{1+\delta}\frac{1}{1-\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m\right)}\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m + \mathcal{O}_{\|\cdot\|}\left(\frac{1}{\sqrt{m}}\right).$$

We conclude that

$$Z_{32} = \frac{1}{n}\frac{\frac{1}{N}\operatorname{Tr}(\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Psi}_{\mathcal{S}})}{1-\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m\right)}\|\bar{\boldsymbol{Q}}_m\boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2 + \mathcal{O}\left(\frac{1}{\sqrt{m}}\right).$$

□

Lemma A.3.6. *When all states have been visited, the empirical transition model matrix* $\hat{\boldsymbol{A}}_m = \hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n) = \boldsymbol{U}_n(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)$ *defined in equation 6.14 is invertible.*

Proof. Let $c : \mathcal{S} \to \mathbb{N}$ and $c' : \mathcal{S} \to \mathbb{N}$ be defined such that, for all $i \in [p]$, $c(\boldsymbol{S}_i)$ and $c'(\boldsymbol{S}_i)$ represent the number of times $\boldsymbol{S}_i$ occurs in $\boldsymbol{X}_n$ and $\boldsymbol{X}'_n$, respectively. If all states have been visited ($m = p$), for all i in $[p]$, we have thus $c(\boldsymbol{S}_i) > 0$. The structure of $\sqrt{n}\boldsymbol{U}_n \in \mathbb{R}^{m\times n}$ indicates each column i of $\boldsymbol{U}_n$ is a one-hot vector, where its j-th element is 1 if the i-th state $\boldsymbol{s}_i$ of $\boldsymbol{X}_n$ is $\boldsymbol{S}_j$. Conversely, each row i of $\sqrt{n}\boldsymbol{U}_n$ has a j-th element is one if the j-th state $\boldsymbol{s}_j$ of $\boldsymbol{X}_n$ is $\boldsymbol{S}_i$. A similar correspondence holds for $\sqrt{n}\boldsymbol{V}_n$ and $\boldsymbol{X}'_n$. From interpretations of $\boldsymbol{U}_n$ and $\boldsymbol{V}_n$, we deduce $n\boldsymbol{U}_n\boldsymbol{U}_n^T \in \mathbb{R}^{m\times m}$ and $n\boldsymbol{V}_n\boldsymbol{V}_n^T \in \mathbb{R}^{m\times m}$ are diagonal matrices where the i-th element of its diagonal are $c(\boldsymbol{S}_i)$ and $c'(\boldsymbol{S}_i)$, respectively. In the same way, $n\boldsymbol{U}_n\boldsymbol{V}_n^T \in \mathbb{R}^{m\times m}$ is matrix for which $[n\boldsymbol{U}_n\boldsymbol{V}_n^T]_{ij}$ is $c(\boldsymbol{S}_i \to \boldsymbol{S}_j)$ which represents the number of times the state $\boldsymbol{S}_i$ follows $\boldsymbol{S}_j$ in $\mathcal{D}_{\text{train}}$. We are going to prove $\hat{\boldsymbol{A}}_m$ is invertible by using the Gershgorin circle theorem to show $\hat{\boldsymbol{A}}_m$ is strictly diagonally dominant, i.e., $|[\hat{\boldsymbol{A}}_m]_{ii}| > \sum_{i\neq j}|[\hat{\boldsymbol{A}}_m]_{ij}|$. From the interpretations of $\boldsymbol{U}_n\boldsymbol{U}_n^T$ and $\boldsymbol{U}_n\boldsymbol{V}_n^T$, we have

$$[\hat{\boldsymbol{A}}_m]_{ii} = [\boldsymbol{U}_n\boldsymbol{U}_n^T]_{ii} - \gamma[\boldsymbol{U}_n\boldsymbol{V}_n^T]_{ii} = \frac{c(\boldsymbol{S}_i) - \gamma c(\boldsymbol{S}_i \to \boldsymbol{S}_i)}{n} > 0, \qquad \forall i \in [n],$$

and

$$[\hat{\boldsymbol{A}}_m]_{ij} = -\gamma[\boldsymbol{U}_n\boldsymbol{V}_n^T]_{ij} = \frac{-\gamma c(\boldsymbol{S}_i \to \boldsymbol{S}_j)}{n} < 0, \qquad \forall i \neq j.$$

To prove $\hat{\boldsymbol{A}}_m$ is invertible it remains to show $\sum_j[\hat{\boldsymbol{A}}_m]_{ij} = \sum_j[\boldsymbol{U}_n(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T]_{ij} > 0$ for all $i \in [m]$. Let $i \in [m]$, we have

$$\sum_j[\boldsymbol{U}_n(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T]_{ij} = \frac{c(\boldsymbol{S}_i)}{n} - \gamma\sum_j\frac{c(\boldsymbol{S}_i \to \boldsymbol{S}_j)}{n} = (1-\gamma)\frac{c(\boldsymbol{S}_i)}{n} > 0,$$

which concludes the proof. □

Lemma A.3.7. *Let* Δ *be the second-order correction factor of* $\overline{\text{MSBE}}(\hat{\boldsymbol{\theta}}_n^\lambda)$ *defined in equation 7.14. If all states have been visited, then*

$$\Delta = \frac{\lambda^2}{n}\frac{\frac{1}{N}\operatorname{Tr}\Big(\boldsymbol{U}_n^T\hat{\boldsymbol{A}}_m^{-1T}\boldsymbol{\Lambda}_{\boldsymbol{P}}\hat{\boldsymbol{A}}_m^{-1}\boldsymbol{U}_n\bar{\boldsymbol{Q}}_m\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\Big)}{1-\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m(\lambda)^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m(\lambda)\right)}\|\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2,$$

where $\hat{\boldsymbol{A}}_m = \hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n) = \boldsymbol{U}_n(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)$ *is the empirical transition model matrix defined in equation 6.14.*

Proof. When all states have been visited, we have $\boldsymbol{U}_n = \hat{\boldsymbol{U}}_n$, $\boldsymbol{V}_n = \hat{\boldsymbol{V}}_n$ and $\boldsymbol{\Sigma}_{\mathcal{S}} = \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}$. Furthermore, from Lemma A.3.6, $\hat{\boldsymbol{A}}_m = \hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n) = \boldsymbol{U}_n(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)$ is invertible. We write

$$\begin{aligned}\boldsymbol{\Theta}_{\mathcal{S}} = \boldsymbol{\Psi}_{\mathcal{S}}\boldsymbol{U}_n\bar{\boldsymbol{Q}}_m(\lambda) &= \hat{\boldsymbol{A}}_m^{-1}\boldsymbol{U}_n(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\Psi}_{\mathcal{S}}\boldsymbol{U}_n\bar{\boldsymbol{Q}}_m(\lambda)\\ &= \hat{\boldsymbol{A}}_m^{-1}\boldsymbol{U}_n\big[\boldsymbol{I}_n - \lambda\bar{\boldsymbol{Q}}_m\big]\end{aligned}$$

Using the equality above and the cyclic properties of the trace, we conclude that

$$\begin{aligned}&\operatorname{Tr}\Big(\boldsymbol{\Lambda}_{\boldsymbol{P}}\big[\boldsymbol{\Theta}_{\mathcal{S}}\boldsymbol{\Psi}_2\boldsymbol{\Theta}_{\mathcal{S}}^T - 2\boldsymbol{\Theta}_{\mathcal{S}}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\Psi}_{\mathcal{S}} + \boldsymbol{\Psi}_{\mathcal{S}}\big]\Big)\\
&= \operatorname{Tr}\Big(\boldsymbol{\Lambda}_{\boldsymbol{P}}\big[\hat{\boldsymbol{A}}_m^{-1}\boldsymbol{U}_n\big[\boldsymbol{I}_n - \lambda\bar{\boldsymbol{Q}}_m\big]\boldsymbol{\Psi}_2\big[\boldsymbol{I}_n - \lambda\bar{\boldsymbol{Q}}_m\big]^T\boldsymbol{U}_n^T\hat{\boldsymbol{A}}_m^{-1T}\\
&\quad - 2\hat{\boldsymbol{A}}_m^{-1}\boldsymbol{U}_n\big[\boldsymbol{I}_n - \lambda\bar{\boldsymbol{Q}}_m\big](\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\Psi}_{\mathcal{S}} + \boldsymbol{\Psi}_{\mathcal{S}}\big]\Big)\\
&= \lambda^2\operatorname{Tr}\Big(\boldsymbol{\Lambda}_{\boldsymbol{P}}\hat{\boldsymbol{A}}_m^{-1}\boldsymbol{U}_n\bar{\boldsymbol{Q}}_m\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{U}_n^T\hat{\boldsymbol{A}}_m^{-1T}\Big) - \lambda\operatorname{Tr}\Big(\boldsymbol{\Lambda}_{\boldsymbol{P}}\hat{\boldsymbol{A}}_m^{-1}\boldsymbol{U}_n\bar{\boldsymbol{Q}}_m(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\Psi}_{\mathcal{S}}\Big)\\
&\quad - \lambda\operatorname{Tr}\Big(\boldsymbol{\Lambda}_{\boldsymbol{P}}\boldsymbol{\Psi}_{\mathcal{S}}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)\boldsymbol{Q}_m^T\boldsymbol{U}_n^T\hat{\boldsymbol{A}}_m^{-1T}\Big) + 2\lambda\operatorname{Tr}\Big(\boldsymbol{\Lambda}_{\boldsymbol{P}}\hat{\boldsymbol{A}}_m^{-1}\boldsymbol{U}_n\bar{\boldsymbol{Q}}_m(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\Psi}_{\mathcal{S}}\Big)\\
&= \lambda^2\operatorname{Tr}\Big(\boldsymbol{\Lambda}_{\boldsymbol{P}}\hat{\boldsymbol{A}}_m^{-1}\boldsymbol{U}_n\bar{\boldsymbol{Q}}_m\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\boldsymbol{U}_n^T\hat{\boldsymbol{A}}_m^{-1T}\Big)\\
&= \lambda^2\operatorname{Tr}\Big(\boldsymbol{U}_n^T\hat{\boldsymbol{A}}_m^{-1T}\boldsymbol{\Lambda}_{\boldsymbol{P}}\hat{\boldsymbol{A}}_m^{-1}\boldsymbol{U}_n\bar{\boldsymbol{Q}}_m\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m^T\Big).\end{aligned}$$

□

A.4 Technical Details on the Resolvent $\boldsymbol{Q}_m(\lambda)$

This section aims to prove that the operator norm of $\boldsymbol{Q}_m(\lambda)$ is uniformly upper bounded under Assumption 2. Indeed, controlling the operator norm of $\boldsymbol{Q}_m(\lambda)$ is crucial for proving the theorems in Chapter 7. When $\gamma = 0$, which corresponds to the supervised learning case on the reward function, the result is straightforward with Lemma A.8.6 since $\frac{1}{m}\boldsymbol{\Sigma}_{\boldsymbol{X}_n}^T\boldsymbol{\Sigma}_{\boldsymbol{X}_n}$ is positive-definite (Louart et al., 2018; Liao et al., 2020). In the RL setting, the conclusion is less straightforward as the resolvent is no longer that of a symmetric positive-definite matrix. This issue is further exacerbated by the lack of results in the literature concerning the upper bounds for operator norm of resolvents of non-positive-definite matrices. Lemma A.4.1 aims to propose a solution for the RL setting under Assumptions 1 and 2. Proof of the widely used Lemma A.4.4 is also presented at the end of this section.

Lemma A.4.1. *Under Assumptions 1 and 2, let* $\lambda > 0$ *and let* $\boldsymbol{Q}_m(\lambda) \in \mathbb{R}^{n\times n}$ *be the resolvent defined in equation 6.13 as*

$$\boldsymbol{Q}_m(\lambda) = \left[\frac{1}{m}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n + \lambda\boldsymbol{I}_n\right]^{-1}.$$

Then there exists a real $K > 0$ *such that, for all* m*, we have*

$$\|\boldsymbol{Q}_m(\lambda)\| \leq K.$$

Proof. Under Assumption 2, the empirical transition model matrix $\hat{\boldsymbol{A}}_m = \hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T$ (equation 6.14) is invertible since the symmetric part of $\hat{\boldsymbol{A}}_m$ is positive-definite. Let

$$0 < \epsilon < \lambda \min\left\{\frac{1}{\xi_{\max}}, \frac{\xi_{\min}}{4}\right\},$$

for $\xi_{\min}, \xi_{\max} > 0$ defined in Assumption 2. We rewrite equation 6.13 as

$$\begin{aligned}\boldsymbol{Q}_m(\lambda) &= \left[\frac{1}{m}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n + \lambda\boldsymbol{I}_n\right]^{-1} \\ &= \left[(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\left[\frac{1}{m}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} + \epsilon\boldsymbol{I}_m\right]\hat{\boldsymbol{U}}_n + \underbrace{\lambda\boldsymbol{I}_n - \epsilon(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\hat{\boldsymbol{U}}_n}_{=\boldsymbol{B}_n}\right]^{-1}.\end{aligned}$$

To apply the Woodbury identity (Lemma A.8.2) on $\boldsymbol{Q}_m(\lambda)$, we check that both

$$\boldsymbol{B}_n = \lambda\boldsymbol{I}_n - \epsilon(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\hat{\boldsymbol{U}}_n,$$

and

$$\begin{aligned}\boldsymbol{M}_m &= \left[\frac{1}{m}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} + \epsilon\boldsymbol{I}_m\right]^{-1} + \hat{\boldsymbol{U}}_n\boldsymbol{B}_n^{-1}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \\ &= \left[\frac{1}{m}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} + \epsilon\boldsymbol{I}_m\right]^{-1} + \left[\lambda\boldsymbol{I}_n - \epsilon\hat{\boldsymbol{A}}_m\right]^{-1}\hat{\boldsymbol{A}}_m \\ &= \left[\frac{1}{m}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} + \epsilon\boldsymbol{I}_m\right]^{-1} + \left[\lambda\hat{\boldsymbol{A}}_m^{-1} - \epsilon\boldsymbol{I}_m\right]^{-1}\end{aligned}$$

are non-singular, since $\frac{1}{m}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} + \epsilon\boldsymbol{I}_m$ is non-singular. Given that $H(\hat{\boldsymbol{A}}_m)$ is positive-definite, $\hat{\boldsymbol{A}}_m$ has eigenvalues with positive real parts. Consequently, by the Weinstein–Aronszajn identity (Lemma A.8.5), $(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\hat{\boldsymbol{U}}_n$ has non-zero eigenvalues with positive real parts. As $\epsilon < \frac{\lambda}{\xi_{\max}} \leq \frac{\lambda}{\nu_{\max}(H(\hat{\boldsymbol{A}}_m))} \leq \frac{\lambda}{\mathrm{Re}(\nu_{\max}(\hat{\boldsymbol{A}}_m))}$, we deduce that the matrix $\boldsymbol{B}_n = \lambda\boldsymbol{I}_n - \epsilon(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\hat{\boldsymbol{U}}_n$ has eigenvalues with positive real parts and is non-singular. To prove that the matrix $\boldsymbol{M}_m$ is non-singular, we propose to show $\boldsymbol{x}^T\boldsymbol{M}_m\boldsymbol{x} > 0$ for all non-zero $\boldsymbol{x} \in \mathbb{R}^m$. Since $\left[\frac{1}{m}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} + \epsilon\boldsymbol{I}_m\right]^{-1}$ is at least positive-semi-definite, the statement $\boldsymbol{x}^T\boldsymbol{M}_m\boldsymbol{x} > 0$ for all non-zero $\boldsymbol{x} \in \mathbb{R}^m$ may be restated as

$$\begin{aligned}&\text{for all non-zero } \boldsymbol{x} \in \mathbb{R}^m, \quad \boldsymbol{x}^T\left[\lambda\hat{\boldsymbol{A}}_m^{-1} - \epsilon\boldsymbol{I}_m\right]^{-1}\boldsymbol{x} > 0 \\ \text{iff} \quad &\text{for all non-zero } \boldsymbol{x} \in \mathbb{R}^m, \quad \boldsymbol{x}^T\left[\lambda\hat{\boldsymbol{A}}_m^{-1} - \epsilon\boldsymbol{I}_m\right]\boldsymbol{x} > 0 \\ \text{iff} \quad &\text{for all non-zero } \boldsymbol{x} \in \mathbb{R}^m, \quad \boldsymbol{x}^T H\big(\hat{\boldsymbol{A}}_m^{-1}\big)\boldsymbol{x} - \frac{\epsilon}{\lambda}\boldsymbol{x}^T\boldsymbol{x} > 0 \\ \text{iff} \quad &\nu_{\min}\big(H(\hat{\boldsymbol{A}}_m^{-1})\big) > \frac{\epsilon}{\lambda}.\end{aligned}$$

By construction of $\hat{\boldsymbol{U}}_n$ and $\hat{\boldsymbol{V}}_n$, we have both $\|\hat{\boldsymbol{U}}_n\| \leq 1$ and $\|\hat{\boldsymbol{V}}_n\| \leq 1$. We deduce thus

$$\big\|\hat{\boldsymbol{A}}_m\big\| = \big\|\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\big\| < 2.$$

Since $H(\hat{\boldsymbol{A}}_m^{-1}) = [\hat{\boldsymbol{A}}_m^{-1}]^T H(\hat{\boldsymbol{A}}_m)\hat{\boldsymbol{A}}_m^{-1}$, we deduce from Ostrowski's Theorem (Lemma A.8.4) that

$$\nu_{\min}\big(H(\hat{\boldsymbol{A}}_m^{-1})\big) \geq \frac{\nu_{\min}\big(H(\hat{\boldsymbol{A}}_m)\big)}{\|\hat{\boldsymbol{A}}_m\|^2} \geq \frac{\xi_{\min}}{4}.$$

Since $\epsilon < \frac{\lambda \xi_{\min}}{4}$, we have $\boldsymbol{x}^T \boldsymbol{M}_m \boldsymbol{x} > 0$ for all non-zero $\boldsymbol{x} \in \mathbb{R}^m$, and thus $\boldsymbol{M}_m$ is non-singular. As a consequence, we apply the Woodbury identity (Lemma A.8.2) on the resolvent $\boldsymbol{Q}_m(\lambda)$ to get

$$\begin{aligned}\boldsymbol{Q}_m(\lambda) &= \left[(\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \left[\frac{1}{m} \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} + \epsilon \boldsymbol{I}_m\right] \hat{\boldsymbol{U}}_n + \boldsymbol{B}_n\right]^{-1} \\ &= \boldsymbol{B}_n^{-1} - \boldsymbol{B}_n^{-1} (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{M}_m^{-1} \hat{\boldsymbol{U}}_n \boldsymbol{B}_n^{-1}.\end{aligned}$$

Multiplying the equation above by $\boldsymbol{B}_n = \lambda \boldsymbol{I}_n - \epsilon (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \hat{\boldsymbol{U}}_n$ on both sides, and after manipulating terms to isolate $\boldsymbol{Q}_n$ on the left-hand side gives

$$\begin{aligned}\boldsymbol{Q}_m(\lambda) = \frac{1}{\lambda^2} \Bigg[& \boldsymbol{B}_n - (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{M}_m^{-1} \hat{\boldsymbol{U}}_n \\ & + \lambda \epsilon \Big[(\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m(\lambda) + \boldsymbol{Q}_m(\lambda) (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \hat{\boldsymbol{U}}_n \Big] \\ & - \epsilon^2 (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m(\lambda) (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \hat{\boldsymbol{U}}_n \Bigg] \\ = \frac{1}{\lambda^2} \Bigg[& \boldsymbol{B}_n - (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{M}_m^{-1} \hat{\boldsymbol{U}}_n \\ & + \lambda \epsilon (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \left[\frac{1}{m} \hat{\boldsymbol{A}}_m \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} + \lambda \boldsymbol{I}_m\right]^{-1} \hat{\boldsymbol{U}}_n \\ & + \lambda \epsilon (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \left[\frac{1}{m} \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \hat{\boldsymbol{A}}_m + \lambda \boldsymbol{I}_m\right]^{-1} \hat{\boldsymbol{U}}_n \\ & - \epsilon^2 (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \left[\frac{1}{m} \hat{\boldsymbol{A}}_m \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} + \lambda \boldsymbol{I}_m\right]^{-1} \hat{\boldsymbol{U}}_n (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \hat{\boldsymbol{U}}_n \Bigg].\end{aligned}$$

Applying the operator nom on the equality above, we find

$$\begin{aligned}\|\boldsymbol{Q}_m(\lambda)\| \leq \frac{1}{\lambda^2} \Bigg[& \lambda + 2\epsilon + 2\|\boldsymbol{M}_m^{-1}\| \\ & + 2\lambda\epsilon \left\| \left[\frac{1}{m} \hat{\boldsymbol{A}}_m \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} + \lambda \boldsymbol{I}_m\right]^{-1} \right\| + 2\lambda\epsilon \left\| \left[\frac{1}{m} \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \hat{\boldsymbol{A}}_m + \lambda \boldsymbol{I}_m\right]^{-1} \right\| \\ & + 4\epsilon^2 \left\| \left[\frac{1}{m} \hat{\boldsymbol{A}}_m \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} + \lambda \boldsymbol{I}_m\right]^{-1} \right\| \Bigg],\end{aligned} \tag{A.45}$$

since

$$\|\boldsymbol{B}_n\| = \left\|\lambda \boldsymbol{I}_n - \epsilon (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \hat{\boldsymbol{U}}_n\right\| \leq \lambda + 2\epsilon.$$

From Lemma A.4.2, we have

$$\left\| \left[\frac{1}{m} \hat{\boldsymbol{A}}_m \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} + \lambda \boldsymbol{I}_m\right]^{-1} \right\| \leq \frac{1}{\lambda} \frac{4}{\xi_{\min}^2},$$

and

$$\left\| \left[\frac{1}{m} \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \hat{\boldsymbol{A}}_m + \lambda \boldsymbol{I}_m\right]^{-1} \right\| \leq \frac{1}{\lambda} \frac{4}{\xi_{\min}^2}.$$

We find an upper bound for $\|\boldsymbol{M}_m^{-1}\|$ to finish the proof. By denoting by $\boldsymbol{Z}^T \boldsymbol{Z}$ the Cholesky decomposition of the positive-semi-definite matrix $\left[\frac{1}{m} \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} + \epsilon \boldsymbol{I}_m\right]^{-1}$, we reuse the Woodbury

identity (Lemma A.8.2) to rewrite $\boldsymbol{M}_m^{-1}$ as

$$\begin{aligned}\boldsymbol{M}_m^{-1} &= \left[\left[\frac{1}{m}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} + \epsilon\boldsymbol{I}_m\right]^{-1} + \left[\lambda\hat{\boldsymbol{A}}_m^{-1} - \epsilon\boldsymbol{I}_m\right]^{-1}\right]^{-1} \\ &= \left[\boldsymbol{Z}^T\boldsymbol{Z} + \left[\lambda\hat{\boldsymbol{A}}_m^{-1} - \epsilon\boldsymbol{I}_m\right]^{-1}\right]^{-1} \\ &= \left[\lambda\hat{\boldsymbol{A}}_m^{-1} - \epsilon\boldsymbol{I}_m\right] - \left[\lambda\hat{\boldsymbol{A}}_m^{-1} - \epsilon\boldsymbol{I}_m\right]\boldsymbol{Z}^T\left[\boldsymbol{Z}\left[\lambda\hat{\boldsymbol{A}}_m^{-1} - \epsilon\boldsymbol{I}_m\right]\boldsymbol{Z}^T + \boldsymbol{I}_m\right]^{-1}\boldsymbol{Z}\left[\lambda\hat{\boldsymbol{A}}_m^{-1} - \epsilon\boldsymbol{I}_m\right].\end{aligned}$$

From Lemma A.8.6,

$$\left\|\left[\boldsymbol{Z}\left[\lambda\hat{\boldsymbol{A}}_m^{-1} - \epsilon\boldsymbol{I}_m\right]\boldsymbol{Z}^T + \boldsymbol{I}_m\right]^{-1}\right\| \leq 1,$$

since $H\left(\boldsymbol{Z}\left[\lambda\hat{\boldsymbol{A}}_m^{-1} - \epsilon\boldsymbol{I}_m\right]\boldsymbol{Z}^T\right)$ is positive-semi-definite, and from Lemma A.4.3 we have

$$\|\hat{\boldsymbol{A}}_m^{-1}\| \leq \frac{1}{\xi_{\min}}.$$

Besides,

$$\|\boldsymbol{Z}\|^2 = \nu_{\max}\left(\left[\frac{1}{m}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} + \epsilon\boldsymbol{I}_m\right]^{-1}\right) \leq \frac{1}{\epsilon}.$$

We deduce for the operator norm of $\boldsymbol{M}_m^{-1}$ that

$$\|\boldsymbol{M}_m^{-1}\| \leq \left(\frac{\lambda}{\xi_{\min}} + \epsilon\right) + \frac{1}{\epsilon}\left(\frac{\lambda}{\xi_{\min}} + \epsilon\right)^2.$$

Setting $\epsilon = \frac{\lambda}{2\epsilon'} < \lambda \min\left\{\frac{1}{\xi_{\max}}, \frac{\xi_{\min}}{4}\right\}$ for $\epsilon' > \frac{1}{2}\min\left\{\frac{1}{\xi_{\max}}, \frac{\xi_{\min}}{4}\right\}$ and putting upper bounds of $\|\boldsymbol{M}_m^{-1}\|$, $\left\|\left[\frac{1}{m}\hat{\boldsymbol{A}}_m\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} + \lambda\boldsymbol{I}_m\right]^{-1}\right\|$, $\left\|\left[\frac{1}{m}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{A}}_m + \lambda\boldsymbol{I}_m\right]^{-1}\right\|$ into equation A.45 give

$$\begin{aligned}\|\boldsymbol{Q}_m(\lambda)\| &\leq \frac{1}{\lambda^2}\left[\lambda + \frac{\lambda}{\epsilon'} + \lambda\left(\frac{2}{\xi_{\min}} + \frac{1}{\epsilon'}\right) + \lambda\epsilon'\left(\frac{2}{\xi_{\min}} + \frac{1}{\epsilon'}\right)^2 + \lambda\frac{8}{\xi_{\min}^2\epsilon'} + \lambda\frac{4}{\xi_{\min}^2\epsilon'^2}\right] \\ &= \frac{1}{\lambda}\left[1 + \frac{1}{\epsilon'} + \left(\frac{2}{\xi_{\min}} + \frac{1}{\epsilon'}\right) + \epsilon'\left(\frac{2}{\xi_{\min}} + \frac{1}{\epsilon'}\right)^2 + \frac{8}{\xi_{\min}^2\epsilon'} + \frac{4}{\xi_{\min}^2\epsilon'^2}\right].\end{aligned}$$

□

Remark 41. *From the proof of Lemma A.4.1, eigenspectrum constraints on the empirical transition model matrix $\hat{\boldsymbol{A}}_m$ in Assumption 2 ensure the resolvent $\boldsymbol{Q}_m(\lambda)$ is uniformly bounded.*

Lemma A.4.2. *Under Assumptions 1 and 2, let $\lambda > 0$ and let $\boldsymbol{Q}'_m(\lambda), \boldsymbol{Q}''_m(\lambda) \in \mathbb{R}^{m\times m}$ be the following resolvents*

$$\boldsymbol{Q}'_m(\lambda) = \left[\frac{1}{m}\hat{\boldsymbol{A}}_m\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} + \lambda\boldsymbol{I}_m\right]^{-1}$$

and

$$\boldsymbol{Q}''_m(\lambda) = \left[\frac{1}{m}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{A}}_m + \lambda\boldsymbol{I}_m\right]^{-1},$$

where $\hat{\boldsymbol{A}}_m = \hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \in \mathbb{R}^{m\times m}$ is the empirical transition model matrix (equation 6.14).

Then, for all m, we have

$$\|\boldsymbol{Q}'_m(\lambda)\| \leq \frac{1}{\lambda}\frac{4}{\xi^2_{\min}} \quad \textit{and} \quad \|\boldsymbol{Q}''_m(\lambda)\| \leq \frac{1}{\lambda}\frac{4}{\xi^2_{\min}}.$$

Proof. Since the symmetric part of the empirical transition model matrix $\hat{\boldsymbol{A}}_m$ is positive-definite under Assumption 2, the matrix $\hat{\boldsymbol{A}}_m$ is non-singular. We write thus

$$\begin{aligned}
\|\boldsymbol{Q}'_m(\lambda)\| &= \left\| \left[\frac{1}{m}\hat{\boldsymbol{A}}_m \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} + \lambda \boldsymbol{I}_m \right]^{-1} \right\| \\
&= \left\| \left[\frac{1}{m}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} + \lambda \hat{\boldsymbol{A}}_m^{-1} \right]^{-1} \hat{\boldsymbol{A}}_m^{-1} \right\| \\
&\leq \left\| \left[\frac{1}{m}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}} + \lambda \hat{\boldsymbol{A}}_m^{-1} - \lambda \nu_{\min}\big(H(\hat{\boldsymbol{A}}_m^{-1})\big)\boldsymbol{I}_m + \lambda \nu_{\min}\big(H(\hat{\boldsymbol{A}}_m^{-1})\big)\boldsymbol{I}_m \right]^{-1} \right\| \|\hat{\boldsymbol{A}}_m^{-1}\| \\
&= \frac{1}{\lambda}\frac{1}{\nu_{\min}\big(H(\hat{\boldsymbol{A}}_m^{-1})\big)}\|\hat{\boldsymbol{A}}_m^{-1}\|.
\end{aligned}$$

The last inequality is obtained with Lemma A.8.6 since $H\Big(\frac{1}{m}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} + \lambda\hat{\boldsymbol{A}}_m^{-1} - \lambda\nu_{\min}\big(H(\hat{\boldsymbol{A}}_m^{-1})\big)\boldsymbol{I}_m\Big)$ is positive-semi-definite. By construction of both $\hat{\boldsymbol{U}}_n$ and $\hat{\boldsymbol{V}}_n$, we have $\|\hat{\boldsymbol{U}}_n\| \leq 1$ and $\|\hat{\boldsymbol{V}}_n\| \leq 1$. We deduce that

$$\big\|\hat{\boldsymbol{A}}_m\big\| = \big\|\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\big\| < 2.$$

Since $H(\hat{\boldsymbol{A}}_m^{-1}) = [\hat{\boldsymbol{A}}_m^{-1}]^T H(\hat{\boldsymbol{A}}_m)\hat{\boldsymbol{A}}_m^{-1}$, we deduce from Ostrowski's theorem (Lemma A.8.4) that

$$\nu_{\min}\big(H(\hat{\boldsymbol{A}}_m^{-1})\big) \geq \frac{\nu_{\min}\big(H(\hat{\boldsymbol{A}}_m)\big)}{\|\hat{\boldsymbol{A}}_m\|^2} \geq \frac{\xi_{\min}}{4}.$$

Furthermore, from Lemma A.4.3, we have $\|\hat{\boldsymbol{A}}_m^{-1}\| \leq \frac{1}{\xi_{\min}}$. We conclude that

$$\|\boldsymbol{Q}'_m(\lambda)\| \leq \frac{1}{\lambda}\frac{4}{\xi^2_{\min}}.$$

With similar reasoning, we can find the same upper bound for $\|\boldsymbol{Q}''_m(\lambda)\|$. □

Lemma A.4.3. *Let $\hat{\boldsymbol{A}}_m = \hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T$ be the empirical transition model matrix defined in equation 6.14. Under Assumption 2, for all m, we have*

$$\|\hat{\boldsymbol{A}}_m^{-1}\| \leq \frac{1}{\xi_{\min}}.$$

Proof. We rewrite $\hat{\boldsymbol{A}}_m$ as

$$\hat{\boldsymbol{A}}_m^{-1} = \Big[\big[\hat{\boldsymbol{A}}_m - \nu_{\min}\big(H(\hat{\boldsymbol{A}}_m)\big)\boldsymbol{I}_m\big] + \nu_{\min}\big(H(\hat{\boldsymbol{A}}_m)\big)\boldsymbol{I}_m\Big]^{-1}.$$

Since the matrix $H\Big(\big[\hat{\boldsymbol{A}}_m - \nu_{\min}\big(H(\hat{\boldsymbol{A}}_m)\big)\boldsymbol{I}_m\big]\Big)$ is positive-semi-definite, we apply Lemma A.8.6

on $\hat{\boldsymbol{A}}_m^{-1}$ to get

$$\|\hat{\boldsymbol{A}}_m^{-1}\| \leq \frac{1}{\nu_{\min}\big(H(\hat{\boldsymbol{A}}_m)\big)} \leq \frac{1}{\xi_{\min}}.$$

□

Lemma A.4.4. *Under Assumption 1 and 2, let $\lambda > 0$ and let $\boldsymbol{Q}_m(\lambda) \in \mathbb{R}^{n\times n}$ be the resolvent defined in equation 6.13 as*

$$\boldsymbol{Q}_m(\lambda) = \left[\frac{1}{m}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n + \lambda\boldsymbol{I}_n\right]^{-1}.$$

Then there exists a real $K > 0$ such that, for all m, we have

$$\left\|\frac{1}{\sqrt{m}}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m(\lambda)\right\| \leq K$$

and

$$\left\|\frac{1}{\sqrt{m}}\boldsymbol{Q}_m(\lambda)(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\right\| \leq 2K.$$

Proof. From Lemma A.4.1, we know there exists a real $K > 0$ such that, for all m, we have $\|\boldsymbol{Q}_m(\lambda)\| \leq K$. Since the symmetric part of the empirical transition model matrix $\hat{\boldsymbol{A}}_m = \hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T$ (equation 6.14) is positive-definite under Assumption 2, the matrix $\hat{\boldsymbol{A}}_m$ is non-singular. Furthermore, from Lemma A.4.3 we have $\|\hat{\boldsymbol{A}}_m^{-1}\| \leq \frac{1}{\xi_{\min}}$, and both $\|\hat{\boldsymbol{U}}_n\|$ and $\|\hat{\boldsymbol{V}}_n\|$ are upper bounded by 1. We deduce that

$$\begin{aligned}
\left\|\frac{1}{\sqrt{m}}\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m(\lambda)\right\| &= \left\|\frac{1}{m}\boldsymbol{Q}_m(\lambda)^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m(\lambda)\right\|^{\frac{1}{2}} \\
&= \left\|\frac{1}{m}\boldsymbol{Q}_m(\lambda)^T\hat{\boldsymbol{U}}_n^T\hat{\boldsymbol{A}}_m^{-1}\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m(\lambda)\right\|^{\frac{1}{2}} \\
&= \left\|\boldsymbol{Q}_m(\lambda)^T\hat{\boldsymbol{U}}_n^T\hat{\boldsymbol{A}}_m^{-1}\hat{\boldsymbol{U}}_n\big[\boldsymbol{I}_n - \lambda\boldsymbol{Q}_m(\lambda)\big]\right\|^{\frac{1}{2}} \\
&\leq \sqrt{\frac{K(1+K)}{\xi_{\min}}}.
\end{aligned}$$

Similarly, we have

$$\begin{aligned}
&\left\|\frac{1}{\sqrt{m}}\boldsymbol{Q}_m(\lambda)(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\right\| \\
&= \left\|\frac{1}{m}\boldsymbol{Q}_m(\lambda)(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\hat{\boldsymbol{A}}_m^{-1}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\boldsymbol{Q}_m(\lambda)^T\right\|^{\frac{1}{2}} \\
&= \left\|\big[\boldsymbol{I}_n - \lambda\boldsymbol{Q}_m(\lambda)\big](\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\hat{\boldsymbol{A}}_m^{-1}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\boldsymbol{Q}_m(\lambda)^T\right\|^{\frac{1}{2}} \\
&\leq 2\sqrt{\frac{K(1+K)}{\xi_{\min}}}.
\end{aligned}$$

□

Lemma A.4.5. *Let $\lambda > 0$ and $\hat{\boldsymbol{U}}_n, \hat{\boldsymbol{V}}_n$ be the auxiliary matrices defined in equation 6.7. The map-*

ping $f : \boldsymbol{W} \to \left\| \left[\frac{1}{m} (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W})^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W}) \hat{\boldsymbol{U}}_n + \lambda \boldsymbol{I}_n \right]^{-1} \right\|$ *is* $K/\sqrt{m}$*-Lipschitz continuous with respect to the Frobenius norm, for* $K > 0$ *independent of* N *and* m*.*

Proof. Let $\boldsymbol{Q}_m : \boldsymbol{W} \mapsto \left[\frac{1}{m} (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W})^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W}) \hat{\boldsymbol{U}}_n + \lambda \boldsymbol{I}_n \right]^{-1}$. From Lemma A.4.1, we know there exists a real $K_{\boldsymbol{Q}_m} > 0$ such that, for all m and $\boldsymbol{W}$, we have

$$\|\boldsymbol{Q}_m(\boldsymbol{W})\| \leq K_{\boldsymbol{Q}_m}.$$

Furthermore, both $\|\hat{\boldsymbol{U}}_n\|$ and $\|\hat{\boldsymbol{V}}_n\|$ are upper bounded by 1. Let $\boldsymbol{H} \in \mathbb{R}^{N \times d}$, we have

$$\begin{aligned}
&|f(\boldsymbol{W} + \boldsymbol{H}) - f(\boldsymbol{W})| \\
&\left| \|\boldsymbol{Q}_m(\boldsymbol{W} + \boldsymbol{H})\| - \|\boldsymbol{Q}_m(\boldsymbol{W})\| \right| \\
&\leq \left\| \boldsymbol{Q}_m(\boldsymbol{W} + \boldsymbol{H}) - \boldsymbol{Q}_m(\boldsymbol{W}) \right\| \\
&= \left\| \frac{1}{m} \boldsymbol{Q}_m(\boldsymbol{W} + \boldsymbol{H}) (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \left[\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W} + \boldsymbol{H})^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W} + \boldsymbol{H}) - \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W})^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W}) \right] \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m(\boldsymbol{W}) \right\| \\
&= \left\| \frac{1}{m} \boldsymbol{Q}_m(\boldsymbol{W} + \boldsymbol{H}) (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \left[\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W} + \boldsymbol{H})^T \left[\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W} + \boldsymbol{H}) - \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W}) \right] \right.\right. \\
&\qquad \left.\left. + \left[\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W} + \boldsymbol{H}) - \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W}) \right]^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W}) \right] \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m(\boldsymbol{W}) \right\| \\
&\leq \left\| \frac{1}{m} \boldsymbol{Q}_m(\boldsymbol{W} + \boldsymbol{H}) (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W} + \boldsymbol{H})^T \left[\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W} + \boldsymbol{H}) - \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W}) \right] \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m(\boldsymbol{W}) \right\| \\
&\qquad + \left\| \frac{1}{m} \boldsymbol{Q}_m(\boldsymbol{W} + \boldsymbol{H}) (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \left[\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W} + \boldsymbol{H}) - \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W}) \right]^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W}) \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m(\boldsymbol{W}) \right\| \\
&\leq K_{\boldsymbol{Q}_m} \left\| \frac{1}{\sqrt{m}} \boldsymbol{Q}_m(\boldsymbol{W} + \boldsymbol{H}) (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W} + \boldsymbol{H})^T \right\| \left\| \frac{1}{\sqrt{m}} \left[\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W} + \boldsymbol{H}) - \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W}) \right] \right\| \\
&\qquad + 2K_{\boldsymbol{Q}_m} \left\| \frac{1}{\sqrt{m}} \left[\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W} + \boldsymbol{H}) - \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W}) \right] \right\| \left\| \frac{1}{\sqrt{m}} \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W}) \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m(\boldsymbol{W}) \right\|.
\end{aligned}$$

From Lemma A.4.4, we know there exists a real $K' > 0$ such that, for all m, we have

$$\left\| \frac{1}{\sqrt{m}} \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W}) \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m(\boldsymbol{W}) \right\| \leq K'$$

and

$$\left\| \frac{1}{\sqrt{m}} \boldsymbol{Q}_m(\boldsymbol{W} + \boldsymbol{H}) (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W} + \boldsymbol{H})^T \right\| \leq 2K'.$$

From those results, we conclude the Lipschitz continuity of f with respect to the Frobenius norm since

$$\begin{aligned}
|f(\boldsymbol{W} + \boldsymbol{H}) - f(\boldsymbol{W})| &\leq 4K_{\boldsymbol{Q}_m} K' \left\| \frac{1}{\sqrt{m}} \left[\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W} + \boldsymbol{H}) - \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W}) \right] \right\| \\
&\leq 4K_{\boldsymbol{Q}_m} K' \left\| \frac{1}{\sqrt{m}} \left[\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W} + \boldsymbol{H}) - \boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W}) \right] \right\|_F \\
&\leq \frac{4K_{\boldsymbol{Q}_m} K' K_{\boldsymbol{\sigma}}}{\sqrt{m}} \|\boldsymbol{H}\boldsymbol{S}\|_F \\
&= \frac{4K_{\boldsymbol{Q}_m} K' K_{\boldsymbol{\sigma}}}{\sqrt{m}} \sqrt{\mathrm{Tr}\left(\boldsymbol{H}\boldsymbol{S}\boldsymbol{S}^T\boldsymbol{H}^T\right)} \\
&\leq \frac{4K_{\boldsymbol{Q}_m} K' K_{\boldsymbol{\sigma}}}{\sqrt{m}} \|\boldsymbol{S}\| \, \|\boldsymbol{H}\|_F.
\end{aligned}$$

□

Lemma A.4.6. *Let $\lambda > 0$, $\hat{\boldsymbol{U}}_n, \hat{\boldsymbol{V}}_n$ be the auxiliary matrices defined in equation 6.7, and $\boldsymbol{Q}_m : \boldsymbol{W} \mapsto \left[\frac{1}{m}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W})^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W})\hat{\boldsymbol{U}}_n + \lambda\boldsymbol{I}_n\right]^{-1}$. The mapping $f : \boldsymbol{W} \mapsto \left\|\boldsymbol{Q}_m(\boldsymbol{W})^T\boldsymbol{Q}_m(\boldsymbol{W})\right\|$ is $K/\sqrt{m}$-Lipschitz continuous with respect to the Frobenius norm, for $K > 0$ independent of N and m.*

Proof. From Lemma A.4.1 and Lemma A.4.5, we know there exists reals $K_{\boldsymbol{Q}_m}, K > 0$ such that, for all m and $\boldsymbol{W}, \boldsymbol{H}$, we have

$$\|\boldsymbol{Q}_m(\boldsymbol{W})\| \leq K_{\boldsymbol{Q}_m}.$$

and

$$\left\|\left[\boldsymbol{Q}_m(\boldsymbol{W}+\boldsymbol{H}) - \boldsymbol{Q}_m(\boldsymbol{W})\right]\right\| \leq \frac{K}{\sqrt{m}}\|\boldsymbol{H}\|_F$$

Let $\boldsymbol{H} \in \mathbb{R}^{N\times d}$, we have

$$\begin{aligned}
|f(\boldsymbol{W}+\boldsymbol{H}) - f(\boldsymbol{W})| &= \left|\left\|\boldsymbol{Q}_m(\boldsymbol{W}+\boldsymbol{H})^T\boldsymbol{Q}_m(\boldsymbol{W}+\boldsymbol{H})\right\| - \left\|\boldsymbol{Q}_m(\boldsymbol{W})^T\boldsymbol{Q}_m(\boldsymbol{W})\right\|\right| \\
&\leq \left\|\boldsymbol{Q}_m(\boldsymbol{W}+\boldsymbol{H})^T\boldsymbol{Q}_m(\boldsymbol{W}+\boldsymbol{H}) - \boldsymbol{Q}_m(\boldsymbol{W})^T\boldsymbol{Q}_m(\boldsymbol{W})\right\| \\
&\leq \left\|\boldsymbol{Q}_m(\boldsymbol{W}+\boldsymbol{H})^T\left[\boldsymbol{Q}_m(\boldsymbol{W}+\boldsymbol{H}) - \boldsymbol{Q}_m(\boldsymbol{W})\right]\right\| \\
&\quad + \left\|\left[\boldsymbol{Q}_m(\boldsymbol{W}+\boldsymbol{H}) - \boldsymbol{Q}_m(\boldsymbol{W})\right]^T\boldsymbol{Q}_m(\boldsymbol{W})\right\| \\
&\leq \frac{2KK_{\boldsymbol{Q}_m}}{\sqrt{m}}\|\boldsymbol{H}\|_F.
\end{aligned}$$

□

A.5 Existence of the Resolvent $\boldsymbol{Q}_m(\lambda)$

In this section, we show that Assumption 2 guarantees the existence of the resolvent $\boldsymbol{Q}_m(\lambda)$ (Lemma A.5.1), but also that Assumption 2 may be true in practice under certain conditions (Lemma A.5.2).

Lemma A.5.1. *Under Assumption 2, for any $\lambda > 0$, the resolvent $\boldsymbol{Q}_m(\lambda)$ defined in equation 6.13 exists.*

Proof. From Assumption 2, we know that $\nu_{\min}\big(H(\hat{\boldsymbol{A}}_m)\big) > \xi_{\min} > 0$, and thus $H(\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{A}}_m\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T)$ is at least semi-positive-definite. From the Min-Max theorem, we deduce that the eigenvalues of $\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{A}}_m\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T$ have nonnegative real parts. Consequently, the eigenvalues of $\frac{1}{m}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n$ have nonnegative real parts since both $\frac{1}{m}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n$ and $\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}\hat{\boldsymbol{A}}_m\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}^T$ share the same nonzero eigenvalues from the Weinstein–Aronszajn identity (Lemma A.8.5). □

Lemma A.5.2. *Let $c : \hat{\mathcal{S}} \to \mathbb{N}$ and $c' : \hat{\mathcal{S}} \to \mathbb{N}$ be defined such that, for all $i \in [m]$, $c(\hat{\boldsymbol{S}}_i)$ and $c'(\hat{\boldsymbol{S}}_i)$ represent the number of times $\hat{\boldsymbol{S}}_i$ occurs in $\boldsymbol{X}_n$ and $\boldsymbol{X}'_n$, respectively. If for all $i \in [m]$, $c(\hat{\boldsymbol{S}}_i) \geq \gamma c'(\hat{\boldsymbol{S}}_i)$ then the symmetric part of the empirical transition model matrix $\hat{\boldsymbol{A}}_m$ (defined in equation 6.14) is positive-definite.*

Proof. The structure of $\sqrt{n}\hat{\boldsymbol{U}}_n \in \mathbb{R}^{m\times n}$ indicates each column i of $\hat{\boldsymbol{U}}_n$ is a one-hot vector, where its j-th element is 1 if the i-th state $\boldsymbol{s}_i$ of $\boldsymbol{X}_n$ is $\hat{\boldsymbol{S}}_j$. Conversely, each row i of $\sqrt{n}\hat{\boldsymbol{U}}_n$ has a

j-th element equal to one if the j-th state $\boldsymbol{s}_j$ of $\boldsymbol{X}_n$ is $\hat{\boldsymbol{S}}_i$. A similar correspondence holds for $\sqrt{n}\hat{\boldsymbol{V}}_n$ and $\boldsymbol{X}'_n$. From interpretations of $\hat{\boldsymbol{U}}_n$ and $\hat{\boldsymbol{V}}_n$, we deduce that $\hat{\boldsymbol{C}}_n = n\hat{\boldsymbol{U}}_n\hat{\boldsymbol{U}}_n^T \in \mathbb{R}^{m\times m}$ and $\hat{\boldsymbol{C}}'_n = n\hat{\boldsymbol{V}}_n\hat{\boldsymbol{V}}_n^T \in \mathbb{R}^{m\times m}$ are diagonal matrices where the i-th element of its diagonal is equal to $c(\hat{\boldsymbol{S}}_i)$ and $c'(\hat{\boldsymbol{S}}_i)$, respectively. In the same way, $\hat{\boldsymbol{N}}_n = n\hat{\boldsymbol{U}}_n\hat{\boldsymbol{V}}_n^T \in \mathbb{R}^{m\times m}$ is matrix for which $[\hat{\boldsymbol{N}}_n]_{ij}$ is $c(\hat{\boldsymbol{S}}_i \to \hat{\boldsymbol{S}}_j)$, i.e., the number of times the state $\hat{\boldsymbol{S}}_i$ follows $\hat{\boldsymbol{S}}_j$ in $\mathcal{D}_{\text{train}}$. We want to prove $H(\hat{\boldsymbol{A}}_m) = \frac{\hat{\boldsymbol{A}}_m + \hat{\boldsymbol{A}}_m^T}{2}$ is positive-definite by using the Gershgorin circle theorem and by showing $H(\hat{\boldsymbol{A}}_m)$ is strictly diagonally dominant, i.e., $|[H(\hat{\boldsymbol{A}}_m)]_{ii}| > \sum_{i\neq j}|[H(\hat{\boldsymbol{A}}_m)]_{ij}|$.

For all $i \in [n]$, we have

$$[H(\hat{\boldsymbol{A}}_m)]_{ii} = \frac{1}{n}\left[[\hat{\boldsymbol{C}}_n]_{ii} - \gamma[\hat{\boldsymbol{N}}_n]_{ii}\right] = \frac{c(\hat{\boldsymbol{S}}_i) - \gamma c(\hat{\boldsymbol{S}}_i \to \hat{\boldsymbol{S}}_i)}{n} > 0,$$

and for all $i \neq j$

$$[H(\hat{\boldsymbol{A}}_m)]_{ij} = \frac{-\gamma[\hat{\boldsymbol{N}}_n]_{ij} - \gamma[\hat{\boldsymbol{N}}_n]_{ji}}{2n} = \frac{-\gamma c(\hat{\boldsymbol{S}}_i \to \hat{\boldsymbol{S}}_j) - \gamma c(\hat{\boldsymbol{S}}_j \to \hat{\boldsymbol{S}}_i)}{2n} < 0.$$

To prove that $H(\hat{\boldsymbol{A}}_m)$ is positive-definite it remains to show that

$$\begin{aligned}\sum_j [H(\hat{\boldsymbol{A}}_m)]_{ij} &= \sum_{j\neq i}[H(\hat{\boldsymbol{A}}_m)]_{ij} + [H(\hat{\boldsymbol{A}}_m)]_{ii} \\ &= \sum_j \left[\frac{\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T}{2}\right]_{ij} + \sum_j \left[\frac{\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T}{2}\right]_{ji} > 0\end{aligned}$$

for all $i \in [m]$. Let $i \in [m]$, we have

$$\sum_j [\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T]_{ij} = \frac{c(\hat{\boldsymbol{S}}_i)}{n} - \gamma\sum_j \frac{c(\hat{\boldsymbol{S}}_i \to \hat{\boldsymbol{S}}_j)}{n} = (1-\gamma)\frac{c(\hat{\boldsymbol{S}}_i)}{n} > 0$$

and

$$\sum_j [\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T]_{ji} = \frac{c(\hat{\boldsymbol{S}}_i)}{n} - \gamma\sum_j \frac{c(\hat{\boldsymbol{S}}_j \to \hat{\boldsymbol{S}}_i)}{n} = \frac{c(\hat{\boldsymbol{S}}_i) - \gamma c'(\hat{\boldsymbol{S}}_i)}{n} > 0,$$

since $c(\hat{\boldsymbol{S}}_i) \geq \gamma c'(\hat{\boldsymbol{S}}_i)$ for all $i \in [m]$. We deduce for all $i \in [m]$ that

$$\sum_j [H(\hat{\boldsymbol{A}}_m)]_{ij} = \sum_j \left[\frac{\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T}{2}\right]_{ij} + \sum_j \left[\frac{\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T}{2}\right]_{ji} > 0,$$

and thus $H(\hat{\boldsymbol{A}}_m)$ is strictly diagonally dominant and positive-definite. □

Remark 42. *Conditions of Lemma A.5.2 may hold in practice. If $\mathcal{D}_{train}$ is derived from a sample path of the MRP, where $\boldsymbol{s}'_{i+1} = \boldsymbol{s}_i$ for all $i \in [n-1]$, and if $\hat{\boldsymbol{S}}_l$ depicts the distinct visited state corresponding to the last next state visited $\boldsymbol{s}'_n$ in $\mathcal{D}_{train}$, then we have $c(\hat{\boldsymbol{S}}_i) = c'(\hat{\boldsymbol{S}}_i)$ for all $i \neq l$ and $c(\hat{\boldsymbol{S}}_l) = c'(\hat{\boldsymbol{S}}_l) - 1$. For sufficiently large n, we may have $c(\hat{\boldsymbol{S}}_l) \geq \frac{\gamma}{1-\gamma}$ which satisfies conditions of Lemma A.5.2. Similarly, conditions of Lemma A.5.2 are satisfied for the pathwise LSTD algorithm, where $\mathcal{D}_{train}$ is perturbed slightly by setting the feature of the next state of the last transition to zero (Lazaric et al., 2012) to get $c(\hat{\boldsymbol{S}}_l) \geq c'(\hat{\boldsymbol{S}}_l)$.*

A.6 About the Existence, Positiveness, and Uniqueness of the correction factor δ

This section is dedicated to proving that the fixed-point solution δ of equation 7.8 is unique and positive under Assumptions 1 and 2. This result is proven in the following Lemma.

Lemma A.6.1. *Under Assumptions 1 and 2, for all m, let δ be the solution to the fixed-point equation 7.8 defined as*

$$\delta = \frac{1}{m}\operatorname{Tr}\left((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\left[\frac{N}{m}\frac{1}{1+\delta}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n + \lambda\boldsymbol{I}_n\right]^{-1}\right).$$

Then δ exists, is positive, and is unique.

Proof. For ease of notations, we define the matrix $\boldsymbol{B}_n = (\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n$. The proof is based on the use of Lemma A.8.7 on the mapping $f : \delta \mapsto \frac{1}{m}\operatorname{Tr}\big(\boldsymbol{B}_n\bar{\boldsymbol{Q}}_m(\delta)\big)$. To apply Lemma A.8.7, we need to show *i.* f is positive on $[0,\infty)$, *ii.* f is monotonically increasing, *iii.* f is scalable, and *iv.* f admits $x_0 \in [0,\infty)$ such that $x_0 \geq f(x_0)$. Following this plan, we will show first *i.*, i.e., $f(\delta) > 0$ for all $\delta > 0$. By denoting $\nu_j\big(\boldsymbol{B}_n\bar{\boldsymbol{Q}}_m(\delta)\big)$ the j-th eigenvalues of the matrix $\boldsymbol{B}_n\bar{\boldsymbol{Q}}_m(\delta)$, we have

$$\begin{aligned}
\nu_j\big(\boldsymbol{B}_n\bar{\boldsymbol{Q}}_m(\delta)\big) &= \nu_j\left(\boldsymbol{B}_n\left[\frac{N}{m}\frac{1}{1+\delta}\boldsymbol{B}_n + \lambda\boldsymbol{I}_n\right]^{-1}\right)\\
&= \nu_j(\boldsymbol{B}_n)\nu_j\left(\left[\frac{N}{m}\frac{1}{1+\delta}\boldsymbol{B}_n + \lambda\boldsymbol{I}_n\right]^{-1}\right) \qquad \text{(from the Schur decomposition of } \boldsymbol{B}_n)\\
&= \frac{\nu_j(\boldsymbol{B}_n)}{\frac{N}{m}\frac{1}{1+\delta}\nu_j\left(\boldsymbol{B}_n\right) + \lambda}\\
&= \frac{1}{\left|\frac{N}{m}\frac{1}{1+\delta}\nu_j\left(\boldsymbol{B}_n\right) + \lambda\right|^2}\left(\frac{N}{m}\frac{1}{1+\delta}|\nu_j(\boldsymbol{B}_n)|^2 + \lambda\nu_j(\boldsymbol{B}_n)\right).
\end{aligned}$$

Let $\hat{\boldsymbol{A}}_m = \hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T$ be the transition model matrix defined in equation 6.14, and $\bar{\boldsymbol{Z}}\bar{\boldsymbol{Z}}^T$ be the Cholesky decompositon of $\boldsymbol{\Phi}_{\hat{\mathcal{S}}}$. From the Weinstein–Aronszajn identity (Lemma A.8.5), the matrices $\boldsymbol{B}_n = (\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n$ and $\bar{\boldsymbol{Z}}^T\hat{\boldsymbol{A}}_m\bar{\boldsymbol{Z}}$ share the same non-zero eigenvalues. Under Assumption 2, the matrix $H(\bar{\boldsymbol{Z}}^T\hat{\boldsymbol{A}}_m\bar{\boldsymbol{Z}})$ is at least semi-positive-definite, which implies that non-zero real parts of eigenvalues of $\bar{\boldsymbol{Z}}^T\hat{\boldsymbol{A}}_m\bar{\boldsymbol{Z}}$ are positive. We deduce that $\operatorname{Re}\big(\nu_j(\boldsymbol{B}_n)\big) \geq 0$, for all $j \in [m]$. As a consequence,

$$\begin{aligned}
f(\delta) &= \frac{1}{m}\operatorname{Tr}\big(\boldsymbol{B}_n\bar{\boldsymbol{Q}}_m(\delta)\big)\\
&= \frac{1}{m}\sum_{j=1}^{n}\frac{1}{\left|\frac{N}{m}\frac{1}{1+\delta}\nu_j\left(\boldsymbol{B}_n\right) + \lambda\right|^2}\left(\frac{N}{m}\frac{1}{1+\delta}|\nu_j\left(\boldsymbol{B}_n\right)|^2 + \lambda\nu_j(\boldsymbol{B}_n)\right)\\
&= \frac{1}{m}\sum_{j=1}^{n}\frac{1}{\left|\frac{N}{m}\frac{1}{1+\delta}\nu_j\left(\boldsymbol{B}_n\right) + \lambda\right|^2}\left(\frac{N}{m}\frac{1}{1+\delta}|\nu_j\left(\boldsymbol{B}_n\right)|^2 + \lambda\operatorname{Re}\big(\nu_j(\boldsymbol{B}_n)\big)\right)\\
&> 0.
\end{aligned} \tag{A.46}$$

To prove *ii.*, i.e., f is monotonically increasing on $[0,\infty)$, we show the derivative f' of f is positive

on $[0, \infty)$. Let $\delta > 0$,

$$
\begin{aligned}
f'(\delta) &= \frac{1}{m}\left(\sum_{j=1}^{n} \frac{\nu_j(\boldsymbol{B}_n)}{\frac{N}{m}\frac{1}{1+\delta}\nu_j\left(\boldsymbol{B}_n\right)+\lambda}\right)' \\
&= \frac{1}{m}\sum_{j=1}^{n} \frac{\frac{N}{m}\frac{1}{(1+\delta)^2}}{\left(\frac{N}{m}\frac{1}{1+\delta}\nu_j\left(\boldsymbol{B}_n\right)+\lambda\right)^2}\nu_j(\boldsymbol{B}_n)^2 \\
&= \frac{1}{m}\sum_{j=1}^{n} \frac{\frac{N}{m}\frac{1}{(1+\delta)^2}}{\left|\frac{N}{m}\frac{1}{1+\delta}\nu_j\left(\boldsymbol{B}_n\right)+\lambda\right|^4}\left(\frac{N}{m}\frac{1}{(1+\delta)}|\nu_j(\boldsymbol{B}_n)|^2+\lambda\nu_j(\boldsymbol{B}_n)\right)^2 \\
&= \frac{1}{m}\sum_{j=1}^{n} \frac{\frac{N}{m}\frac{1}{(1+\delta)^2}}{\left|\frac{N}{m}\frac{1}{1+\delta}\nu_j\left(\boldsymbol{B}_n\right)+\lambda\right|^4}\left(\frac{N^2}{m^2}\frac{1}{(1+\delta)^2}|\nu_j(\boldsymbol{B}_n)|^4+2\lambda\frac{N}{m}\frac{1}{1+\delta}|\nu_j(\boldsymbol{B}_n)|^2\nu_j(\boldsymbol{B}_n)+\lambda^2\nu_j(\boldsymbol{B}_n)^2\right) \\
&= \underbrace{\frac{1}{m}\sum_{j=1}^{n} \frac{\frac{N}{m}\frac{1}{(1+\delta)^2}}{\left|\frac{N}{m}\frac{1}{1+\delta}\nu_j\left(\boldsymbol{B}_n\right)+\lambda\right|^4}\left(\frac{N^2}{m^2}\frac{1}{(1+\delta)^2}|\nu_j(\boldsymbol{B}_n)|^4+2\lambda\frac{N}{m}\frac{1}{1+\delta}|\nu_j(\boldsymbol{B}_n)|^2\,\mathrm{Re}\big(\nu_j(\boldsymbol{B}_n)\big)\right)}_{(1)} \\
&\quad \underbrace{+\sum_{j=1}^{n}\lambda^2\frac{\frac{N}{m}\frac{1}{(1+\delta)^2}}{\left|\frac{N}{m}\frac{1}{1+\delta}\nu_j\left(\boldsymbol{B}_n\right)+\lambda\right|^4}\mathrm{Re}\big(\nu_j(\boldsymbol{B}_n^2)\big)}_{(2)}
\end{aligned}
$$

Since real parts of eigenvalues of $\boldsymbol{B}_n$ are positive, (1) is clearly positive. Since $\mathrm{Tr}(\boldsymbol{B}_n^2) > 0$ (Lemma A.6.2) and thus (2) is positive, we can conclude *ii.*. We can use a similar proof for the scalability in *iii.*, i.e., $\alpha f(\delta) > f(\alpha\delta),\ \forall\alpha > 1$. Let $\alpha > 1$ and $\delta > 0$,

$$
\begin{aligned}
\alpha f(\delta) - f(\alpha\delta) &= \alpha\frac{1}{m}\,\mathrm{Tr}\big(\boldsymbol{B}_n\bar{\boldsymbol{Q}}_m(\delta)\big) - \frac{1}{m}\,\mathrm{Tr}\big(\boldsymbol{B}_n\bar{\boldsymbol{Q}}_m(\alpha\delta)\big) && \text{(A.47)}\\
&= \frac{1}{m}\,\mathrm{Tr}\big(\boldsymbol{B}_n\big[\alpha\bar{\boldsymbol{Q}}_m(\delta) - \bar{\boldsymbol{Q}}_m(\alpha\delta)\big]\big) && \text{(A.48)}\\
&= \frac{1}{m}\,\mathrm{Tr}\left(\alpha\boldsymbol{B}_n\bar{\boldsymbol{Q}}_m(\delta)\left[\frac{N}{m}\left(\frac{1}{1+\alpha\delta}-\frac{1}{\alpha(1+\delta)}\right)\boldsymbol{B}_n + \left(\lambda-\frac{\lambda}{\alpha}\right)\boldsymbol{I}_n\right]\bar{\boldsymbol{Q}}_m(\alpha\delta)\right) && \text{(A.49)}\\
&= \underbrace{\alpha\frac{1}{m}\frac{N}{m}\left(\frac{1}{1+\alpha\delta}-\frac{1}{\alpha(1+\delta)}\right)}_{>0}\underbrace{\mathrm{Tr}\big(\boldsymbol{B}_n\bar{\boldsymbol{Q}}_m(\delta)\boldsymbol{B}_n\bar{\boldsymbol{Q}}_m(\alpha\delta)\big)}_{(1)} \\
&\quad + \underbrace{\alpha\frac{1}{m}\left(\lambda-\frac{\lambda}{\alpha}\right)}_{>0}\underbrace{\mathrm{Tr}\big(\boldsymbol{B}_n\bar{\boldsymbol{Q}}_m(\delta)\bar{\boldsymbol{Q}}_m(\alpha\delta)\big)}_{(2)}. && \text{(A.50)}
\end{aligned}
$$

To prove *iii.*, we can show that both (1) and (2) in equation A.50 are positive. We prove in *ii.* that $\mathrm{Tr}\big(\boldsymbol{B}_n\bar{\boldsymbol{Q}}_m(\delta')\boldsymbol{B}_n\bar{\boldsymbol{Q}}_m(\delta)\big) > 0$ for any $\delta' > \delta$. Since $\alpha\delta > \delta$, we also deduce (1) is positive. For (2), we can write

$$
\begin{aligned}
&\mathrm{Tr}\big(\boldsymbol{B}_n\bar{\boldsymbol{Q}}_m(\delta)\bar{\boldsymbol{Q}}_m(\alpha\delta)\big) \\
&= \sum_{j=1}^{n}\nu_j\big(\boldsymbol{B}_n\bar{\boldsymbol{Q}}_m(\delta)\bar{\boldsymbol{Q}}_m(\alpha\delta)\big)
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{j=1}^{n} \frac{\nu_j(\boldsymbol{B}_n)}{\left(\frac{N}{m}\frac{1}{1+\delta}\nu_j(\boldsymbol{B}_n)+\lambda\right)\left(\frac{N}{m}\frac{1}{1+\alpha\delta}\nu_j(\boldsymbol{B}_n)+\lambda\right)} \\
&= \sum_{j=1}^{n} c_j \left(\left(\frac{N^2}{m^2}\frac{|\nu_j(\boldsymbol{B}_n)|^2}{(1+\delta)(1+\alpha\delta)} + \lambda^2 \right) \mathrm{Re}\big(\nu_j(\boldsymbol{B}_n)\big) + \frac{N}{m}\left(\frac{\lambda}{1+\delta}+\frac{\lambda}{1+\alpha\delta}\right)|\nu_j(\boldsymbol{B}_n)|^2 \right) \\
&> 0,
\end{aligned}
$$

where

$$
c_j = \frac{1}{\left|\left(\frac{N}{m}\frac{1}{1+\delta}\nu_j(\boldsymbol{B}_n)+\lambda\right)\left(\frac{N}{m}\frac{1}{1+\alpha\delta}\nu_j(\boldsymbol{B}_n)+\lambda\right)\right|^2}.
$$

In order to apply Lemma A.8.7, we still need to demonstrate *iv.*, i.e., f admits $x_0 \in [0,\infty)$ such that $x_0 \geq f(x_0)$. To prove *iv.*, it is sufficient to notice that if f is bounded, i.e., $\forall \delta, f(\delta) \leq C$. Let $\delta > 0$, we have

$$
\begin{aligned}
f(\delta) = \frac{1}{m}\mathrm{Tr}\big(\boldsymbol{B}_n\bar{\boldsymbol{Q}}_m(\delta)\big) &= \frac{1}{m}\mathrm{Tr}\big((\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\bar{\boldsymbol{Q}}_m(\delta)\big) \\
&= \frac{1}{m}\mathrm{Tr}\big(\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\bar{\boldsymbol{Q}}_m(\delta)(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\big) \\
&\leq \frac{1}{m}\mathrm{Tr}(\boldsymbol{\Phi}_{\hat{\mathcal{S}}})\|\hat{\boldsymbol{U}}_n\bar{\boldsymbol{Q}}_m(\delta)(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\| \\
&\leq \frac{2}{n}\mathrm{Tr}(\boldsymbol{\Phi}_{\hat{\mathcal{S}}})\|\bar{\boldsymbol{Q}}_m(\delta)\| \\
&= \mathcal{O}(1),
\end{aligned}
$$

where we used for the first inequality $|\mathrm{Tr}(\boldsymbol{AB})| \leq \|\boldsymbol{B}\|\,\mathrm{Tr}(\boldsymbol{A})$ for non-negative definite matrix $\boldsymbol{A}$. The last inequality is obtained since $\frac{1}{m}\mathrm{Tr}(\boldsymbol{\Phi}_{\hat{\mathcal{S}}})$ is uniformly bounded under Assumptions 1 and 2 (see equation A.4). Furthermore, both $\|\hat{\boldsymbol{U}}_n\|$ and $\|\hat{\boldsymbol{V}}_n\|$ are upper bounded by 1 and, with a similar proof than for Lemma A.4.1, we can show there exists a real $K_{\bar{\boldsymbol{Q}}} > 0$ such that, for all m and for all $\delta \in [0,\infty)$, we have $\|\bar{\boldsymbol{Q}}_m(\delta)\| \leq K_{\bar{\boldsymbol{Q}}}$. Since all hypotheses required on f to apply Lemma A.8.7 are satisfied, we can apply this Lemma, which concludes the proof. □

Lemma A.6.2. *We have*

$$
\mathrm{Tr}\big((\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\big) > 0.
$$

Proof. Let $\boldsymbol{A} = \hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T$. We denote by $S(\boldsymbol{A}) = \frac{\boldsymbol{A}-\boldsymbol{A}^T}{2}$ the skew-symmetric part of $\boldsymbol{A}$. We have

$$
\begin{aligned}
&\mathrm{Tr}\big((\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n(\hat{\boldsymbol{U}}_n-\gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\big) \\
&= \mathrm{Tr}\big(\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\boldsymbol{A}\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\boldsymbol{A}\big) \\
&= \mathrm{Tr}\big(\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\boldsymbol{A}\boldsymbol{\Phi}_{\hat{\mathcal{S}}}H(\boldsymbol{A})\big) + \mathrm{Tr}\big(\boldsymbol{\Phi}_{\hat{\mathcal{S}}}\boldsymbol{A}\boldsymbol{\Phi}_{\hat{\mathcal{S}}}S(\boldsymbol{A})\big) \\
&= \mathrm{Tr}\big(\boldsymbol{\Phi}_{\hat{\mathcal{S}}}H(\boldsymbol{A})\boldsymbol{\Phi}_{\hat{\mathcal{S}}}H(\boldsymbol{A})\big) + \mathrm{Tr}\big(\boldsymbol{\Phi}_{\hat{\mathcal{S}}}S(\boldsymbol{A})\boldsymbol{\Phi}_{\hat{\mathcal{S}}}H(\boldsymbol{A})\big) + \mathrm{Tr}\big(\boldsymbol{\Phi}_{\hat{\mathcal{S}}}H(\boldsymbol{A})\boldsymbol{\Phi}_{\hat{\mathcal{S}}}S(\boldsymbol{A})\big) + \mathrm{Tr}\big(\boldsymbol{\Phi}_{\hat{\mathcal{S}}}S(\boldsymbol{A})\boldsymbol{\Phi}_{\hat{\mathcal{S}}}S(\boldsymbol{A})\big) \\
&= \mathrm{Tr}\big(\boldsymbol{\Phi}_{\hat{\mathcal{S}}}H(\boldsymbol{A})\boldsymbol{\Phi}_{\hat{\mathcal{S}}}H(\boldsymbol{A})\big) + \mathrm{Tr}\big(\boldsymbol{\Phi}_{\hat{\mathcal{S}}}S(\boldsymbol{A})\boldsymbol{\Phi}_{\hat{\mathcal{S}}}S(\boldsymbol{A})\big) > 0.
\end{aligned}
$$

□

Lemma A.6.3. *Under Assumptions 1 and 2, let δ be the correction factor defined in equation 7.8.*

δ is a decreasing function with respect to N.

Proof. For ease of notations, we define the matrix $\boldsymbol{B}_n = (\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \boldsymbol{\Phi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n$ and we denote by $\bar{\boldsymbol{Q}}_m$ the resolvent $\bar{\boldsymbol{Q}}_m(\lambda)$. The derivative of δ as function of N is denoted as $\delta'(N)$ and defined as

$$\delta'(N) = -\frac{1}{m} \frac{\frac{1}{m}\operatorname{Tr}(\boldsymbol{B}_n \bar{\boldsymbol{Q}}_m \boldsymbol{B}_n \bar{\boldsymbol{Q}}_m)}{(1+\delta)} }{1 - \frac{N}{m}\frac{1}{m}\frac{\operatorname{Tr}(\boldsymbol{B}_n \bar{\boldsymbol{Q}}_m \boldsymbol{B}_n \bar{\boldsymbol{Q}}_m)}{(1+\delta)^2}}$$

For all N, we have $\delta'(N) \leq 0$ since $\frac{1}{m}\frac{\operatorname{Tr}(\boldsymbol{B}_n \bar{\boldsymbol{Q}}_m \boldsymbol{B}_n \bar{\boldsymbol{Q}}_m)}{(1+\delta)} > 0$ and $\frac{N}{m}\frac{1}{m}\frac{\operatorname{Tr}(\boldsymbol{B}_n \bar{\boldsymbol{Q}}_m \boldsymbol{B}_n \bar{\boldsymbol{Q}}_m)}{(1+\delta)^2} < 1$ using a similar reasoning than for equation A.22. □

Lemma A.6.4. *Under Assumptions 1 and 2, let δ be the correction factor defined in equation 7.8. δ is a decreasing function with respect to λ.*

Proof. For ease of notations, we define the matrix $\boldsymbol{B}_n = (\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T \boldsymbol{\Phi}_{\hat{\mathcal{S}}} \hat{\boldsymbol{U}}_n$ and we denote by $\bar{\boldsymbol{Q}}_m$ the resolvent $\bar{\boldsymbol{Q}}_m(\lambda)$. The derivative of δ as function of λ is denoted as $\delta'(\lambda)$ and defined as

$$\delta'(\lambda) = -\frac{1}{m}\operatorname{Tr}(\bar{\boldsymbol{Q}}_m \boldsymbol{B}_n \bar{\boldsymbol{Q}}_m)$$

For all λ, we have $\delta'(\lambda) \leq 0$ using a similar reasoning than for *iii.* in Lemma A.6.1. □

A.7 Concentration Results

The following section is dedicated to a set of concentration results used for the proofs of Theorems. Preliminary results yield a concentration of measure properties for the random feature matrix $\boldsymbol{\Sigma}_{\hat{\mathcal{S}}} \in \mathbb{R}^{N\times m}$, which stem from the concentration inequality of Lemma 7.2.1 for Lipschitz applications of a Gaussian vector. Essentially, the guideline of the proofs involves the following steps: given $\boldsymbol{W}_{ij} = \varphi(\tilde{\boldsymbol{W}}_{ij})$, for which $\tilde{\boldsymbol{W}}_{ij} \sim \mathcal{N}(0,1)$ and φ a Lipschitz function, the normal concentration of $\tilde{\boldsymbol{W}}$ is transferred to $\boldsymbol{W}$. This process induces a normal concentration of the random vector $\sigma(\boldsymbol{w}^T\hat{\boldsymbol{S}})$, for $\boldsymbol{w} = \varphi(\tilde{\boldsymbol{w}})$ and $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$, and of the matrix $\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}$. This implies that Lipschitz functionals of $\sigma(\boldsymbol{w}^T\hat{\boldsymbol{S}})$ or $\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}$ also concentrate. As highlighted earlier, these concentration results have multiple consequences on convergence of random variables and are traditionally employed in Random Matrix theory and in Theorem 7.2.3. We start by revisiting Lemma A.7.1 and Lemma A.7.2, which are derived from Lemma 7.2.1 and that were previously introduced in Louart et al. (2018). Subsequently, we provide intermediary Lemma 7.2.2 and Lemma A.7.3 to reach the principal result of this section articulated by Lemma A.7.4, which is employed in proofs of Theorems. In the remainder of this section, we denote by $\|\cdot\|_F$ the Frobenius norm of a matrix.

Lemma A.7.1. *Let $\boldsymbol{\sigma} : \mathbb{R} \to \mathbb{R}$ be a $K_{\boldsymbol{\sigma}}$-Lipschitz continuous function, let $\boldsymbol{X} \in \mathbb{R}^{d\times m}$ be a matrix, and let $\boldsymbol{w} = \varphi(\tilde{\boldsymbol{w}})$ be a vector for which $\varphi : \mathbb{R} \to \mathbb{R}$ is a K_φ-Lipschitz continuous function and $\tilde{\boldsymbol{w}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$. Let*

$$t_0 = |\sigma(0)| + K_{\boldsymbol{\sigma}} K_\varphi \|\boldsymbol{X}\| \sqrt{\frac{d}{m}}.$$

Then, for all $t \geq 4t_0$, we have

$$\Pr\left(\left\| \frac{1}{\sqrt{m}} \sigma(\boldsymbol{w}^T\boldsymbol{X}) \right\| \geq t \right) \leq C e^{-\frac{cmt^2}{2K_{\boldsymbol{\sigma}}^2 K_\varphi^2 \|\boldsymbol{X}\|^2}},$$

for some $C, c > 0$ are independent of all other parameters.

Proof. The proof of this Lemma can be found in the first half of proof of Louart et al. (2018, Lemma 2), and is based on Lemma 7.2.1. □

Corollary A.7.1.1. *(Louart et al., 2018, Remark 2) Let $\boldsymbol{X} \in \mathbb{R}^{d\times m}$ and let $\boldsymbol{\Sigma_X} = \sigma(\boldsymbol{WX}) \in \mathbb{R}^{N\times m}$ be its random features matrix defined as in equation 6.5. For all $t \geq 4t_0$, we have*

$$\Pr\left(\left\| \frac{1}{m}\boldsymbol{\Sigma_X} \right\| \geq t \right) \leq CNe^{-\frac{cm^2t^2}{2N\|\boldsymbol{X}\|^2}},$$

where $t_0 = |\sigma(0)| + \|\boldsymbol{X}\|\sqrt{\frac{d}{m}}$.

From the previous Lemma, we deduce the following key concentration result.

Lemma A.7.2. *(Louart et al., 2018, Lemma 2) Let $\boldsymbol{\sigma} : \mathbb{R} \to \mathbb{R}$ be a $K_{\boldsymbol{\sigma}}$-Lipschitz continuous function, let $\boldsymbol{X} \in \mathbb{R}^{d\times m}$ be a matrix, and let $\boldsymbol{w} = \varphi(\tilde{\boldsymbol{w}})$ be a vector for which $\varphi : \mathbb{R} \to \mathbb{R}$ is a K_φ-Lipschitz continuous function and $\tilde{\boldsymbol{w}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$. Let $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ be a matrix independent of $\boldsymbol{w}$ such that $\|\boldsymbol{A}\| \leq K_{\boldsymbol{A}}$. Then, we have*

$$\begin{aligned}&\Pr\left(\left| \frac{1}{m}\sigma(\boldsymbol{w}^T\boldsymbol{X})^T\boldsymbol{A}\sigma(\boldsymbol{w}^T\boldsymbol{X}) - \frac{1}{m}\operatorname{Tr}\left(\boldsymbol{A}\mathbb{E}\left[\sigma(\boldsymbol{w}^T\boldsymbol{X})\sigma(\boldsymbol{w}^T\boldsymbol{X})^T\right]\right) \right| > t \right)\\ &\leq Ce^{-\frac{cm}{2K_{\boldsymbol{\sigma}}^2K_\varphi^2\|\boldsymbol{X}\|^2}\min\left(\frac{t^2}{2^6t_0^2K_{\boldsymbol{A}}^2}, \frac{t}{K_{\boldsymbol{A}}}\right)},\end{aligned}$$

for $t_0 = |\sigma(0)| + \sqrt{\frac{d}{m}}K_{\boldsymbol{\sigma}}K_\varphi\|\boldsymbol{X}\|$, and $c, C \in \mathbb{R}$ independent of all other parameters.

Lemma A.7.3. *Under Assumptions 1 and 2, let $\lambda > 0$, let $\boldsymbol{W} \in \mathbb{R}^{N\times d}$, and let the resolvent*

$$\boldsymbol{Q}_m(\boldsymbol{W}) = \left[\frac{1}{m}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W})^T\boldsymbol{\Sigma}_{\hat{\mathcal{S}}}(\boldsymbol{W})\hat{\boldsymbol{U}}_n + \lambda\boldsymbol{I}_n\right]^{-1}$$

defined as in equation 6.13. Let $\boldsymbol{\sigma} \in \mathbb{R}^m$ independent of $\boldsymbol{W}$ such that $\frac{1}{\sqrt{m}}\|\boldsymbol{\sigma}\| \leq \sqrt{K_v}$ for $K_v > 0$. Then

$$\Pr\left(\left| \frac{1}{m}\boldsymbol{\sigma}^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m(\boldsymbol{W})(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma} - \frac{1}{m}\boldsymbol{\sigma}^T\hat{\boldsymbol{U}}_n\mathbb{E}[\boldsymbol{Q}_m(\boldsymbol{W})](\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma} \right| > t \right) \leq Ce^{-cmt^2},$$

for some $C, c > 0$ independent of m and N.

Proof. Let the function $f : \boldsymbol{W} \mapsto \frac{1}{m}\boldsymbol{\sigma}^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_m(\boldsymbol{W})(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma}$. We want to show f is Lipschitz continuous to apply Lemma 7.2.2. Both $\|\hat{\boldsymbol{U}}_n\|$ and $\|\hat{\boldsymbol{V}}_n\|$ are upper bounded by 1. Let $\boldsymbol{H} \in \mathbb{R}^{N\times d}$, we have

$$\begin{aligned}&|f(\boldsymbol{W} + \boldsymbol{H}) - f(\boldsymbol{W})|\\ &= \left| \frac{1}{m}\boldsymbol{\sigma}^T\hat{\boldsymbol{U}}_n\Big[\boldsymbol{Q}_m(\boldsymbol{W} + \boldsymbol{H}) - \boldsymbol{Q}_m(\boldsymbol{W})\Big](\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\sigma} \right|\\ &\leq 2K_v\big\|\big[\boldsymbol{Q}_m(\boldsymbol{W} + \boldsymbol{H}) - \boldsymbol{Q}_m(\boldsymbol{W})\big]\big\|\end{aligned}$$

From Lemma A.4.5, we know there exists a real $K > 0$ independent of N and m such that

$$\left\| \left[\boldsymbol{Q}_m(\boldsymbol{W} + \boldsymbol{H}) - \boldsymbol{Q}_m(\boldsymbol{W}) \right] \right\| \leq \frac{K}{\sqrt{m}} \|\boldsymbol{H}\|_F$$

We prove that f is Lipschitz with parameter $\frac{2K_v K}{\sqrt{m}}$, and applying Lemma 7.2.1 gives

$$\begin{aligned} &\Pr\left(\left| \frac{1}{m} \boldsymbol{\sigma}^T \hat{\boldsymbol{U}}_n \boldsymbol{Q}_m(\boldsymbol{W}) (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\sigma} - \frac{1}{m} \boldsymbol{\sigma}^T \hat{\boldsymbol{U}}_n \mathbb{E}[\boldsymbol{Q}_m(\boldsymbol{W})] (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\sigma} \right| > t \right) \\ &\leq C e^{-\frac{cmt^2}{4K_v^2 K^2}}, \end{aligned}$$

for some $C, c > 0$ independent of other parameters. □

Lemma A.7.4. *Under Assumptions 1 and 2, let $\boldsymbol{Q}_- \in \mathbb{R}^{n \times n}$ be the resolvent defined in equation A.3, let $\boldsymbol{w}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$ be a Gaussian vector independent of $\boldsymbol{Q}_-$, and let $\boldsymbol{\sigma} : \mathbb{R} \to \mathbb{R}$ be a real 1-Lipschitz function. Then*

$$\begin{aligned} &\Pr\Bigg(\Bigg| \frac{1}{m} \sigma(\boldsymbol{w}_i^T \hat{\boldsymbol{S}}) \hat{\boldsymbol{U}}_n \boldsymbol{Q}_{-i} (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \sigma(\hat{\boldsymbol{S}}^T \boldsymbol{w}_i) \\ &\qquad - \frac{1}{m} \operatorname{Tr}\left(\hat{\boldsymbol{U}}_n \mathbb{E}[\boldsymbol{Q}_{-i}] (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \mathbb{E}[\sigma(\hat{\boldsymbol{S}}^T \boldsymbol{w}_i) \sigma(\boldsymbol{w}_i^T \hat{\boldsymbol{S}})] \right) \Bigg| > t \Bigg) \\ &\leq C e^{-cm \max(t^2, t)}, \end{aligned}$$

for some $C, c > 0$ independent of N, m.

Proof. We can observe that

$$\begin{aligned} &\Pr\Bigg(\Bigg| \frac{1}{m} \sigma(\boldsymbol{w}_i^T \hat{\boldsymbol{S}}) \hat{\boldsymbol{U}}_n \boldsymbol{Q}_{-i} (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \sigma(\hat{\boldsymbol{S}}^T \boldsymbol{w}_i) \\ &\qquad - \frac{1}{m} \operatorname{Tr}\left(\hat{\boldsymbol{U}}_n \mathbb{E}[\boldsymbol{Q}_{-i}] (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \mathbb{E}[\sigma(\hat{\boldsymbol{S}}^T \boldsymbol{w}_i) \sigma(\boldsymbol{w}_i^T \hat{\boldsymbol{S}})] \right) \Bigg| > t \Bigg) \\ &\leq \Pr\Bigg(\Bigg| \frac{1}{m} \sigma(\boldsymbol{w}^T \hat{\boldsymbol{S}})^T \hat{\boldsymbol{U}}_n \boldsymbol{Q}_{-i} (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \sigma(\boldsymbol{w}^T \hat{\boldsymbol{S}}) \\ &\qquad - \frac{1}{m} \sigma(\boldsymbol{w}^T \hat{\boldsymbol{S}})^T \hat{\boldsymbol{U}}_n \mathbb{E}[\boldsymbol{Q}_{-i}] (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \sigma(\boldsymbol{w}^T \hat{\boldsymbol{S}}) \Bigg| > \frac{t}{2} \Bigg) \\ &+ \Pr\Bigg(\Bigg| \frac{1}{m} \sigma(\boldsymbol{w}^T \hat{\boldsymbol{S}})^T \hat{\boldsymbol{U}}_n \mathbb{E}[\boldsymbol{Q}_{-i}] (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \sigma(\boldsymbol{w}^T \hat{\boldsymbol{S}}) \\ &\qquad - \frac{1}{m} \operatorname{Tr}\left(\hat{\boldsymbol{U}}_n \mathbb{E}[\boldsymbol{Q}_{-i}] (\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \mathbb{E}[\sigma(\hat{\boldsymbol{S}}^T \boldsymbol{w}_i) \sigma(\boldsymbol{w}_i^T \hat{\boldsymbol{S}})] \right) \Bigg| > \frac{t}{2} \Bigg). \end{aligned} \tag{A.51}$$

From Lemma A.4.1, there exists a real $K > 0$ such that, for all m, we have

$$\|\boldsymbol{Q}_{-i}\| \leq K.$$

Besides, both $\|\hat{\boldsymbol{U}}_n\|$ and $\|\hat{\boldsymbol{V}}_n\|$ are upper bounded by 1. We thus bound the probability of the

right-hand part with Lemma A.7.2 as

$$\begin{aligned}
&\Pr\Bigg(\bigg| \frac{1}{m}\sigma(\boldsymbol{w}^T\hat{\boldsymbol{S}})^T\hat{\boldsymbol{U}}_n\mathbb{E}[\boldsymbol{Q}_{-i}](\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\sigma(\boldsymbol{w}^T\hat{\boldsymbol{S}}) \\
&\qquad - \frac{1}{m}\operatorname{Tr}\left(\hat{\boldsymbol{U}}_n\mathbb{E}[\boldsymbol{Q}_{-i}](\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\mathbb{E}[\sigma(\hat{\boldsymbol{S}}^T\boldsymbol{w}_i)\sigma(\boldsymbol{w}_i^T\hat{\boldsymbol{S}})]\right)\bigg| > t\Bigg) \\
&\leq Ce^{-\frac{cm}{2K_{\boldsymbol{\sigma}}^2K_\varphi^2\|\hat{\boldsymbol{S}}\|^2}\min\left(\frac{t^2}{2^8t_0^2K^2},\frac{t}{2K}\right)},
\end{aligned} \tag{A.52}$$

for $t_0 = |\sigma(0)| + \sqrt{\frac{d}{m}}K_{\boldsymbol{\sigma}}K_\varphi\|\hat{\boldsymbol{S}}\|$, and $c, C \in \mathbb{R}$ independent of all other parameters. Let define the real $K' > 0$ and let $\mathcal{A}_{K'}$ be the probability space defined as

$$\mathcal{A}_{K'} = \{\boldsymbol{w} \in \mathbb{R}^m,\ \|\sigma(\boldsymbol{w}^T\hat{\boldsymbol{S}})\| \leq K'\sqrt{m}\}.$$

From Lemma A.7.1, we bound the second term $\Pr(\mathcal{A}_{K'}^c)$ as

$$\Pr(\mathcal{A}_{K'}^c) = \Pr(\{\|\sigma(\boldsymbol{w}^T\hat{\boldsymbol{S}})\| > K'\sqrt{m}\}) \leq C'e^{-\frac{c'mK'^2}{2K_{\boldsymbol{\sigma}}^2K_\varphi^2\|\boldsymbol{X}\|^2}},$$

for some $c', C' > 0$ independent of other parameters. Conditioning the random variable of interest with respect to $\mathcal{A}_{K'}$ and its complementary $\mathcal{A}_{K'}^c$ gives with Lemma A.7.3

$$\begin{aligned}
&\Pr\left(\left|\frac{1}{m}\sigma(\boldsymbol{w}^T\hat{\boldsymbol{S}})^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_-(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\sigma(\boldsymbol{w}^T\hat{\boldsymbol{S}}) - \frac{1}{m}\sigma(\boldsymbol{w}^T\hat{\boldsymbol{S}})^T\hat{\boldsymbol{U}}_n\mathbb{E}[\boldsymbol{Q}_-](\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\sigma(\boldsymbol{w}^T\hat{\boldsymbol{S}})\right| > t\right) \\
&\leq \Pr\Big(\Big|\frac{1}{m}\sigma(\boldsymbol{w}^T\hat{\boldsymbol{S}})^T\hat{\boldsymbol{U}}_n\boldsymbol{Q}_-(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\sigma(\boldsymbol{w}^T\hat{\boldsymbol{S}}) \\
&\qquad - \frac{1}{m}\sigma(\boldsymbol{w}^T\hat{\boldsymbol{S}})^T\hat{\boldsymbol{U}}_n\mathbb{E}[\boldsymbol{Q}_-](\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\sigma(\boldsymbol{w}^T\hat{\boldsymbol{S}})\Big| > t \cap \mathcal{A}_{K'}\Big) + \Pr(\mathcal{A}_{K'}^c) \\
&\leq C''e^{-c''mt^2} + C'e^{-\frac{c'mK'^2}{2K_{\boldsymbol{\sigma}}^2K_\varphi^2\|\hat{\boldsymbol{S}}\|^2}},
\end{aligned} \tag{A.53}$$

where $c'', C'' > 0$. Combing both equation A.52 and equation A.53 with equation A.51 gives

$$\begin{aligned}
&\Pr\Bigg(\bigg|\frac{1}{m}\sigma(\boldsymbol{w}_i^T\hat{\boldsymbol{S}})\hat{\boldsymbol{U}}_n\boldsymbol{Q}_{-i}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\sigma(\hat{\boldsymbol{S}}^T\boldsymbol{w}_i) \\
&\qquad - \frac{1}{m}\operatorname{Tr}\left(\hat{\boldsymbol{U}}_n\mathbb{E}[\boldsymbol{Q}_{-i}](\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\mathbb{E}[\sigma(\hat{\boldsymbol{S}}^T\boldsymbol{w}_i)\sigma(\boldsymbol{w}_i^T\hat{\boldsymbol{S}})]\right)\bigg| > t\Bigg) \\
&\leq Ce^{-\frac{cm}{2K_{\boldsymbol{\sigma}}^2K_\varphi^2\|\hat{\boldsymbol{S}}\|^2}\min\left(\frac{t^2}{2^{10}t_0^2K^2},\frac{t}{4K}\right)} + C''e^{-\frac{c''mt^2}{4}} + C'e^{-\frac{c'mK'^2}{2K_{\boldsymbol{\sigma}}^2K_\varphi^2\|\boldsymbol{X}\|^2}}.
\end{aligned} \tag{A.54}$$

□

A.7.1 Reformulation of the Second-Order Correction Factors

In this section, we provide details of the reformulation of second-order correction factors:

$$\hat{\Delta} = \frac{\lambda^2}{n}\frac{\frac{1}{N}\operatorname{Tr}\left(\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m(\lambda)^T\right)}{1-\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m(\lambda)^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m(\lambda)\right)}\|\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2 \quad \text{(equation 7.11)},$$

$$\Delta = \frac{1}{n}\frac{\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Lambda}_{\boldsymbol{P}}\left[\boldsymbol{\Theta}_{\mathcal{S}}\boldsymbol{\Psi}_2\boldsymbol{\Theta}_{\mathcal{S}}^T - 2\boldsymbol{\Theta}_{\mathcal{S}}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\Psi}_{\mathcal{S}} + \boldsymbol{\Psi}_{\mathcal{S}}\right]\right)}{1-\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m(\lambda)^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m(\lambda)\right)}\|\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2 \quad \text{(equation 7.14), and}$$

$$\Delta' = \frac{1}{n}\frac{\frac{1}{N}\operatorname{Tr}\left(\boldsymbol{D}_{\boldsymbol{\mu}^\pi}\left[\boldsymbol{\Theta}_{\mathcal{S}}\boldsymbol{\Psi}_2\boldsymbol{\Theta}_{\mathcal{S}}^T - 2\boldsymbol{\Theta}_{\mathcal{S}}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\Psi}_{\mathcal{S}} + \boldsymbol{\Psi}_{\mathcal{S}}\right]\right)}{1 - \frac{1}{N}\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m(\lambda)^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m(\lambda)\right)}\|\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2 \quad \text{(equation 7.18)}$$

in the Mercer feature space defined in Section 8.2.2. $\hat{\Delta}, \Delta$ and Δ' depend on $\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m(\lambda)^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m(\lambda)\right)$ and $\|\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2$, which can be reformulated as

$$\begin{aligned}
&\operatorname{Tr}\left(\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m(\lambda)^T\boldsymbol{\Psi}_1\bar{\boldsymbol{Q}}_m(\lambda)\right)\\
&= \tfrac{N^2}{m^2}\tfrac{1}{(1+\delta)^2}\operatorname{Tr}\left((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Omega}_{\hat{\mathcal{S}}}^T\boldsymbol{\Omega}_{\hat{\mathcal{S}}}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\bar{\boldsymbol{Q}}_m(\lambda)^T\hat{\boldsymbol{U}}_n^T\boldsymbol{\Omega}_{\hat{\mathcal{S}}}^T\boldsymbol{\Omega}_{\hat{\mathcal{S}}}\hat{\boldsymbol{U}}_n\bar{\boldsymbol{Q}}_m(\lambda)\right)\\
&= \operatorname{Tr}\left(\boldsymbol{\Pi}(\tilde{\lambda})^T\boldsymbol{\Pi}(\tilde{\lambda})\right) \quad \text{(Lemma A.8.8)}\\
&= \left\|\boldsymbol{\Pi}(\tilde{\lambda})\right\|_F^2
\end{aligned}$$

and

$$\|\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{r}\|_{\boldsymbol{\Psi}_1}^2 = \tfrac{m^2(1+\delta)^2}{N^2}\left\|\boldsymbol{\Pi}(\tilde{\lambda})\boldsymbol{\theta}_n^*\right\|^2 = \tfrac{\tilde{\lambda}^2}{\lambda^2}\left\|\boldsymbol{\Pi}(\tilde{\lambda})\boldsymbol{\theta}_n^*\right\|^2 = \tfrac{\tilde{\lambda}^2}{\lambda^2}\left\|\bar{\boldsymbol{\theta}}_n(\tilde{\lambda})\right\|^2.$$

In the Mercer feature space, we can also rewrite in Δ'

$$\begin{aligned}
&\operatorname{Tr}\left(\boldsymbol{D}_{\mu^\pi}\boldsymbol{\Theta}_{\mathcal{S}}\boldsymbol{\Psi}_2\boldsymbol{\Theta}_{\mathcal{S}}^T\right)\\
&= \tfrac{N}{m}\tfrac{1}{1+\delta}\operatorname{Tr}\left(\boldsymbol{\Pi}(\tilde{\lambda})^T\boldsymbol{\Omega}_{\mathcal{S}}\boldsymbol{D}_{\mu^\pi}\boldsymbol{\Omega}_{\mathcal{S}}^T\boldsymbol{\Pi}(\tilde{\lambda})\right)\\
&= \tfrac{\lambda}{\tilde{\lambda}}\left\|\boldsymbol{\Omega}_{\mathcal{S}}^T\boldsymbol{\Pi}(\tilde{\lambda})\right\|_{F,\boldsymbol{D}_{\mu^\pi}}^2
\end{aligned}$$

$$\tfrac{N}{m}\tfrac{1}{1+\delta}\operatorname{Tr}\left(\boldsymbol{D}_{\mu^\pi}\boldsymbol{\Phi}_{\mathcal{S}}\right) = \tfrac{\lambda}{\tilde{\lambda}}\left\|\boldsymbol{\Omega}_{\mathcal{S}}^T\right\|_{F,\boldsymbol{D}_{\mu^\pi}}^2,$$

and

$$\begin{aligned}
\tfrac{N}{m}\tfrac{1}{1+\delta}\operatorname{Tr}\left(\boldsymbol{D}_{\mu^\pi}\boldsymbol{\Theta}_{\mathcal{S}}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\Phi}_{\mathcal{S}}\right) &= \tfrac{N}{m}\tfrac{1}{1+\delta}\operatorname{Tr}\left(\boldsymbol{\Omega}_{\mathcal{S}}\boldsymbol{D}_{\mu^\pi}\boldsymbol{\Omega}_{\mathcal{S}}^T\boldsymbol{\Pi}(\tilde{\lambda})\right)\\
&= \tfrac{\lambda}{\tilde{\lambda}}\left\langle\boldsymbol{\Omega}_{\mathcal{S}}^T, \boldsymbol{\Omega}_{\mathcal{S}}^T\boldsymbol{\Pi}(\tilde{\lambda})\right\rangle_{F,\boldsymbol{D}_{\mu_\pi}}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\operatorname{Tr}\left(\boldsymbol{D}_{\mu^\pi}\left[\boldsymbol{\Theta}_{\mathcal{S}}\boldsymbol{\Psi}_2\boldsymbol{\Theta}_{\mathcal{S}}^T - 2\tfrac{N}{m}\tfrac{1}{1+\delta}\boldsymbol{\Theta}_{\mathcal{S}}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\Psi}_{\mathcal{S}} + \tfrac{N}{m}\tfrac{1}{1+\delta}\boldsymbol{\Psi}_{\mathcal{S}}\right]\right)\\
&= \tfrac{\lambda}{\tilde{\lambda}}\left\|\boldsymbol{\Omega}_{\mathcal{S}}^T - \boldsymbol{\Omega}_{\mathcal{S}}^T\boldsymbol{\Pi}(\tilde{\lambda})\right\|_{F,\boldsymbol{D}_{\mu^\pi}}^2\\
&= \tfrac{\lambda}{\tilde{\lambda}}\left\|\boldsymbol{D}_{\mu^\pi}^{\frac{1}{2}}\boldsymbol{\Omega}_{\mathcal{S}}^T\left[\boldsymbol{I}_M - \boldsymbol{\Pi}(\tilde{\lambda})\right]\right\|_F^2.
\end{aligned}$$

With a similar reformulation, we have in Δ

$$\operatorname{Tr}\left(\boldsymbol{\Lambda}_{\boldsymbol{P}}\left[\boldsymbol{\Theta}_{\mathcal{S}}\boldsymbol{\Psi}_2\boldsymbol{\Theta}_{\mathcal{S}}^T - 2\boldsymbol{\Theta}_{\mathcal{S}}(\boldsymbol{U}_n - \gamma\boldsymbol{V}_n)^T\boldsymbol{\Psi}_{\mathcal{S}} + \boldsymbol{\Psi}_{\mathcal{S}}\right]\right) = \tfrac{\lambda}{\tilde{\lambda}}\left\|\boldsymbol{D}_{\mu^\pi}^{\frac{1}{2}}\left[\boldsymbol{I}_{|\mathcal{S}|} - \gamma\boldsymbol{P}^\pi\right]\boldsymbol{\Omega}_{\mathcal{S}}^T\left[\boldsymbol{I}_M - \boldsymbol{\Pi}(\tilde{\lambda})\right]\right\|_F^2.$$

For $\hat{\Delta}$, we have

$$\begin{aligned}
&\lambda^2\operatorname{Tr}\left(\bar{\boldsymbol{Q}}_m(\lambda)\boldsymbol{\Psi}_2\bar{\boldsymbol{Q}}_m(\lambda)^T\right)\\
&= \lambda^2\tfrac{N}{m}\tfrac{1}{1+\delta}\operatorname{Tr}\left(\bar{\boldsymbol{Q}}_m(\lambda)(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Phi}_{\hat{\mathcal{S}}}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\bar{\boldsymbol{Q}}_m(\lambda)^T\right)\\
&= \lambda^2\tfrac{1}{\tilde{\lambda}^2}\tfrac{m(1+\delta)}{N}\operatorname{Tr}\left((\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)^T\boldsymbol{\Omega}_{\hat{\mathcal{S}}}^T\left[\boldsymbol{I}_M - \boldsymbol{\Pi}(\tilde{\lambda})\right]\left[\boldsymbol{I}_M - \boldsymbol{\Pi}(\tilde{\lambda})\right]^T\boldsymbol{\Omega}_{\hat{\mathcal{S}}}(\hat{\boldsymbol{U}}_n - \gamma\hat{\boldsymbol{V}}_n)\right)
\end{aligned}$$

$$= \tfrac{\lambda}{\tilde{\lambda}} \left\| (\boldsymbol{U}_n - \gamma \boldsymbol{V}_n)^T \boldsymbol{\Omega}_{\mathcal{S}}^T \left[\boldsymbol{I}_M - \boldsymbol{\Pi}(\tilde{\lambda}) \right] \right\|_F^2 .$$

We deduce that

$$\begin{aligned}
\hat{\Delta} &= \bar{\Delta}\big((\hat{\boldsymbol{U}}_n - \gamma \hat{\boldsymbol{V}}_n)^T \boldsymbol{\Omega}_{\hat{\mathcal{S}}}^T\big), \\
\Delta &= \bar{\Delta}\big(\boldsymbol{D}_{\mu^\pi}^{\frac{1}{2}} \left[\boldsymbol{I}_{|\mathcal{S}|} - \gamma \boldsymbol{P}^\pi \right] \boldsymbol{\Omega}_{\mathcal{S}}^T\big), \quad \text{and} \\
\Delta' &= \bar{\Delta}\big(\boldsymbol{D}_{\mu^\pi}^{\frac{1}{2}} \boldsymbol{\Omega}_{\mathcal{S}}^T\big);
\end{aligned}$$

where $\bar{\Delta}(\boldsymbol{M})$ is defined, for any Mercer feature matrix $\boldsymbol{M} \in \mathbb{R}^{p \times M}$ of dimension $p > 0$, as

$$\bar{\Delta}(\boldsymbol{M}) = \tfrac{1}{n} \tfrac{\tilde{\lambda}}{\lambda} \frac{\frac{1}{N} \left\| \boldsymbol{\Pi}(\tilde{\lambda}) \boldsymbol{\theta}_n^* \right\|^2}{1 - \frac{1}{N} \left\| \boldsymbol{\Pi}(\tilde{\lambda}) \right\|_F^2} \left\| \boldsymbol{M} \left[\boldsymbol{I}_M - \boldsymbol{\Pi}(\tilde{\lambda}) \right] \right\|_F^2 .$$

A.8 Intermediary Lemmas

Lemma A.8.1 (Resolvent Identity)**.** *For invertible matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n \times n}$,*

$$\boldsymbol{A}^{-1} - \boldsymbol{B}^{-1} = \boldsymbol{A}^{-1}(\boldsymbol{B} - \boldsymbol{A})\boldsymbol{B}^{-1}$$

Lemma A.8.2 (Sherman–Morrison–Woodbury Matrix Identity)**.** *(Horn and Johnson, 2012, Theorem 0.7.4) Let $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ be a non-singular matrix with a known inverse $\boldsymbol{A}^{-1}$; let $\boldsymbol{M} = \boldsymbol{A} + \boldsymbol{UCV}$, in which $\boldsymbol{U} \in \mathbb{R}^{k \times n}$, $\boldsymbol{V} \in \mathbb{R}^{n \times k}$, and $\boldsymbol{C}^{k \times k}$ is non-singular. If $\boldsymbol{M}$ and $\boldsymbol{C}^{-1} + \boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U}$ are non-singular then*

$$(\boldsymbol{A} + \boldsymbol{UCV})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}\left(\boldsymbol{C}^{-1} + \boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U}\right)^{-1}\boldsymbol{V}\boldsymbol{A}^{-1}, \tag{A.55}$$

In particular $(\boldsymbol{A} + \boldsymbol{UV})^{-1}\boldsymbol{U} = \boldsymbol{A}^{-1}\boldsymbol{U}\left(\boldsymbol{I}_n + \boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U}\right)^{-1}$ and $\boldsymbol{V}(\boldsymbol{A} + \boldsymbol{UV})^{-1} = \left(\boldsymbol{I}_n + \boldsymbol{V}\boldsymbol{A}^{-1}\boldsymbol{U}\right)^{-1}\boldsymbol{V}\boldsymbol{A}^{-1}$.

Lemma A.8.3 (Sherman–Morrison Formula)**.** *Let $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ be a non-singular matrix with a known inverse $\boldsymbol{A}^{-1}$; let $\boldsymbol{M} = \boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^T$, in which $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$. If $\boldsymbol{M}$ is non-singular and $1 + \boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{u} \neq 0$ then*

$$\left(\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^T\right)^{-1} = \boldsymbol{A}^{-1} - \frac{\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^T\boldsymbol{A}^{-1}}{1 + \boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{u}}. \tag{A.56}$$

In particular, $\left(\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^T\right)^{-1}\boldsymbol{u} = \frac{\boldsymbol{A}^{-1}\boldsymbol{u}}{1 + \boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{u}}$ and $\boldsymbol{v}^T\left(\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^T\right)^{-1} = \frac{\boldsymbol{v}^T\boldsymbol{A}^{-1}}{1 + \boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{u}}$. This Lemma is an extension of Lemma A.8.2.

Lemma A.8.4 (Ostrowski's Theorem)**.** *(Horn and Johnson, 2012, Theorem 4.5.9) Let $\boldsymbol{A}, \boldsymbol{S} \in \mathbb{R}^{n \times n}$ with $\boldsymbol{A}$ Hermitian and $\boldsymbol{S}$ nonsingular. Let the eigenvalues of $\boldsymbol{A}$, $\boldsymbol{SAS}^T$, and $\boldsymbol{SS}^T$ be arranged in nondecreasing order. Let $\sigma_1 \geq \ldots \geq \sigma_n > 0$ be the singular values of $\boldsymbol{S}$. For each $k \in [n]$ there is a positive real number $\theta_k \in [\sigma_n^2, \sigma_1^2]$ such that*

$$\nu_k(\boldsymbol{SAS}^T) = \theta_k \nu_k(\boldsymbol{A})$$

Lemma A.8.5 (Weinstein–Aronszajn Identity)**.** *For $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{B} \in \mathbb{R}^{n \times m}$ and $\lambda \in \mathbb{R} \setminus \{0\}$,*

$$\det(\boldsymbol{AB} - \lambda \boldsymbol{I}_m) = (-\lambda)^{m-n} \det(\boldsymbol{BA} - \lambda \boldsymbol{I}_n).$$

It follows that the non-zero eigenvalues of $\boldsymbol{AB}$ and $\boldsymbol{BA}$ are the same.

Lemma A.8.6. *Let $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and $\lambda > 0$.*

$$\|(\boldsymbol{A} + \lambda \boldsymbol{I}_n)^{-1}\| \leq \frac{1}{\lambda}$$

if and only if $\boldsymbol{A}\boldsymbol{A}^T + \lambda(\boldsymbol{A} + \boldsymbol{A}^T)$ is positive definite. In particular, for matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ whose the Hermitian part $H(\boldsymbol{A}) = \frac{\boldsymbol{A}+\boldsymbol{A}^T}{2}$ is semi-positive-definite we have

$$\|(\boldsymbol{A} + \lambda \boldsymbol{I}_n)^{-1}\| \leq \frac{1}{\lambda}$$

Proof.

$$\begin{aligned}
\|(\boldsymbol{A} + \lambda \boldsymbol{I}_n)^{-1}\|^2 &= \nu_{max}\left((\boldsymbol{A} + \lambda \boldsymbol{I}_n)^{-1\,T}(\boldsymbol{A} + \lambda \boldsymbol{I}_n)^{-1}\right) \\
&= \nu_{max}\left(\left[(\boldsymbol{A} + \lambda \boldsymbol{I}_n)\left(\boldsymbol{A}^T + \lambda \boldsymbol{I}_n\right)\right]^{-1}\right) \\
&= \nu_{max}\left(\left(\boldsymbol{A}\boldsymbol{A}^T + \lambda(\boldsymbol{A} + \boldsymbol{A}^T) + \lambda^2 \boldsymbol{I}_n\right)^{-1}\right) \\
&= \nu_{min}\left(\left(\boldsymbol{A}\boldsymbol{A}^T + \lambda(\boldsymbol{A} + \boldsymbol{A}^T) + \lambda^2 \boldsymbol{I}_n\right)\right)^{-1}
\end{aligned} \tag{A.57}$$

where $\nu_{max}(\boldsymbol{B})$ and $\nu_{min}(\boldsymbol{B})$ denotes the maximum eigenvalue and minimum eigenvalues of a matrix $\boldsymbol{B}$. Since $\boldsymbol{A}$ is positive-definite the matrix $\boldsymbol{A}\boldsymbol{A}^T + \lambda(\boldsymbol{A} + \boldsymbol{A}^T)$ is semi-positive-definite and has positive nonzeros eigenvalues. Therefore, $\nu_{min}\left(\left(\boldsymbol{A}\boldsymbol{A}^T + \lambda(\boldsymbol{A} + \boldsymbol{A}^T) + \lambda^2 \boldsymbol{I}_n\right)\right) > \lambda^2$ and $\|(\boldsymbol{A} + \lambda \boldsymbol{I}_n)^{-1}\| \leq \frac{1}{\lambda}$

□

Lemma A.8.7. *(Yates, 1995, Theorem 2) If a mapping $f : [0, \infty) \to [0, \infty)$*

- *is monotonically increasing, i.e $x \geq x' \implies f(x) \geq f(x')$,*
- *is scalable, i.e $\forall \alpha > 1,\ \alpha f(x) > f(\alpha x)$,*
- *admits $x_0 \in [0, \infty)$ such that $x_0 \geq f(x_0)$,*

then f has a unique fixed-point.

Lemma A.8.8. *Let $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n\times m}$. If $\boldsymbol{A}\boldsymbol{B} + \lambda \boldsymbol{I}_m$ is invertible, then*

$$\left[\boldsymbol{A}\boldsymbol{B} + \lambda \boldsymbol{I}_m\right]^{-1}\boldsymbol{A} = \boldsymbol{A}\left[\boldsymbol{B}\boldsymbol{A} + \lambda \boldsymbol{I}_n\right]^{-1}.$$

Proof. We have

$$\boldsymbol{A}\left[\boldsymbol{B}\boldsymbol{A} + \lambda \boldsymbol{I}_n\right] = \left[\boldsymbol{A}\boldsymbol{B} + \lambda \boldsymbol{I}_m\right]\boldsymbol{A}$$

Since both $\boldsymbol{A}\boldsymbol{B}$ and $\boldsymbol{B}\boldsymbol{A}$ share the same non-zero eigenvalues from Lemma A.8.5, we deduce $\boldsymbol{B}\boldsymbol{A}+\lambda \boldsymbol{I}_n$ is also invertible. By multiplying the equation above with both the inverse of $[\boldsymbol{B}\boldsymbol{A}+\lambda \boldsymbol{I}_n]$ and $\left[\boldsymbol{A}\boldsymbol{B} + \lambda \boldsymbol{I}_m\right]$, we get

$$\left[\boldsymbol{A}\boldsymbol{B} + \lambda \boldsymbol{I}_m\right]^{-1}\boldsymbol{A} = \boldsymbol{A}\left[\boldsymbol{B}\boldsymbol{A} + \lambda \boldsymbol{I}_n\right]^{-1}$$

□

Appendix B

Additional Experiments: Features Encoding in Deep Reinforcement Learning

B.1 Sparsity Curves for DQN on Discrete Control Tasks

This appendix shows the evolution of the normalized activation overlap defined in Section 12.2.2 during the learning with respect to the environment steps for experiments of Section 12.2.3.

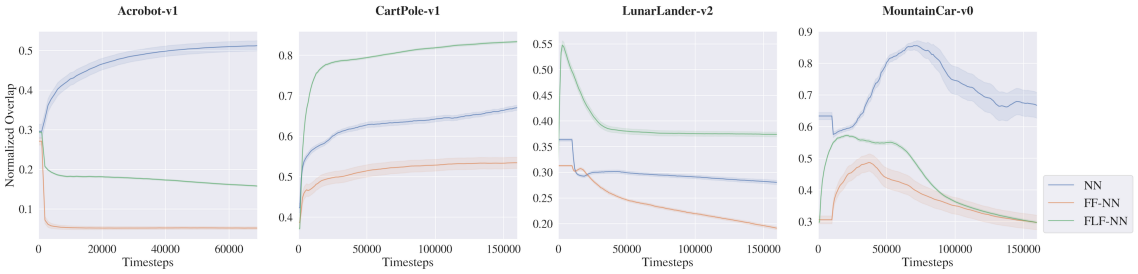


Figure B.1: Normalized Overlap over environment steps during the training for neural networks fed with raw inputs (blue), Fourier features (orange), and Fourier Light Features (green). Results are averaged over 30 trainings with shading indicating the 95% CI.

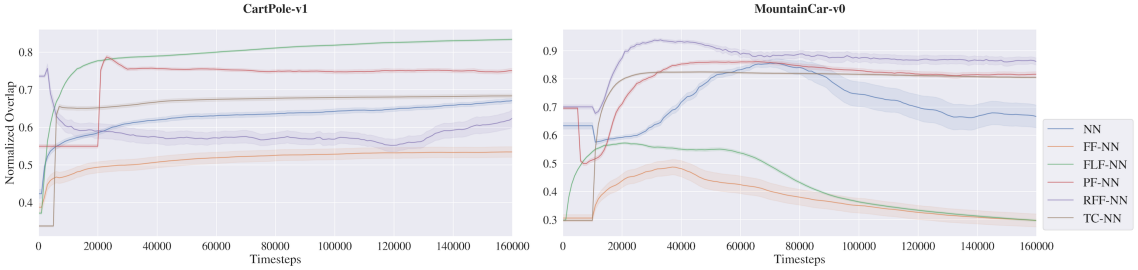
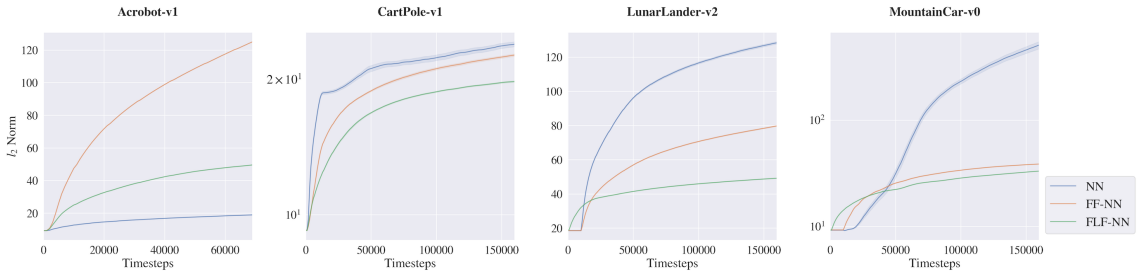


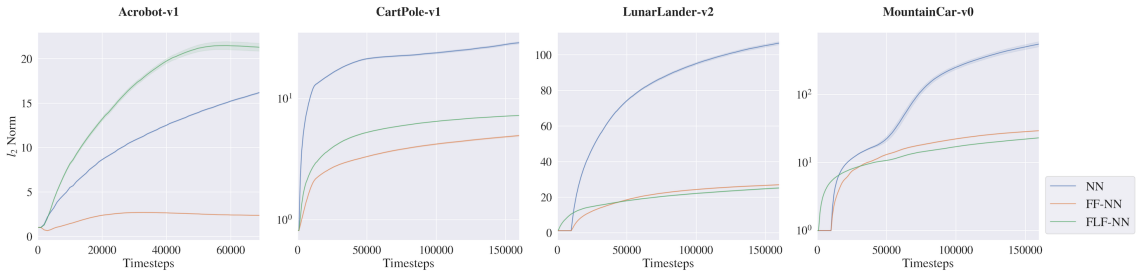
Figure B.2: Normalized Overlap over environment steps during the training for neural networks fed with raw inputs (blue), Fourier features (orange), Fourier Light Features (green), Fourier Light Features (green), Polynomial Features (red), Random Fourier Features (purple) and Tile Coding features (brown). Results are averaged over 30 trainings with shading indicating the 95% CI.

B.2 Smoothness Curves for DQN on Discrete Control Tasks

This appendix shows the evolution of the l_2, l_1, l_∞ weight norms of layers of a two-layers neural networks during the learning with respect to the environment steps for experiments of Section 12.4.

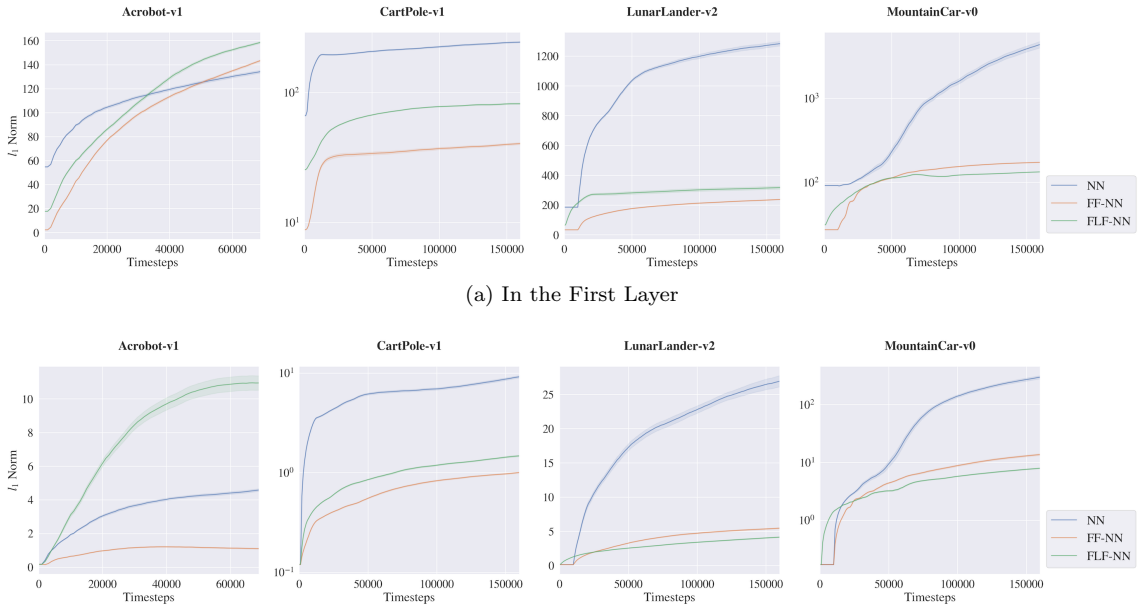


(a) In the First Layer



(b) In the Second Layer

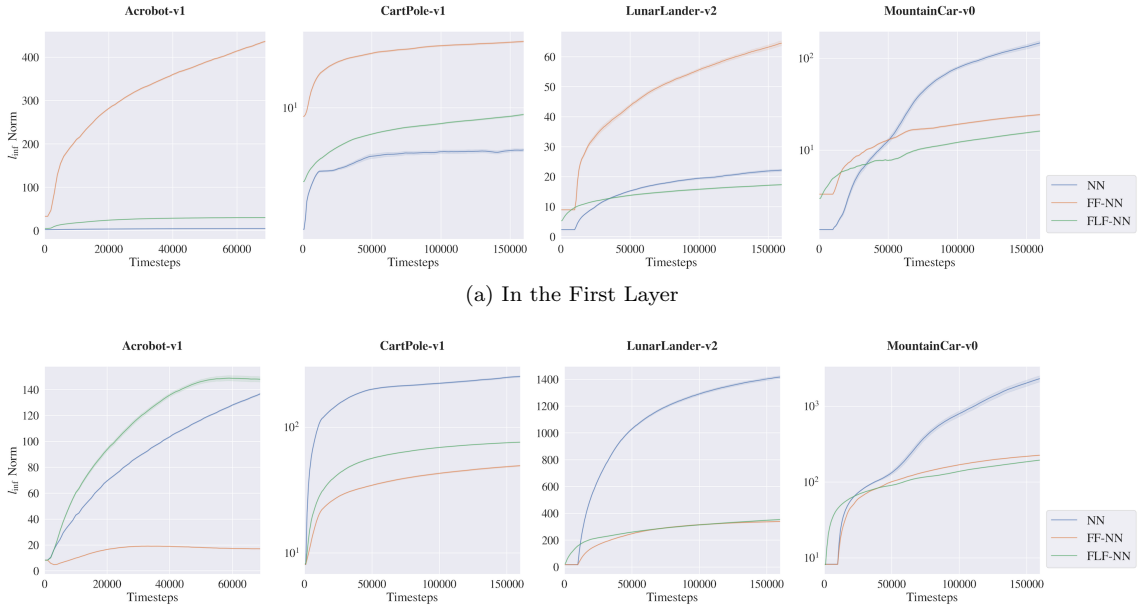
Figure B.3: L_2 weight norm of layers of a two-layers neural networks over environment steps during the training for neural networks fed with raw inputs (blue), Fourier features (orange), and Fourier Light Features (green). Results are averaged over 30 trainings with shading indicating the 95% CI.



(a) In the First Layer

(b) In the Second Layer

Figure B.4: L_1 weight norm of layers of a two-layers neural networks over environment steps during the training for neural networks fed with raw inputs (blue), Fourier features (orange), and Fourier Light Features (green). Results are averaged over 30 trainings with shading indicating the 95% CI.



(a) In the First Layer

(b) In the Second Layer

Figure B.5: L_∞ weight norm of layers of a two-layers neural networks over environment steps during the training for neural networks fed with raw inputs (blue), Fourier features (orange), and Fourier Light Features (green). Results are averaged over 30 trainings with shading indicating the 95% CI.

B.3 Interference Curves for DQN on Discrete Control Tasks

This appendix shows the evolution of interference measures defined in Section 12.1.1 during the learning with respect to the environment steps for experiments of Section 12.1.2.

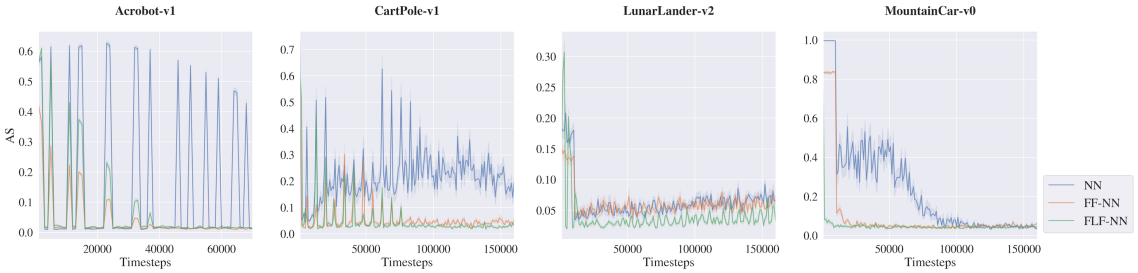


Figure B.6: Average Stiffness (AS) over environment steps during the training for neural networks fed with raw inputs (blue), Fourier features (orange), and Fourier Light Features (green). Results are averaged over 30 trainings with shading indicating the 95% CI.

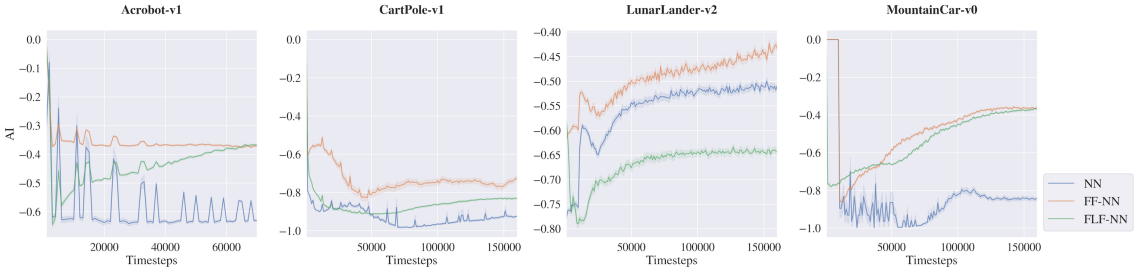


Figure B.7: Average Interference (AI) over environment steps during the training for neural networks fed with raw inputs (blue), Fourier features (orange), and Fourier Light Features (green). Results are averaged over 30 trainings with shading indicating the 95% CI.

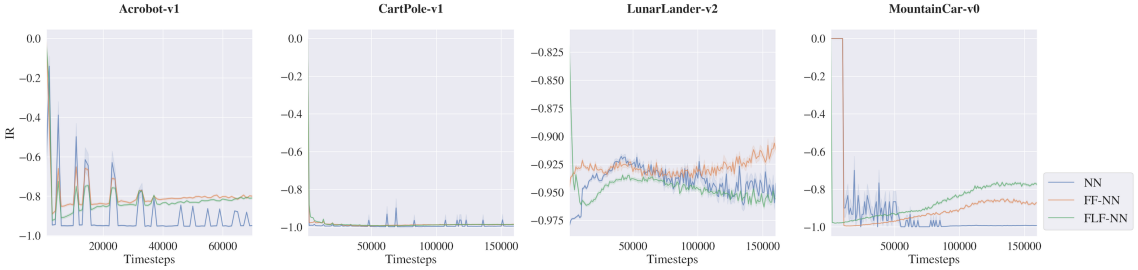


Figure B.8: Interference Risk (IR) over environment steps during the training for neural networks fed with raw inputs (blue), Fourier features (orange), and Fourier Light Features (green). Results are averaged over 30 trainings with shading indicating the 95% CI.

Experiments conducted on traditional feature encodings considered in Section 11.2.4 are depicted below.

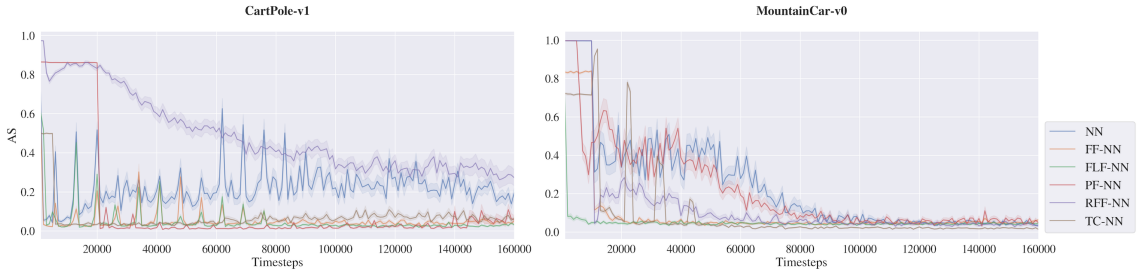


Figure B.9: Average Stiffness (AS) over environment steps during the training for neural networks fed with raw inputs (blue), Fourier features (orange), Fourier Light Features (green), Fourier Light Features (green), Polynomial Features (red), Random Fourier Features (purple) and Tile Coding features (brown). Results are averaged over 30 trainings with shading indicating the 95% CI.

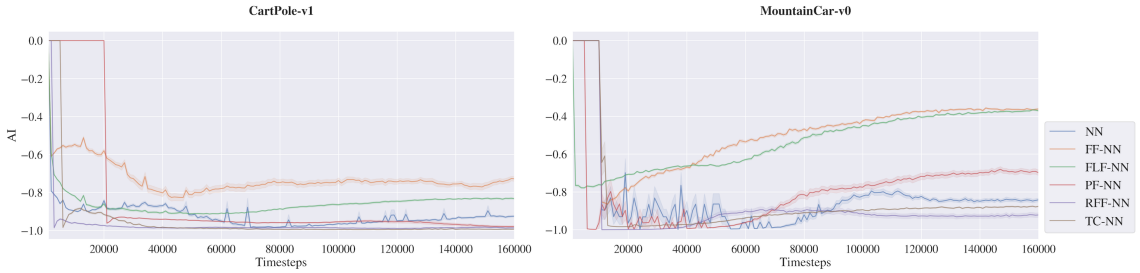


Figure B.10: Average Interference (AI) over environment steps during the training for neural networks fed with raw inputs (blue), Fourier features (orange), Fourier Light Features (green), Fourier Light Features (green), Polynomial Features (red), Random Fourier Features (purple) and Tile Coding features (brown). Results are averaged over 30 trainings with shading indicating the 95% CI.

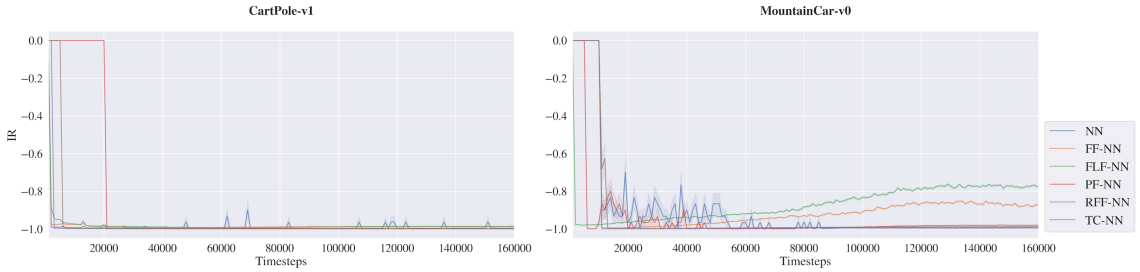


Figure B.11: Interference Risk (IR) over environment steps during the training for neural networks fed with raw inputs (blue), Fourier features (orange), Fourier Light Features (green), Fourier Light Features (green), Polynomial Features (red), Random Fourier Features (purple) and Tile Coding features (brown). Results are averaged over 30 trainings with shading indicating the 95% CI.

Bibliography

M. M. Afsar, T. Crump, and B. Far. Reinforcement learning based recommender systems: A survey. ACM Computing Surveys, 55(7):1–38, 2022.

A. Agarwal, N. Jiang, S. M. Kakade, and W. Sun. Reinforcement learning: Theory and algorithms. CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep, 32, 2019.

A. Agazzi and J. Lu. Temporal-difference learning with nonlinear function approximation: lazy training and mean field regimes. In Mathematical and Scientific Machine Learning, pages 37–74, 2022.

Z. Ahmed. emdp. https://github.com/zafarali/emdp, 2018.

T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019.

J. S. Albus. A theory of cerebellar function. Mathematical biosciences, 1971.

A. Antos, C. Szepesvári, and R. Munos. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. Machine Learning, 71: 89–129, 2008.

S. Arora, S. Du, W. Hu, Z. Li, and R. Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In International Conference on Machine Learning, pages 322–332, 2019.

F. Bach. High-dimensional analysis of double descent for linear regression with random projections. SIAM Journal on Mathematics of Data Science, 6(1):26–50, 2024.

L. Baird. Residual algorithms: Reinforcement learning with function approximation. In Machine Learning Proceedings 1995, pages 30–37. Elsevier, 1995.

E. Barnard. Temporal-difference methods and markov models. IEEE Transactions on Systems, Man, and Cybernetics, 23(2):357–365, 1993.

P. L. Bartlett, D. J. Foster, and M. J. Telgarsky. Spectrally-normalized margin bounds for neural networks. Advances in Neural Information Processing Systems, 2017.

M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine learning and the bias-variance trade-of. arXiv preprint arXiv:1812.11118, 321, 2018a.

M. Belkin, S. Ma, and S. Mandal. To understand deep learning we need to understand kernel learning. In International Conference on Machine Learning, pages 541–549. PMLR, 2018b.

M. Belkin, D. Hsu, and J. Xu. Two models of double descent for weak features. SIAM Journal on Mathematics of Data Science, 2(4):1167–1180, 2020.

R. Bellman. On the theory of dynamic programming. Proceedings of the national Academy of Sciences, 38(8):716–719, 1952.

R. Bellman. A markovian decision process. Journal of Mathematics and Mechanics, 6(5):679–684, 1957. ISSN 00959057, 19435274. URL http://www.jstor.org/stable/24900506.

N. Benbarka, T. Höfer, A. Zell, et al. Seeing implicit neural representations as fourier series. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 2041–2050, 2022.

E. Bengio, J. Pineau, and D. Precup. Interference and generalization in temporal difference learning. In International Conference on Machine Learning. PMLR, 2020.

Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8):1798–1828, 2013.

J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. Advances in neural information processing systems, 2011.

F. Berkenkamp, A. P. Schoellig, and A. Krause. Safe controller optimization for quadrotors with gaussian processes. In 2016 IEEE international conference on robotics and automation (ICRA), pages 491–496. IEEE, 2016.

E. Berthier, Z. Kobeissi, and F. Bach. A non-asymptotic analysis of non-parametric temporal-difference learning. Advances in Neural Information Processing Systems, 35:7599–7613, 2022.

D. Bertsekas. Dynamic Programming and Optimal Control: Volume I, volume 4. Athena scientific, 2012.

D. P. Bertsekas and H. Yu. Projected equation methods for approximate solution of large linear systems. Journal of Computational and Applied Mathematics, 227(1):27–50, 2009.

D. P. Bertsekas et al. Dynamic programming and optimal control 3rd edition, volume ii. Belmont, MA: Athena Scientific, 1, 2011.

A. Bietti and J. Mairal. On the inductive bias of neural tangent kernels. Advances in Neural Information Processing Systems, 32, 2019.

C. M. Bishop et al. Neural networks for pattern recognition. Oxford university press, 1995.

B. Bordelon, A. Canatar, and C. Pehlevan. Spectrum dependent learning curves in kernel regression and wide neural networks. In International Conference on Machine Learning, pages 1024–1034. PMLR, 2020.

J. A. Boyan. Least-squares temporal difference learning. In International Conference on Machine Learning, pages 49–56, 1999.

S. J. Bradtke and A. G. Barto. Linear least-squares algorithms for temporal difference learning. Machine Learning, 22(1):33–57, 1996.

G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym, 2016.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. Advances in neural information processing systems, 33:1877–1901, 2020.

Q. Cai, Z. Yang, J. D. Lee, and Z. Wang. Neural temporal-difference learning converges to global optima. Advances in Neural Information Processing Systems, 32, 2019.

A. Canatar, B. Bordelon, and C. Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. Nature communications, 12 (1):2914, 2021.

A. Canatar, J. Feather, A. Wakhloo, and S. Chung. A spectral theory of neural prediction and alignment. Advances in Neural Information Processing Systems, 36, 2024.

Y. Cao, Z. Fang, Y. Wu, D.-X. Zhou, and Q. Gu. Towards understanding the spectral bias of deep learning. arXiv preprint arXiv:1912.01198, 2019.

Y. Cao, Z. Fang, Y. Wu, D.-X. Zhou, and Q. Gu. Towards understanding the spectral bias of deep learning. In IJCAI, 2021.

A. Caponnetto and E. De Vito. Optimal rates for the regularized least-squares algorithm. Foundations of Computational Mathematics, 7:331–368, 2007.

S. Chen, G. Chen, and R. Gu. An efficient L2-norm regularized least-squares temporal difference learning algorithm. Knowledge-Based Systems, 45:94–99, 2013.

X. Chen, S. Li, H. Li, S. Jiang, Y. Qi, and L. Song. Generative adversarial user model for reinforcement learning based recommendation system. In International Conference on Machine Learning, pages 1052–1061. PMLR, 2019.

C. Cheng and A. Montanari. Dimension free ridge regression. arXiv preprint arXiv:2210.08571, 2022.

L. Chizat and F. Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. Advances in Neural Information Processing Systems, 31, 2018.

L. Chizat, E. Oyallon, and F. Bach. On lazy training in differentiable programming. Advances in Neural Information Processing Systems, 32, 2019.

K. Ciosek. Properties of the least squares temporal difference learning algorithm. arXiv preprint arXiv:1301.5220, 2013.

K. Cobbe, O. Klimov, C. Hesse, T. Kim, and J. Schulman. Quantifying generalization in reinforcement learning. In International Conference on Machine Learning. PMLR, 2019.

R. Couillet and M. Debbah. Random Matrix Methods for Wireless Communications. Cambridge University Press, 2011.

R. Couillet and Z. Liao. Random matrix methods for machine learning. Cambridge University Press, 2022.

P. Covington, J. Adams, and E. Sargin. Deep neural networks for youtube recommendations. In Proceedings of the 10th ACM conference on recommender systems, pages 191–198, 2016.

C. Dann, G. Neumann, J. Peters, et al. Policy evaluation with temporal differences: A survey and comparison. Journal of Machine Learning Research, 15:809–883, 2014.

R. C. Deo. Machine learning in medicine. Circulation, 132(20):1920–1930, 2015.

K. Dong, Y. Luo, T. Yu, C. Finn, and T. Ma. On the expressivity of neural networks for deep reinforcement learning. In International Conference on Machine Learning, pages 2627–2637, 2020.

S. Du, J. Lee, H. Li, L. Wang, and X. Zhai. Gradient descent finds global minima of deep neural networks. In International conference on machine learning, pages 1675–1685. PMLR, 2019.

Y. Duan, M. Wang, and M. J. Wainwright. Optimal policy evaluation using kernel-based temporal difference methods. arXiv preprint arXiv:2109.12002, 2021.

L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In International conference on machine learning, pages 1407–1416. PMLR, 2018.

Z. Fan and Z. Wang. Spectra of the conjugate kernel and neural tangent kernel for linear-width neural networks. Advances in neural information processing systems, 33:7710–7721, 2020.

A. Farahmand, M. Ghavamzadeh, S. Mannor, and C. Szepesvári. Regularized policy iteration. Advances in Neural Information Processing Systems, 21, 2008.

J. Farebrother, M. C. Machado, and M. Bowling. Generalization and regularization in dqn. arXiv preprint arXiv:1810.00123, 2018.

E. A. Feinberg. Total expected discounted reward mdps: existence of optimal policies, 2011.

S. Fort, P. K. Nowak, S. Jastrzebski, and S. Narayanan. Stiffness: A new perspective on generalization in neural networks, 2020.

R. French. Using semi-distributed representations to overcome catastrophic forgetting in connectionist networks. Proceedings of the AAAI Conference on Artificial Intelligence, 1991.

M. Geist and B. Scherrer. l_1-penalized projected bellman residual. In European Workshop on Reinforcement Learning, pages 89–101. Springer, 2011.

M. Geist, B. Scherrer, A. Lazaric, and M. Ghavamzadeh. A dantzig selector approach to temporal difference learning. arXiv preprint arXiv:1206.6480, 2012.

M. Geist, B. Scherrer, et al. Off-policy learning with eligibility traces: a survey. J. Mach. Learn. Res., 15(1):289–333, 2014.

A. Géron. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. " O'Reilly Media, Inc.", 2022.

M. Ghavamzadeh, A. Lazaric, O. Maillard, and R. Munos. LSTD with random projections. Advances in Neural Information Processing Systems, 23, 2010.

S. Ghiassian and R. S. S. Huizhen Yu, Banafsheh Rafiee. Two geometric input transformation methods for fast online reinforcement learning with neural nets. arXiv, 2018.

S. Ghiassian, B. Rafiee, Y. L. Lo, and A. White. Improving performance in reinforcement learning by breaking generalization in neural networks. In Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems. International Foundation for Autonomous Agents and Multiagent Systems, 2020.

D. Ghosh and M. G. Bellemare. Representations for stable off-policy reinforcement learning. In International Conference on Machine Learning, pages 3556–3565, 2020.

J. Gillberg, J. Bergdahl, A. Sestini, A. Eakins, and L. Gisslén. Technical challenges of deploying reinforcement learning agents for game testing in aaa games. In 2023 IEEE Conference on Games (CoG), pages 1–8. IEEE, 2023.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 2010.

P. W. Glynn and D. L. Iglehart. Importance sampling for stochastic simulations. Management science, 35(11):1367–1392, 1989.

F. Gogianu, T. Berariu, M. Rosca, C. Clopath, L. Busoniu, and R. Pascanu. Spectral normalisation for deep reinforcement learning: an optimisation perspective. arXiv preprint arXiv:2105.05246, 2021.

E. Golikov, E. Pokonechnyy, and V. Korviakov. Neural tangent kernel: A survey. arXiv preprint arXiv:2208.13614, 2022.

H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree. Regularisation of neural networks by enforcing lipschitz continuity. Machine Learning, 2021.

T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine. Soft actor-critic algorithms and applications. CoRR, 2018.

W. Hachem, P. Loubaton, and J. Najim. Deterministic equivalents for certain functionals of large random matrices. The Annals of Applied Probability, pages 875–930, 2007.

B. Hambly, R. Xu, and H. Yang. Recent advances in reinforcement learning in finance. Mathematical Finance, 33(3):437–503, 2023.

T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman. The elements of statistical learning: data mining, inference, and prediction, volume 2. Springer, 2009.

J. He, J. Chen, X. He, J. Gao, L. Li, L. Deng, and M. Ostendorf. Deep reinforcement learning with a natural language action space. arXiv preprint arXiv:1511.04636, 2015.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In Proceedings of the 26th international conference on world wide web, pages 173–182, 2017.

P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger. Deep reinforcement learning that matters. In Proceedings of AAAI Conference on Artificial Intelligence, (AAAI-18), 2018.

J. F. Hernandez-Garcia and R. S. Sutton. Learning sparse representations incrementally in deep reinforcement learning. arXiv, 2019.

M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver. Rainbow: Combining improvements in deep reinforcement learning. In Proceedings of the AAAI conference on artificial intelligence, volume 32, 2018.

D. C. Hoaglin and R. E. Welsch. The hat matrix in regression and anova. The American Statistician, 32(1):17–22, 1978.

M. W. Hoffman, A. Lazaric, M. Ghavamzadeh, and R. Munos. Regularized least squares temporal difference learning with nested L2 and L1 penalization. In European Workshop on Reinforcement Learning, pages 102–114. Springer, 2011.

R. A. Horn and C. R. Johnson. Matrix Analysis. Cambridge University Press, 2012.

D. Hsu, S. M. Kakade, and T. Zhang. Random design analysis of ridge regression. In Conference on learning theory, pages 9–1. JMLR Workshop and Conference Proceedings, 2012.

Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In 2008 Eighth IEEE international conference on data mining, pages 263–272. Ieee, 2008.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning. PMLR, 2015.

R. Islam, P. Henderson, M. Gomrokchi, and D. Precup. Reproducibility of benchmarked deep reinforcement learning tasks for continuous control. arXiv preprint arXiv:1708.04133, 2017.

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. Advances in Neural Information Processing Systems, 31, 2018.

A. Jacot, B. Simsek, F. Spadaro, C. Hongler, and F. Gabriel. Implicit regularization of random feature models. In International Conference on Machine Learning, pages 4631–4640. PMLR, 2020a.

A. Jacot, B. Simsek, F. Spadaro, C. Hongler, and F. Gabriel. Kernel alignment risk estimator: Risk prediction from training data. Advances in neural information processing systems, 33: 15568–15578, 2020b.

J. Johns, C. Painter-Wakefield, and R. Parr. Linear complementarity for regularized policy evaluation and improvement. Advances in neural information processing systems, 23, 2010.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez. Deep reinforcement learning for autonomous driving: A survey. IEEE Transactions on Intelligent Transportation Systems, 23(6):4909–4926, 2021.

J. Z. Kolter and A. Y. Ng. Regularization and feature selection in least-squares temporal difference learning. In Proceedings of the 26th annual international conference on machine learning, pages 521–528, 2009.

G. Konidaris, S. Osentoski, and P. Thomas. Value function approximation in reinforcement learning using the fourier basis. In Twenty-fifth AAAI conference on artificial intelligence, 2011.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6):84–90, 2017.

A. Kumar, R. Agarwal, D. Ghosh, and S. Levine. Implicit under-parameterization inhibits data-efficient deep reinforcement learning. In International Conference on Learning Representations, 2020.

A. Kumar, Z. Fu, D. Pathak, and J. Malik. Rma: Rapid motor adaptation for legged robots. arXiv preprint arXiv:2107.04034, 2021.

M. G. Lagoudakis and R. Parr. Least-squares policy iteration. The Journal of Machine Learning Research, 2003.

A. Lazaric, M. Ghavamzadeh, and R. Munos. Finite-sample analysis of least-squares policy iteration. Journal of Machine Learning Research, 13:3041–3074, 2012.

M. Ledoux. The Concentration of Measure Phenomenon. American Mathematical Soc., 2001.

J. Lee, L. Xiao, S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. Advances in Neural Information Processing Systems, 32, 2019.

A. Li and D. Pathak. Functional regularization for reinforcement learning via learned fourier features. Advances in Neural Information Processing Systems, 2021.

Z. Liao, R. Couillet, and M. W. Mahoney. A random matrix analysis of random fourier features: beyond the Gaussian kernel, a precise phase transition, and the corresponding double descent. Advances in Neural Information Processing Systems, 33:13939–13950, 2020.

A. Likmeta, A. M. Metelli, A. Tirinzoni, R. Giol, M. Restelli, and D. Romano. Combining reinforcement learning with rule-based controllers for transparent and general decision-making in autonomous driving. Robotics and Autonomous Systems, 131:103568, 2020.

T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971, 2015.

L. J. Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. Mach. Learn., 1992.

B. Liu, Q. Cai, Z. Yang, and Z. Wang. Neural trust region/proximal policy optimization attains globally optimal policy. Advances in Neural Information Processing Systems, 32, 2019a.

F. Liu, Z. Liao, and J. Suykens. Kernel regression in high dimensions: Refined analysis beyond double descent. In International Conference on Artificial Intelligence and Statistics, pages 649–657. PMLR, 2021.

V. Liu, R. Kumaraswamy, L. Le, and M. White. The utility of sparse representations for control in reinforcement learning. Proceedings of the AAAI Conference on Artificial Intelligence, Jul. 2019b.

Z. Liu, X. Li, B. Kang, and T. Darrell. Regularization matters in policy optimization-an empirical study on continuous control. In International Conference on Learning Representations, 2020.

D. Lopez-Paz and M. Ranzato. Gradient episodic memory for continual learning. In Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017.

C. Louart, Z. Liao, and R. Couillet. A random matrix approach to neural networks. The Annals of Applied Probability, 28(2):1190–1248, 2018.

J. Luketina, N. Nardelli, G. Farquhar, J. Foerster, J. Andreas, E. Grefenstette, S. Whiteson, and T. Rocktäschel. A survey of reinforcement learning informed by natural language. arXiv preprint arXiv:1906.03926, 2019.

X. Luo, Q. Meng, D. He, W. Chen, and Y. Wang. I4r: Promoting deep reinforcement learning by the indicator for expressive representations. In IJCAI, 2020.

C. Lyle, M. Rowland, and W. Dabney. Understanding and preventing capacity loss in reinforcement learning. In International Conference on Learning Representations, 2021.

C. Lyle, M. Rowland, W. Dabney, M. Kwiatkowska, and Y. Gal. Learning dynamics and generalization in deep reinforcement learning. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, Proceedings of the 39th International Conference on Machine Learning, volume 162 of Proceedings of Machine Learning Research, pages 14560–14581. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/lyle22a.html.

V. A. Marchenko and L. A. Pastur. Distribution of eigenvalues for some sets of random matrices. Matematicheskii Sbornik, 114(4):507–536, 1967.

M. McCloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In Psychology of learning and motivation. Elsevier, 1989.

S. Mehrkanoon and J. A. Suykens. Deep hybrid neural-kernel networks using random fourier features. Neurocomputing, 2018.

S. Mei and A. Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. Communications on Pure and Applied Mathematics, 75(4): 667–766, 2022.

S. Mei, A. Montanari, and P.-M. Nguyen. A mean field view of the landscape of two-layer neural networks. Proceedings of the National Academy of Sciences, 115(33):E7665–E7671, 2018.

V. Meshram, K. Patil, V. Meshram, D. Hanchate, and S. Ramkteke. Machine learning in agriculture domain: A state-of-art survey. Artificial Intelligence in the Life Sciences, 1:100010, 2021.

R. Mitra and G. Kaddoum. Random fourier feature based deep learning for wireless communications. arXiv preprint arXiv:2101.05254, 2021.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, 2015.

T. M. Moerland, J. Broekens, A. Plaat, C. M. Jonker, et al. Model-based reinforcement learning: A survey. Foundations and Trends® in Machine Learning, 16(1):1–118, 2023.

P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever. Deep double descent: Where bigger models and more data hurt. Journal of Statistical Mechanics: Theory and Experiment, 2021(12):124003, 2021.

A. Nedić and D. P. Bertsekas. Least squares policy evaluation algorithms with linear function approximation. Discrete Event Dynamic Systems, 13(1-2):79–110, 2003.

B. Neyshabur. Implicit regularization in deep learning. arXiv preprint arXiv:1709.01953, 2017.

B. Neyshabur, R. Tomioka, and N. Srebro. Norm-based capacity control in neural networks. In Conference on Learning Theory, 2015.

B. Neyshabur, S. Bhojanapalli, D. Mcallester, and N. Srebro. Exploring generalization in deep learning. Advances in Neural Information Processing Systems, 2017.

E. Nikishin, M. Schwarzer, P. D'Oro, P.-L. Bacon, and A. Courville. The primacy bias in deep reinforcement learning. In International Conference on Machine Learning, pages 16828–16847. PMLR, 2022.

Y. Pan, K. Banman, and M. White. Fuzzy tiling activations: A simple approach to learning sparse representations online. In International Conference on Learning Representations, 2020.

R. Parr, C. Painter-Wakefield, L. Li, and M. Littman. Analyzing feature generation for value-function approximation. In Proceedings of the 24th international conference on Machine learning, pages 737–744, 2007.

R. Parr, L. Li, G. Taylor, C. Painter-Wakefield, and M. L. Littman. An analysis of linear models, linear value-function approximation, and feature selection for reinforcement learning. In Proceedings of the 25th international conference on Machine learning, pages 752–759, 2008.

J. Perolat, B. De Vylder, D. Hennes, E. Tarassov, F. Strub, V. de Boer, P. Muller, J. T. Connor, N. Burch, T. Anthony, et al. Mastering the game of stratego with model-free multiagent reinforcement learning. Science, 378(6623):990–996, 2022.

M. Petrik. An analysis of laplacian methods for value function approximation in mdps. In IJCAI, pages 2574–2579, 2007.

M. Plappert, M. Andrychowicz, A. Ray, B. McGrew, B. Baker, G. Powell, J. Schneider, J. Tobin, M. Chociej, P. Welinder, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. arXiv preprint arXiv:1802.09464, 2018.

M. Popova, O. Isayev, and A. Tropsha. Deep reinforcement learning for de novo drug design. Science advances, 4(7):eaap7885, 2018.

M. L. Puterman. Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, 2014.

A. Raffin. Rl baselines3 zoo. https://github.com/DLR-RM/rl-baselines3-zoo, 2020.

A. Raffin, A. Hill, M. Ernestus, A. Gleave, A. Kanervisto, and N. Dormann. Stable baselines3. https://github.com/DLR-RM/stable-baselines3, 2019.

N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville. On the spectral bias of neural networks. In International Conference on Machine Learning, pages 5301–5310. PMLR, 2019.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. Advances in Neural Information Processing Systems, 20, 2007.

A. Rajeswaran, K. Lowrey, E. Todorov, and S. Kakade. Towards generalization and simplicity in continuous control. In Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.

M. Riedmiller. Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method. In European conference on machine learning, pages 317–328. Springer, 2005.

M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. arXiv preprint arXiv:1810.11910, 2018.

H. Robbins and S. Monro. A stochastic approximation method. The Annals of Mathematical Statistics, pages 400–407, 1951.

M. Rosca, T. Weber, A. Gretton, and S. Mohamed. A case for new neural network smoothness constraints. In Proceedings on "I Can't Believe It's Not Better!" at NeurIPS Workshops. PMLR, 2020.

G. Rotskoff and E. Vanden-Eijnden. Parameters as interacting particles: long time convergence and asymptotic error scaling of neural networks. Advances in Neural Information Processing Systems, 31, 2018.

A. Rudi and L. Rosasco. Generalization properties of learning with random features. Advances in neural information processing systems, 30, 2017.

G. A. Rummery and M. Niranjan. On-line Q-learning using Connectionist Systems, volume 37. University of Cambridge, Department of Engineering Cambridge, UK, 1994.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. International journal of computer vision, 115:211–252, 2015.

T. Salimans and D. P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. Advances in neural information processing systems, 2016.

K. Scaman and A. Virmaux. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018.

T. Schaul, D. Borsa, J. Modayil, and R. Pascanu. Ray interference: a source of plateaus in deep reinforcement learning. arXiv preprint arXiv:1904.11455, 2019.

J. Schmidhuber. Deep learning in neural networks: An overview. Neural networks, 61:85–117, 2015.

B. Schölkopf and A. J. Smola. Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, 2002.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. Science, 362(6419):1140–1144, 2018.

J. B. Simon, M. Dickens, D. Karkada, and M. Deweese. The eigenlearning framework: A conservation law perspective on kernel ridge regression and wide neural networks. Transactions on Machine Learning Research, 2023a.

J. B. Simon, D. Karkada, N. Ghosh, and M. Belkin. More is better in modern machine learning: when infinite overparameterization is optimal and overfitting is obligatory. arXiv preprint arXiv:2311.14646, 2023b.

S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. Machine learning, 38:287–308, 2000.

Z. Song, R. E. Parr, X. Liao, and L. Carin. Linear feature encoding for reinforcement learning. Advances in neural information processing systems, 29, 2016.

J. Stachurski. Economic dynamics: theory and computation. MIT Press, 2009.

C. Stephenson and T. Lee. When and how epochwise double descent happens. arXiv preprint arXiv:2108.12006, 2021.

R. S. Sutton. Learning to predict by the methods of temporal differences. Machine Learning, 3 (1):9–44, 1988.

R. S. Sutton and A. G. Barto. Reinforcement Learning: An Introduction. MIT Press, 2018.

M. Tagorti and B. Scherrer. On the rate of convergence and error bounds for LSTD(λ). In International Conference on Machine Learning, pages 1521–1529. PMLR, 2015.

M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng. Fourier features let networks learn high frequency functions in low dimensional domains. arXiv preprint arXiv:2006.10739, 2020.

T. Tao. Topics in Random Matrix Theory, volume 132. American Mathematical Soc., 2012.

N. Tasfi. Pygame learning environment. https://github.com/ntasfi/PyGame-Learning-Environment, 2016.

G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, et al. Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805, 2023.

V. Thomas. On the role of overparameterization in off-policy temporal difference learning with linear function approximation. Advances in Neural Information Processing Systems, 35:37228–37240, 2022.

E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012.

H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971, 2023.

M. Towers, J. K. Terry, A. Kwiatkowski, J. U. Balis, G. d. Cola, T. Deleu, M. Goulão, A. Kallinteris, A. K.G., M. Krimmel, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, A. T. J. Shen, and O. G. Younis. Gymnasium, Mar. 2023. URL https://zenodo.org/record/8127025.

P. Trautman and A. Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 797–803. IEEE, 2010.

J. Tsitsiklis and B. Van Roy. Analysis of temporal-difference learning with function approximation. Advances in Neural Information Processing Systems, 9, 1996.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. Nature, 575(7782):350–354, 2019.

H. Wang, S. Zheng, C. Xiong, and R. Socher. On the generalization gap in reparameterizable reinforcement learning. In International Conference on Machine Learning. PMLR, 2019.

S. Wang, H. Wang, and P. Perdikaris. On the eigenvector bias of fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks. Computer Methods in Applied Mechanics and Engineering, 2021.

C. J. Watkins and P. Dayan. Q-learning. Machine Learning, 8:279–292, 1992.

C. Xiao, B. Dai, J. Mei, O. A. Ramirez, R. Gummadi, C. Harris, and D. Schuurmans. Understanding and leveraging overparameterization in recursive value estimation. In International Conference on Learning Representations, 2021.

Z.-Q. J. Xu, Y. Zhang, and T. Luo. Overview frequency principle/spectral bias in deep learning. arXiv preprint arXiv:2201.07395, 2022.

G. Yang, A. Ajay, and P. Agrawal. Overcoming the spectral bias of neural value approximation. In International Conference on Learning Representations, 2021.

R. D. Yates. A framework for uplink power control in cellular radio systems. IEEE Journal on Selected Areas in Communications, 13(7):1341–1347, 1995.

P. C. Young. Recursive Estimation and Time-Series Analysis: an Introduction. Springer science & business media, 2012.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning (still) requires rethinking generalization. Communications of the ACM, 64(3):107–115, 2021.

S. Zhang and R. S. Sutton. A deeper look at experience replay. arXiv preprint arXiv:1712.01275, 2017.

ECOLE DOCTORALE

Titre: Analyse Expérimentale et Théorique des Algorithmes d'Apprentissage par Renforcement

Mots clés: Apprentissage automatique, apprentissage par renforcement, réseaux de neurones, matrice aléatoire, prétraitement

Résumé: En apprentissage par renforcement (RL), un agent apprend comment agir dans un environnement inconnu de façon à maximiser sa récompense sur le long terme. Ces dernières années, l'utilisation de réseaux de neurones a conduit à de nombreuses avancées, notamment en termes de scalabilité. Cependant, de nombreuses lacunes subsistent dans notre compréhension de la meilleure manière d'employer les réseaux de neurones en RL. Dans cette thèse, nous proposons d'améliorer l'utilisation des réseaux de neurones en RL de deux manières, présentées dans deux parties distinctes. La première partie présente une analyse théorique de l'impact du nombre de paramètres sur la performance d'apprentissage. La seconde partie propose un prétraitement simple des données, basé sur la série de Fourier, qui améliore empiriquement les performances des réseaux de neurones de plusieurs façons.

Dans la première partie de cette thèse, nous étudions l'influence du nombre de paramètres sur la performance. Nous identifions le rapport entre le nombre de paramètres et le nombre d'états visités comme un facteur crucial. En particulier, nous observons un phénomène de double descente caractérisé par une chute soudaine de performance au-delà d'un rapport de un. Notre analyse est basée sur l'algorithme de Least-Squares Temporal Difference learning (LSTD) doté de caractéristiques aléatoires et d'un terme de regularisation L2 dans un régime asymptotique, où le nombre de paramètres et d'états visités tendent vers l'infini tout en maintenant un rapport constant. Nous dérivons des limites déterministes de mesures de performance qui comportent des termes correctifs induits par le rapport fini nombre de paramètres/états visités. Nous associons expérimentallement ces termes correctifs au phénomène de double descente et à une régularisation implicite du modèle. Nous démontrons que ces termes correctifs diminuent avec l'augmentation de la régularisation L2, du nombre de paramètres, ou de la diminution du nombre d'états non visités.

Dans la seconde partie de cette thèse, nous proposons l'étude d'un prétraitement des données basé sur la série de Fourier. Nous présentons des expériences indiquant que ce prétraitement peut conduire à des améliorations significatives des performances, en termes de récompenses obtenues et de données utilisées. De plus, nous observons que ce prétraitement favorise une plus grande robustesse face aux hyperparamètres, conduit à l'élaboration de politiques plus régulières, et bénéficie au processus d'entraînement en réduisant l'interférence d'apprentissage, en encourageant l'apprentissage de caractéristiques distinctes, et en augmentant l'expressivité des caractéristiques apprises.

Title: Experimental and Theoretical Analysis of Reinforcement Learning Algorithms

Keywords: Machine Learning, Reinforcement Learning, neural networks, random matrix, preprocessing

Abstract: In Reinforcement Learning (RL), an agent learns how to act in an unknown environment in order to maximize its reward in the long run. In recent years, the use of neural networks has led to breakthroughs, e.g., in scalability. However, there are still gaps in our understanding of how to best employ neural networks in RL. In this thesis, we improve the usability of neural networks in RL in two ways, presented in two separate parts. First, we present a theoretical analysis of the influence of the number of parameters on learning performance. Second, we propose a simple feature preprocessing based on the Fourier series, which empirically improves performance in several ways.

In the first part of this thesis, we study how the number of parameters influences performance. We identify the ratio between the number of parameters and the number of visited states as a crucial factor. We observe a double descent phenomenon, i.e., a sudden drop in performance around the parameter/state ratio of one. Our analysis is based on the Least-Squared Temporal Difference (LSTD) algorithm with random features and an L_2 regularization penality in an asymptotic regime, as both the number of parameters and states go to infinity while maintaining a constant ratio. We derive deterministic limits of performance measures that feature correction due to the constant ratio between the number of parameters and distinct visited states. We experimentally associate those correction terms with the double descent phenomenon and an implict regularization of the model. We demonstrate that the correction terms vanish as either the L2 regularization increases, the number of parameters increases, or the number of unvisited states decreases.

In the second part of this thesis, we study the preprocessing of features through a Fourier series. We present experiments indicating that this can lead to significant performance gains in terms of rewards and sample efficiency. Furthermore, we observe that this preprocessing increases the robustness with respect to hyperparameters, leads to smoother policies, and benefits the training process by reducing learning interference, encouraging sparsity, and increasing the expressiveness of the learned features.

Institut Polytechnique de Paris

91120 Palaiseau, France