# Stochastic Second Order Methods and Finite Time Analysis of Policy Gradient Methods

Rui Yuan

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2023IPPAT010

Thèse de doctorat

# Stochastic Second Order Methods and Finite Time Analysis of Policy Gradient Methods

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom Paris

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

Thèse présentée et soutenue à Palaiseau, le 17 mars 2023, par

**RUI YUAN**

Composition du Jury :

| | |
|---|---|
| Olivier Fercoq<br>Professeur, Télécom Paris (LTCI) | Président / Examinateur |
| Niao He<br>Professeur Assistant, ETH Zurich | Rapporteur |
| Matthieu Geist<br>Professeur, Université de Lorraine; Chercheur Scientifique, Google Brain | Rapporteur |
| Nicolas Le Roux<br>Professeur Assistant, Université de Montréal et McGill; Chercheur Principal Senior, Microsoft Research | Examinateur |
| François Roueff<br>Professeur, Télécom Paris (LTCI) | Directeur de thèse |
| Robert M. Gower<br>Chercheur Scientifique, Flatiron Institute (CCM); précédemment Professeur Assistant, Télécom Paris | Co-encadrant de thèse / Invité |
| Alessandro Lazaric<br>Chercheur Scientifique, Meta AI (FAIR) | Co-encadrant de thèse / Invité |

*À ma partenaire civile Xinhe, à mes parents Yao et Ximing*

# Remerciements

This journey is impossible without the help of my colleagues, my friends and my family. My first words of acknowledgement are for my advisors Robert and Alessandro. I would like to express my sincere gratitude to both of you, for being a mentor, a collaborator and a friend. Robert, thank you for introducing me to the world of optimization. We worked together during my internship even before my thesis, which became the starting point of my research project. Thank you for your patient and unwavering guidance which is an invaluable part of my PhD experience. I now realize how fortunate I was to have an advisor as available as you were to discuss about science, projects and broader subjects. You are a great example of what I want to be in my future career. Alessandro, thank you for giving me the opportunity to work at Meta. I am very grateful for your research advice and your care, especially during the difficult WFH period. Today I am proud to say that I am becoming an independent researcher, as you expected of me at the beginning of my thesis.

Je tiens également à remercier François Roueff, mon directeur de thèse, pour sa disponibilité et la liberté qu'il m'a laissée durant ma thèse. I would also like to thank Niao He and Matthieu Geist for their careful review of my thesis, and my defense committee members, Nicolas Le Roux and Olivier Fercoq, for their thoughtful questions during the defense.

Many thanks to my wonderful collaborators: Lin Xiao, Guillaume Garrigos, Jiabin Chen, Carlo Alfano, Simon Du and Patrick Rebeschini. In particular, Lin, I really learned a lot from your proving and writing skills. It was a pleasure working with you. I also want to thank you for coming to my defense online, even though it was 6 a.m. in Seattle. Guillaume, j'ai beaucoup apprécié que t'as vérifié chaque détail des preuves et que tu étais toujours motivé par notre projet. Je voudrais également te remercier pour notre discussion sur la carrière académique. Cet échange m'a beaucoup aidé lorsque j'ai dû prendre des décisions importantes.

Bien sûr, je remercie toute l'équipe de Meta AI, avec qui j'ai partagé de grands moments. En particulier, Nicolas Usunier et Hervé Jégou pour leur écoute permanente, Xavier Martinet pour toutes les techniques informatiques qu'il m'a montrées, Guillaume Lample, Aurélien Rodriguez, et François Charton pour m'avoir fait découvrir la beauté de l'IA en mathématiques. Et, bien évidemment, je remercie l'équipe exceptionnelle de CIFREs, qui a rendu ce projet de doctorat tellement plus agréable au quotidien. En particulier, je voudrais remercier Jean

Tarbouriech, qui m'a appris toutes ses magies en Latex et Keynote et, surtout, qui m'a beaucoup encouragé dans les moments difficiles. I was also very lucky to have Maria Zameshina as my neighbor, with whom I had casual conversations during the difficult moment of multiple confinements.

Merci à mes amis pour leur compagnie en dehors de la recherche. Surtout Adrien, pour nos matches de foot au Stade de France, au Parc des Princes ou de tennis à Roland-Garros, pour m'avoir initié à la coinche, et pour tes tours de magie tous aussi bluffants les uns que les autres. Clément, pour toutes nos soirées galettes des rois. Jean-Louis, pour ta gentillesse et nos passionnantes discussions artistiques et politiques. Shufan, pour tes bons plans de randonnée. Xujia, Fangyan, Vincent, Sihan, Gaston, Mingxing et tous ceux que j'ai oubliés pour tous les bons moments passés ensemble.

Un grand merci à mes parents qui m'ont donné le privilège de poursuivre mes rêves et m'ont inculqué l'amour des sciences et des mathématiques dès mon plus jeune âge.

Pour finir, Xinhe, tu es entrée dans ma vie presque en même temps que ma thèse. Je te remercie infiniment pour ton soutien quotidien et pour avoir partagé beaucoup de mon stress pendant cette période difficile. Et surtout merci pour ton amour indéfectible qui a fait de moi une meilleure personne. Je suis sûr que l'avenir nous réserve encore bien des aventures.

# Résumé

Pour résoudre les problèmes de l'apprentissage automatique à grande échelle, les méthodes de premier ordre telles que la descente du gradient stochastique et l'ADAM sont les méthodes de choix en raison de leur coût pas cher par itération. Le problème des méthodes du premier ordre est qu'elles peuvent nécessiter un réglage lourd des paramètres et/ou une connaissance des paramètres du problème. Il existe aujourd'hui un effort considérable pour développer des méthodes du second ordre stochastiques efficaces afin de résoudre des problèmes de l'apprentissage automatique à grande échelle. La motivation est qu'elles demandent moins de réglage des paramètres et qu'elles convergent pour une plus grande variété de modèles et de datasets. Dans la première partie de la thèse, nous avons présenté une approche de principe pour désigner des méthodes de Newton stochastiques à fin de résoudre à la fois des équations non linéaires et des problèmes d'optimisation d'une manière efficace. Notre approche comporte deux étapes. Premièrement, nous pouvons réécrire les équations non linéaires ou le problème d'optimisation sous forme d'équations non linéaires souhaitées. Ensuite, nous appliquons de nouvelles méthodes du second ordre stochastiques pour résoudre ce système d'équations non linéaires. Grâce à notre approche générale, nous présentons de nombreux nouveaux algorithmes spécifiques du second ordre qui peuvent résoudre efficacement les problèmes de l'apprentissage automatique à grande échelle sans nécessiter de connaissance du problème ni de réglage des paramètres. Dans la deuxième partie de la thèse, nous nous concentrons sur les algorithmes d'optimisation appliqués à un domaine spécifique : l'apprentissage par renforcement (RL). Cette partie est indépendante de la première partie de la thèse. Pour atteindre de telles performances dans les problèmes de RL, le policy-gradient (PG) et sa variante, le policy-gradient naturel (NPG), sont les fondements de plusieurs algorithmes de l'état de l'art (par exemple, TRPO et PPO) utilisés dans le RL profond. Malgré le succès empirique des méthodes de RL et de PG, une compréhension théorique solide du PG original a longtemps fait défaut. En utilisant la structure du RL du problème et des techniques modernes de preuve d'optimisation, nous obtenons nouvelles analyses en temps fini de la PG et de la NPG. Grâce à notre analyse, nous apportons également de nouvelles perspectives aux méthodes avec de meilleurs choix d'hyperparamètres.

# Abstract

To solve large scale machine learning problems, first-order methods such as stochastic gradient descent and ADAM are the methods of choice because of their low cost per iteration. The issue with first order methods is that they can require extensive parameter tuning, and/or knowledge of the parameters of the problem. There is now a concerted effort to develop efficient stochastic second order methods to solve large scale machine learning problems. The motivation is that they require less parameter tuning and converge for wider variety of models and datasets. In the first part of the thesis, we presented a principled approach for designing stochastic Newton methods for solving both nonlinear equations and optimization problems in an efficient manner. Our approach has two steps. First, we can re-write the nonlinear equations or the optimization problem as desired nonlinear equations. Second, we apply new stochastic second order methods to solve this system of nonlinear equations. Through our general approach, we showcase many specific new second-order algorithms that can solve the large machine learning problems efficiently without requiring knowledge of the problem nor parameter tuning. In the second part of the thesis, we then focus on optimization algorithms applied in a specific domain: reinforcement learning (RL). This part is independent to the first part of the thesis. To achieve such high performance of RL problems, policy gradient (PG) and its variant, natural policy gradient (NPG), are the foundations of the several state of the art algorithms (e.g., TRPO and PPO) used in deep RL. In spite of the empirical success of RL and PG methods, a solid theoretical understanding of even the "vanilla" PG has long been elusive. By leveraging the RL structure of the problem together with modern optimization proof techniques, we derive new finite time analysis of both PG and NPG. Through our analysis, we also bring new insights to the methods with better hyperparameter choices.

# Contents

# Chapter 1

# Introduction

## 1.1 Stochastic Second Order Methods in Optimization

### 1.1.1 Context and scope

**Optimization in AI.** We have witnessed the progress of artificial intelligence (AI), also called machine learning, over the last decade. It has been widely applied in society from natural language processing, computer vision to online advertising and even robotics, to name a few. For instance, in natural language processing, there are machine translation problem, like google translate, and chat bot problem, like ChatGPT. In computer vision, there are image segmentation, image classification, object detection problems, and so on. In particular, optimization plays an important role in AI. That is, the problem of interest can be formalized as a loss function $f$ of parameter $w \in \mathbb{R}^d$ in dimension $d$. For instance, a function $f$ can look like the one in Figure 1.1. Here the dimension of the function is 3, for simplicity. The goal is to design an algorithm to find automatically the best parameter $w^*$ to minimize the loss function so that it can fit the AI model.

**First order methods.** A very classic method to solve this is the iterative method – *Gradient Descent*. That is, at $k$-th iteration, the parameter $w^k$ is updated as follows,

$$w^{k+1} \;=\; w^k \;-\; \eta^k \, \nabla f(w^k),$$

where $\eta^k$ is the step size. Gradient descent is also called the first order method, because it involves the first derivative of the function.

This is a very simple method in optimization. However, the issue with first order methods is that they can require extensive parameter tuning, and/or knowledge of the parameters of

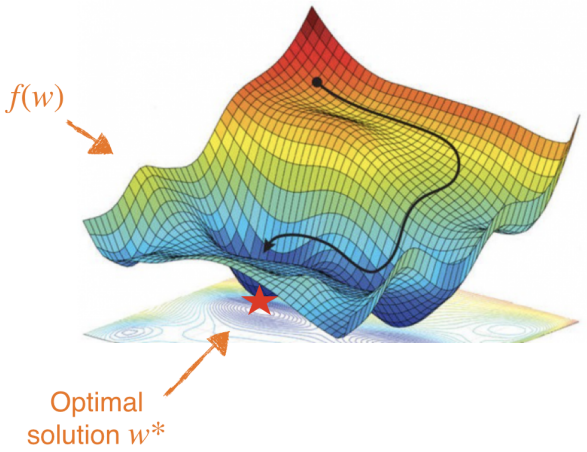


**Figure 1.1** – Optimization paradigm.

the problem. For instance, the step size depends on the scale of the function. Indeed, given a function $f$, the minimum of $f$ is identical to the same problem multiplied by a positive number $C$ as shown in Figure 1.2. While the updates of their gradient descent are not the same. The second update is proportional to $C$. Therefore, gradient descent is hard to tune, as it depends heavily on the scale of the function. Consequently, the practitioner needs to inject domain knowledge at a higher-level of how modeling components interact to make a first order method work well. The reliance on first order methods ultimately restricts the choice and development of alternative models.

$$\arg\min_{w \in \mathbb{R}^d} f(w) \quad \Leftrightarrow \quad \arg\min_{w \in \mathbb{R}^d} C \times f(w)$$


$$w^{k+1} = w^k - \eta^k \nabla f(w^k) \quad \Leftrightarrow \quad w^{k+1} = w^k - \eta^k C \nabla f(w^k)$$

**Figure 1.2** – Gradient descent depends on the scale of the function.

**Second order methods.** Now let look at the *Newton's method*, which is another classic iterative method for minimizing $f$. That is, at $k$-th iteration, we have

$$w^{k+1} = w^k - \eta \nabla^2 f(w^k)^{-1} \nabla f(w^k).$$

Newton's method is also called the second order method, because it involves the second derivative of the function.

Because of the access of the second order information, the update of the Newton's method is able to capture the local curvature of the function $f$, which allows to have an improved update direction compared to the gradient descent method. This leads to a faster convergence of Newton's method compared to the gradient descent method as well.

More importantly, Newton's method is scale invariant. Indeed, when the problem is multiplied by a positive number $C$, the update of Newton's method remains the same. That is,

$$w^{k+1} = w^k - \eta \nabla^2 f(w^k)^{-1} \nabla f(w^k) \quad \Longleftrightarrow \quad w^{k+1} = w^k - \eta \nabla^2 (C \cdot f(w^k))^{-1} \nabla (C \cdot f(w^k)).$$

Consequently, it is much easier to tune the step size of Newton's method than the one of gradient descent. However, the inverse operator $\nabla^2 f(w^k)^{-1}$ is expensive to compute. The cost per iteration is $d^3$, which is prohibitive when $d$ is large. Here comes a natural question:

*Can we achieve the best of the two worlds ?*

That is, having an algorithm that does not suffer from parameter tuning, such as step size, and still maintains efficient computational cost that is as cheap as first order methods. In the first part of the thesis, this question will be addressed positively.

### 1.1.2 New stochastic second order methods

The significant increase in the number of data samples in recent machine learning applications (such as web advertising and bioinformatics) precludes the use of either the exact gradient methods or the exact Newton's methods. To solve large scale machine learning problems, stochastic first order methods such as *Stochastic Gradient Descent* (Robbins and Monro, 1951, SGD), ADAGRAD (Duchi et al., 2011) and ADAM (Kingma and Ba, 2015) are the methods of choice in practice because of their low cost per iteration. As mentioned above, tuning step sizes can be time consuming. There is now a concerted effort to develop efficient stochastic second order methods (Gupta et al., 2018) to solve large scale machine learning problems. The motivation is that they require less parameter tuning and converge for wider variety of models and datasets.

In this first part of the thesis, we presented a principled approach for designing stochastic second order methods for solving both nonlinear equations and finite sum optimization problems in an efficient manner. Our approach has two steps. First, we can re-write the nonlinear equations or the finite sum problem as desired nonlinear equations, using variable splitting or function splitting tricks. Second, we apply new stochastic second order methods to

solve this system of nonlinear equations. For the new stochastic second order methods, we introduce Sketched Newton-Raphson (SNR) in Chapter 2 and Sketched Newton-Raphson with Variable Metric (SNRVM), which is an extension of SNR in Section 3.4 Chapter 3. Both SNR and SNRVM are variants of the Newton-Raphson (NR) method and can have the same cost as SGD per iteration when solving finite sum problems. This overcomes the issue of the original NR method where the cost per iteration is prohibitive when the dimension of the nonlinear equations is large. The idea of having low cost per iteration is the use of the stochastic tool: sketch-and-project technique (Gower and Richtárik, 2015b), which allows us to reduce the dimension of the Newton system and hence make the cost per iteration cheap. Through the general SNR and SNRVM, we showcase many specific new second order algorithms that can solve the large machine learning problems with finite sum structure efficiently without requiring much knowledge of the problem nor parameter tuning. See more details of our contributions next.

### 1.1.3 Outline and contributions of Part I

The general research goal driving the first part of the thesis can be framed as

*designing an optimization algorithm for solving large scale machine learning problems, that is incremental, efficient, scales well with the feature dimension, and that requires less parameter tuning.*

To reach this goal, our first attempt is to propose a new stochastic second order method – Sketched Newton-Raphson (SNR) in Chapter 2 , which combines the Newton-Raphson method with the sketch-and-project technique (Gower and Richtárik, 2015b). Overall, our main contribution is a thorough analysis of SNR through various forms (e.g. TCS in Section 2.8, nonlinear Kaczmarz method (Wang et al., 2022) and Stochastic Newton method (Rodomanov and Kropotov, 2016; Kovalev et al., 2019, SNM)), accompanied with the global convergence theory of SNR. At a high level concept, we demonstrate how SNR opens the gate of designing and analyzing many new stochastic second order methods (e.g. TCS and nonlinear Kaczmarz method (Wang et al., 2022)), or recovering existent stochastic second order methods with their new convergence theories (e.g. SNM (Rodomanov and Kropotov, 2016; Kovalev et al., 2019) and the original Newton-Raphson method). As for the convergence theories of SNR, we reformulate the method as a variant of the SGD method. This reformulation is interesting. It turns that the reformulation is always a smooth and interpolated function. The interpolation condition means that, the function has zero noise for stochastic gradient at the optimum. These properties are frequently used in the SGD convergence proofs (Ma et al., 2018; Vaswani et al., 2019a). Thanks to this reformulation, we establish the global convergence theory and rates of convergence under convex type assumptions by leveraging proof techniques of SGD. Being beneficial from the reformulation, our theory also provides a new global convergence theory

for the original Newton-Raphson method under strictly weaker assumptions as compared to the classic monotone convergence theory (Ortega and Rheinboldt, 2000; Deuflhard, 2011). Through the general framework of SNR, we advocate to "Tossing-Coin-Sketch" – in short, TCS – which solves large scale machine learning problems efficiently. Regarding to our research goal, TCS is incremental. It is able to require only a single data point per iteration. When sampling only one single data point per iteration, TCS scales well with the feature dimension. In this case, it has the same cost per iteration as SGD. TCS also requires less parameter tuning of the step size compared to the first order method (e.g. SGD and ADAM (Kingma and Ba, 2015)), which is expected as being a second order method. We show through numerical experiments that TCS is competitive as compared to classical variance reduced gradient methods (e.g. SAG (Schmidt et al., 2017) and SVRG (Johnson and Zhang, 2013)). However, TCS is efficient when using a batch sampling. It converges slowly in experiment for single data point per iteration. To make TCS work efficiently, one needs to tune the sketch size. Consequently, the research goal is partially achieved, as we still need to tune the sketch size to make the algorithm efficient.

Motivated by our research goal, more specifically, by finding a new algorithm on top of TCS that requires less parameter tuning, including not only the step size but also the sketch size, we propose *Stochastic Average Newton* (SAN) in Chapter 3. Using similar approach of designing new stochastic second order methods from SNR, we develop SAN, which is incremental, in that it requires only a single data point per iteration. It is also cheap to implement when solving large scale regularized generalized linear models, with the same cost per iteration as SGD. We show through extensive numerical experiments that SAN is parameter-free and remains competitive as compared to variance reduced gradient methods (e.g. SAG (Schmidt et al., 2017) and SVRG (Johnson and Zhang, 2013)). To provide a convergence theory of our methods, we extend SNR to SNRVM that allows for a variable metric and that includes SAN as a special case.

In total, Part I conveys the conceptual message that it is possible to design many new stochastic second order methods that are able to solve large scale machine learning problems efficiently without the knowledge about the problem, neither parameter tuning.

## 1.2 Finite Time Analysis of Policy Gradient Methods in Reinforcement Learning

In the second part of the thesis, we then focus on optimization algorithms applied in a specific domain: reinforcement learning (RL). This part is independent to the first part of the thesis.

### 1.2.1 Reinforcement learning

We have got some of the most impressive AI results from RL, such as game playing (Mnih et al., 2015; Silver et al., 2017; OpenAI et al., 2019; Vinyals et al., 2019), autonomous driving (Shalev-Shwartz et al., 2016; Kiran et al., 2022), robotics (Kober et al., 2013; Levine et al., 2016; Gu* et al., 2017; Levine et al., 2018) and beyond. So, what is RL ? The short answer is that, RL is about learning in an unknown environment through trial and failure to make sequential decisions.

**Markov decision process (MDP).** In the traditional RL paradigm as shown in Figure 1.3, an agent interacts with an environment modeled as a Markov decision process (Puterman, 1994, MDP). At time $t$, the agent is at state $s_t$ somewhere in the environment. The environment can be seen as a state space $\mathcal{S}$. Then the agent takes an action $a_t$ among all possible actions in the action space $\mathcal{A}$. Based on the current state and action $s_t, a_t$, the environment will lead the agent to the next state $s_{t+1}$ with the transition probability $\mathcal{P}$, also known as the dynamic of the environment. Through this interaction, the agent will get a reward $r(s_t, a_t)$[1]. In particular, the action $a_t$ is chosen through the policy $\pi \in \Delta(\mathcal{A})^{\mathcal{S}}$, which is a function from state space $\mathcal{S}$ to the probability simplex of the action space $\Delta(\mathcal{A})$. We note $\pi_{s_t,a_t} \in \mathbb{R}$ the density of choosing action $a_t$ over action space at $s_t$ and $\pi_{s_t} \in \Delta(\mathcal{A})$ is the distribution over actions at state $s_t$. Thus, a policy induces a distribution over trajectories $\{s_t, a_t, r(s_t, a_t)\}_{t\geq 0}$.

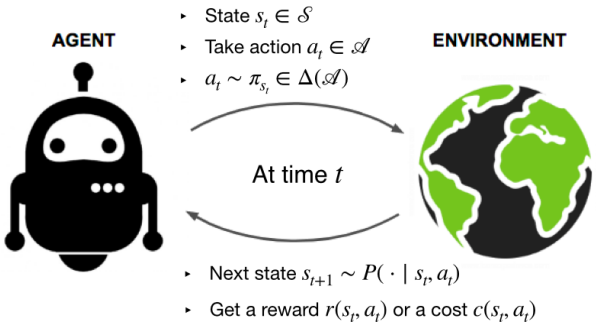


**Figure 1.3** – An agent interacts with the environment, trying to take smart actions to maximize cumulative rewards.

**Policy optimization.** The objective of the agent is to solve the MDP. That is, to find the optimal policy such that the total expected cumulative rewards over the trajectory $V_\rho(\pi)$, defined as

$$V_\rho(\pi) \overset{\text{def}}{=} \mathbb{E}_{s_0\sim\rho,\ a_t\sim\pi_{s_t},\ s_{t+1}\sim\mathcal{P}(\cdot|s_t,a_t)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right],$$

[1] In Chapter 5, we use cost instead of reward to better align with the minimization convention in the optimization literature.

are maximum. The problem is also called policy optimization. Here the expectation is with respect to the initial state distribution $\rho \in \Delta(\mathcal{S})$ for $s_0$, followed by the policy $\pi$ and the dynamic $\mathcal{P}$. The $\gamma \in [0, 1)$ is the discounted factor that defines the importance of future rewards. The $\gamma$ close to 0 means that only short-term costs are considered so that old rewards will have a small impact; $\gamma$ close to 1 means that we focus on long-term rewards.

In practice, the policy space is very large. To reduce the dimensions and make the computation feasible, the policy $\pi$ is often parametrized as $\pi(\theta)$ with $\theta \in \Theta \subset \mathbb{R}^d$ belonged to certain family $\Theta$. So the function $V_\rho(\pi(\theta))$ depends on the parameter $\theta$ and we use the shorthand $V_\rho(\theta) \stackrel{\text{def}}{=} V_\rho(\pi(\theta))$. Now our goal switches to find the optimal parameter $\theta$ to maximize $V_\rho(\theta)$, which can be formulated as the following problem

$$\arg\max_{\theta\in\Theta} V_\rho(\theta) \stackrel{\text{def}}{=} \mathbb{E}_{s_0\sim\rho,\ a_t\sim\pi_{s_t}(\theta),\ s_{t+1}\sim\mathcal{P}(\cdot|s_t,a_t)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right].$$

Throughout the thesis, we consider $\Theta = \mathbb{R}^d$ in general without specification.

### 1.2.2 Policy gradient methods

Naturally, we can consider maximizing $V_\rho(\theta)$ as an optimization problem. So we can solve it with gradient ascent type methods, which is known as *Policy Gradient* (PG) method in RL. That is, at $k$-th iteration, we have

$$\theta^{k+1} \ = \ \theta^k \ + \ \eta^k \nabla_\theta V_\rho(\theta).$$

PG method is very popular in RL due to its simplicity. For instance, it is easier to implement and use in practice, compared to value-based or model-based methods, which are RL specific methods. PG method can solve a wide range of problems including non-Markov and partially-observable environments.

PG is popular also due to its versatility. First of all, PG has several forms of updates, such as REINFORCE (Williams, 1992), PGT (Sutton et al., 2000), GPOMDP (Baxter and Bartlett, 2001) and actor-critic (Konda and Tsitsiklis, 2000). It can be effectively paired with optimization techniques to obtain more sophisticated algorithms. For instance, natural policy gradient (Kakade, 2001, NPG) is a direct application of natural gradient method (Amari, 1998) from optimization to RL, and policy mirror descent (Lan, 2022; Xiao, 2022, PMD) is inspired from mirror descent (Nemirovski and Yudin, 1983; Beck and Teboulle, 2003) in optimization. Combined with the variance reduction techniques, such as SVRG (Johnson and Zhang, 2013), SARAH (Nguyen et al., 2017), SPIDER (Fang et al., 2018), SpiderBoost (Wang et al., 2019), STORM (Cutkosky and Orabona, 2019), SNVRG (Zhou et al., 2020), PAGE (Li et al., 2021b), and more (Tran-Dinh et al., 2021), lots of variance reduced PG methods in RL (Papini et al., 2018; Shen et al., 2019; Xu et al., 2020b; Yuan et al., 2020; Huang et al., 2020; Pham et al., 2020; Yang

et al., 2022; Huang et al., 2022) have been developed recently. In fact, the current state of the art algorithms in policy optimization TRPO (Schulman et al., 2015) and PPO (Schulman et al., 2017) are developed by leveraging specific structures of RL and the optimization techniques (e.g., trust-region and proximal method). Overall, variants of PG methods with optimization techniques were shown to have impressive empirical successes (Schulman et al., 2015; Lillicrap et al., 2016; Mnih et al., 2016; Schulman et al., 2017; Haarnoja et al., 2018), especially in the deep RL.

Despite the success of PG methods in practice, a solid theoretical understanding of even the "vanilla" PG has long been elusive until recent. However, the literature still remains fragmentary. Some of the literature focus on the analysis of the exact PG, including the pioneer work of Agarwal et al. (2021) and other works (Zhang et al., 2020a; Mei et al., 2020); some focus on the stochastic PG (Papini, 2020; Liu et al., 2020; Zhang et al., 2020b; Xiong et al., 2021). In terms of results, they are with different criteria of convergence, such as first-order stationary point convergence (Papini, 2020; Zhang et al., 2020b), global optimum convergence (Agarwal et al., 2021; Zhang et al., 2020a; Mei et al., 2020) and average regret to the global optimum (Zhang et al., 2021b; Liu et al., 2020). Different results are applied in different RL settings, such as the softmax tabular policy with or without different regularizations (Agarwal et al., 2021; Zhang et al., 2020a; Zhang et al., 2021b; Mei et al., 2020), or with different assumptions, such as the Lipschitz and smooth policy (Liu et al., 2020; Zhang et al., 2020b; Xiong et al., 2021) and the assumption of the bijection between the primal and the dual space (Zhang et al., 2020a). In particular, lots of the literature require large mini-batch of sampled trajectories, such as $\mathcal{O}(\epsilon^{-1})$ or $\mathcal{O}(\epsilon^{-2})$ trajectories per iteration for stochastic updates (Papini, 2020; Liu et al., 2020; Zhang et al., 2020b; Xiong et al., 2021). Here $\epsilon$ is the accuracy of the performance. This is strange, as in SGD convergence theory literature in optimization, single data per iteration is usually not an issue.

The second challenge about PG is that, unlike value-based or model-based methods, the existing PG methods, are not sample efficient in theory. Recently, NPG is proved by Xiao (2022) to be efficient for tabular case with linear convergence, which matches the convergence rate of value-based methods, such as policy iteration method (Puterman, 1994; Bertsekas, 2012). As mentioned, NPG (Kakade, 2001) inspires from natural gradient method (Amari, 1998), uses a preconditioner to improve PG direction, similar to quasi-Newton methods in classical optimization (Martens, 2020). So, can we extend the linear convergence of NPG from tabular to function approximation regime, which is a more realistic setting in practice ? Besides, NPG is important. It is the building block of TRPO (Schulman et al., 2015) and PPO (Schulman et al., 2017). Thus, it is of great interest to understand NPG well and to push its limits further. We address these two challenges of PG in the second part of the thesis, separately.

### 1.2.3 Outline and contributions of Part II

In the light of the above, this second part of the thesis is dedicated to deriving better theoretical understandings of the PG methods. We ask the following question: why PG methods are efficient and how to provably choose their hyper parameters ? By leveraging the RL structure of the problem together with modern optimization proof techniques (Khaled and Richtárik, 2023; Lan, 2022; Xiao, 2022), we derive novel finite time analysis of both the original PG and NPG in Chapter 4 and 5, respectively.

First, in Chapter 4 we adapt recent tools developed for the analysis of SGD in non-convex optimization from Khaled and Richtárik (2023) to obtain convergence and sample complexity guarantees for the original PG, including REINFORCE (Williams, 1992), PGT (Sutton et al., 2000) and GPOMDP (Baxter and Bartlett, 2001). Throughout the thesis, we will call the updates of REINFORCE, PGT and GPOMDP as vanilla PG. Our main contribution is to provide a general vanilla PG analysis with weaker assumptions compared to the literature. This general analysis not only unifies much of the fragmented results in the literature under one guise, but also recovers the best results for each setting, with a wider range of hyper parameter choices, which can be of great practical interest, and sometimes even improve the existing results with additional gradient domination assumption. More precisely, we provide a single convergence theorem that recovers the $\widetilde{\mathcal{O}}(\epsilon^{-4})$ sample complexity of vanilla PG to a stationary point. That is, consider a sample as a triple $(s_t, a_t, r(s_t, a_t))$ which is a single step interaction with the environment at time $t$ among a single sampled trajectory $\{s_{t'}, a_{t'}, r(s_{t'}, a_{t'})\}_{t' \geq 0}$ per iteration. With $\widetilde{\mathcal{O}}(\epsilon^{-4})$ samples, the vanilla PG is guaranteed to converge to an $\epsilon$-stationary point. Our results also affords greater flexibility in the choice of hyper parameters such as the step size and the batch size $m$ of the trajectories, including the single trajectory case (i.e., $m = 1$). When an additional *relaxed weak gradient domination* assumption is available, we establish a novel global optimum convergence theory of PG with $\widetilde{\mathcal{O}}(\epsilon^{-3})$ sample complexity. We then instantiate our theorems in different settings, where we both recover existing results and obtain improved sample complexity, e.g., $\widetilde{\mathcal{O}}(\epsilon^{-3})$ sample complexity for the convergence to the global optimum for Fisher-non-degenerated parametrized policies. The key ingredient of the analysis is to consider the so-called ABC assumption (Khaled and Richtárik, 2023), bounding the empirical gradient in terms of the suboptimality gap (A), the expected but truncated gradient (B) and an additive constant (C). This ABC assumption may appear a bit obscure at a first sight, but it is indeed a clever way to unify many of the current assumptions used in the RL literature. Notably, expected Lipschitz and smooth policies, softmax tabular policies with or without regularization, and Fisher non-degenerate policies are considered as special cases of the ABC assumption. Through our general analysis, we thus derive a better theoretical understanding of the vanilla PG and get a freedom to choose the hyper parameters of PG in practice according to the available computational resources.

As mentioned in the previous section, the vanilla PG is not sample efficient. In Chapter 5, we develop the linear convergence of another popular RL algorithm known as NPG and its variant, Q-NPG, for the class of log-linear policies. The resulting theorems extend the work of Xiao (2022) from the tabular softmax policies to the function approximation regime. We show that using a geometrically increasing step size, these algorithms can achieve a linear convergence rate, similar as the tabular setting, up to the function approximation error. The core of the analysis is that, leveraging the compatible function approximation framework developed by Agarwal et al. (2021), NPG can be interpreted as a mirror ascent approach. This is the viewpoint adopted for the analysis, where an inexact mirror ascent update is considered. The chapter further provides $\widetilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity results under some addition technical assumptions, which improve over best known results in the literature.

Throughout Part II, we derive a better understanding and sample efficiency in PG methods in RL.

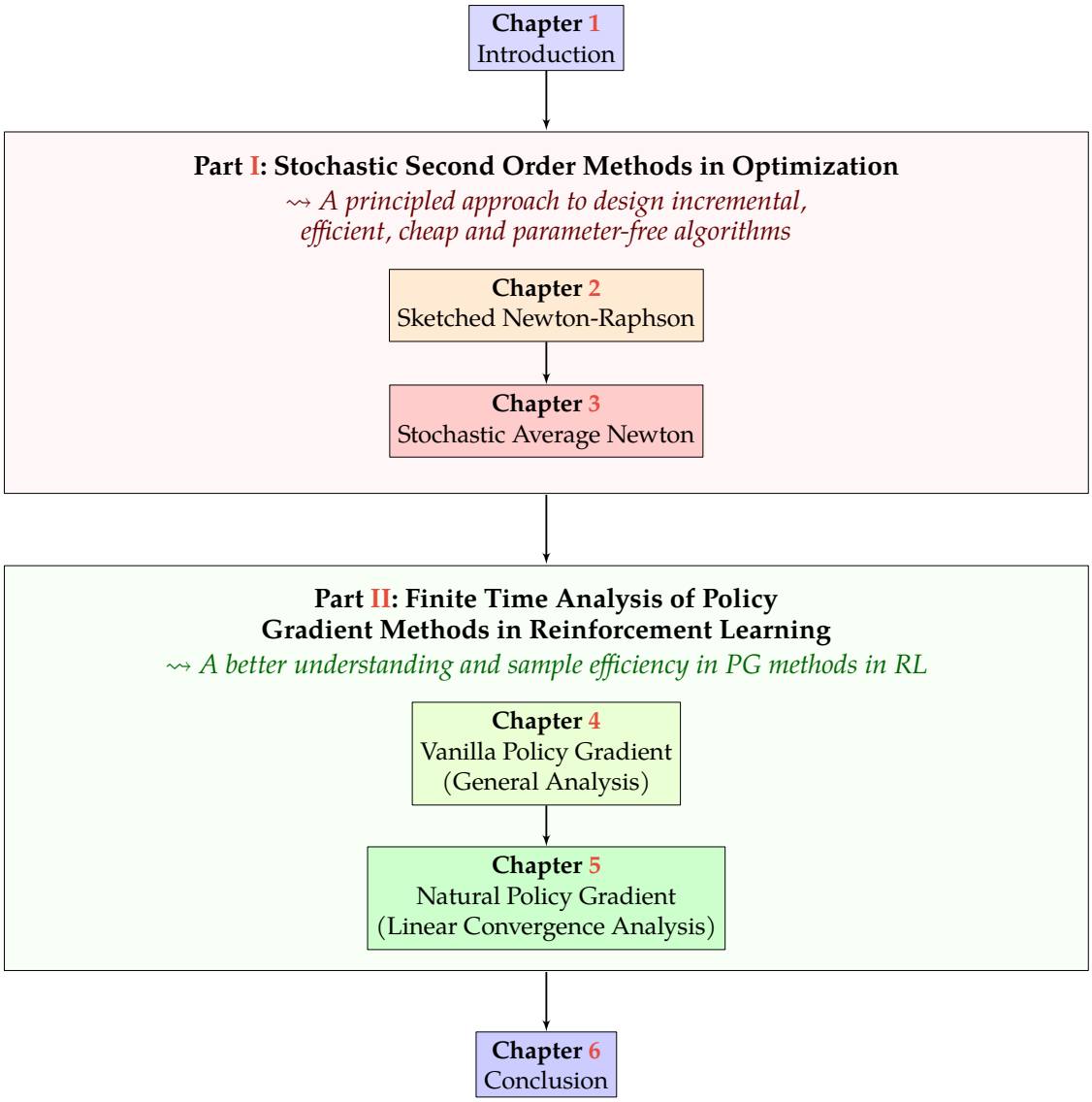


**Figure 1.4** – This thesis is separated in two parts. We start with optimization in Part I where we design new efficient stochastic second order methods with convergence guarantees. Leveraging the optimization proof techniques, we then move to reinforcement learning (RL) in Part II that focuses on the theoretical foundations of the policy gradient (PG) methods, including both the vanilla and natural policy gradient. These two topics are presented as being orthogonal, but there is a common thread of being optimization.

## List of publications

### Publications in international conferences with proceedings

- Rui Yuan, Alessandro Lazaric, Robert M. Gower. **Sketched Newton-Raphson**. In *Society for Industrial and Applied Mathematics* (*SIAM*) *Journal on Optimization* (SIOPT), 2022 (presented in Chapter 2)
- Jiabin Chen *, Rui Yuan *, Guillaume Garrigos, Robert M. Gower. **SAN: Stochastic Average Newton Algorithm for Minimizing Finite Sums**. In *International Conference on Artificial Intelligence and Statistics* (AISTATS), 2022 (presented in Chapter 3)
- Rui Yuan, Robert M. Gower, Alessandro Lazaric. **A general sample complexity analysis of vanilla policy gradient**. In *International Conference on Artificial Intelligence and Statistics* (AISTATS), 2022 (presented in Chapter 4)
- Rui Yuan, Simon S. Du, Robert M. Gower, Alessandro Lazaric, Lin Xiao. **Linear Convergence of Natural Policy Gradient Methods with Log-Linear Policies**. In *International Conference on Learning Representations* (ICLR), 2023 (presented in Chapter 5)

### Publication discussed in this thesis

- Carlo Alfano, Rui Yuan, Patrick Rebeschini. **A Novel Framework for Policy Mirror Descent with General Parametrization and Linear Convergence**. Preprint, 2023 (discussed in Chapter 5).

*denotes equal contribution.

# Part I

# Stochastic Second Order Methods in Optimization

# Chapter 2

# Sketched Newton-Raphson

In this chapter, we propose a new globally convergent stochastic second order method. Our starting point is the development of a new Sketched Newton-Raphson (SNR) method for solving large scale nonlinear equations of the form $F(x) = 0$ with $F : \mathbb{R}^p \to \mathbb{R}^m$. We then show how to design several stochastic second order optimization methods by re-writing the optimization problem of interest as a system of nonlinear equations and applying SNR. For instance, by applying SNR to find a stationary point of a generalized linear model (GLM), we derive completely new and scalable stochastic second order methods. We show that the resulting method is very competitive as compared to state-of-the-art variance reduced methods. Furthermore, using a variable splitting trick, we also show that the *Stochastic Newton method* (SNM) is a special case of SNR, and use this connection to establish the first global convergence theory of SNM.

We establish the global convergence of SNR by showing that it is a variant of the online stochastic gradient descent (SGD) method, and then leveraging proof techniques of SGD. As a special case, our theory also provides a new global convergence theory for the original Newton-Raphson method under strictly weaker assumptions as compared to the classic monotone convergence theory. [1]

## Contents



[1] This chapter is based on an article published in Society for Industrial and Applied Mathematics (SIAM) Journal on Optimization (SIOPT 2022) (Yuan et al., 2022b).

## 2.1 Introduction

One of the fundamental problems in numerical computing is to find roots of systems of nonlinear equations such as

$$F(x) = 0, \tag{2.1}$$

where $F : \mathbb{R}^p \to \mathbb{R}^m$. We assume throughout the chapter that $F : \mathbb{R}^p \to \mathbb{R}^m$ is continuously differentiable and that there exists a solution to (2.1), that is

**Assumption 2.1.** *$\exists x^* \in \mathbb{R}^p$ such that $F(x^*) = 0$.*

This includes a wide range of applications from solving the phase retrieval problems (Candès et al., 2015), systems of polynomial equations related to cryptographic primitives (Björklund et al., 2019), discretized integral and differential equations (Ortega and Rheinboldt, 2000), the optimal power flow problem (Torres and Quintana, 2000) and, our main interest here, solving nonlinear minimization problems in machine learning. Most convex optimization problems such as those arising from training a Generalized Linear Model (GLM), can be re-written as a system of nonlinear equations (2.1) either by manipulating the stationarity conditions or as the Karush-Kuhn-Tucker equations[2] (Karush, 1939; Kuhn and Tucker, 1951).

When dealing with non-convex optimization problems, such as training a Deep Neural Network (DNN), finding the global minimum is often infeasible (or not needed (Kawaguchi, 2016)). Instead, the objective is to find a good stationary point $x$ such that $\nabla f(x) = 0$, where $f$ is the total loss we want to minimize.

In particular, the task of training an overparametrized DNN (as they often are) can be cast as solving a special nonlinear system. That is, when the DNN is sufficiently overparametrized, the DNN can interpolate the data. As a consequence, if $f_i(x)$ is the loss function over the $i$th data point, then there is a solution to the system of nonlinear equations $\|\nabla f_i(x)\|^2 = 0, \forall i$.

The building block of many iterative methods for solving nonlinear equations is the Newton-Raphson (NR) method given by

$$x^{k+1} = x^k - \gamma \left(DF(x^k)^\top\right)^\dagger F(x^k) \tag{2.2}$$

at $k$th iteration, where $DF(x) \overset{\text{def}}{=} [\nabla F_1(x) \ \cdots \ \nabla F_m(x)] \in \mathbb{R}^{p\times m}$ is the transpose of the Jacobian matrix of $F$ at $x$, $\left(DF(x^k)^\top\right)^\dagger$ is the Moore-Penrose pseudoinverse of $DF(x^k)^\top$ (Moore, 1920; Bjerhammar, 1951; Penrose, 1955) and $\gamma > 0$ is the stepsize.

[2] Under suitable constraint qualifications (Nocedal and Wright, 1999).

The NR method is at the heart of many commercial solvers for nonlinear equations (Ortega and Rheinboldt, 2000). The success of NR can be partially explained by its invariance to affine coordinate transformations, which in turn means that the user does not need to tune any parameters (standard NR sets $\gamma = 1$). The downside of NR is that we need to solve a linear least squares problem given in (2.2) which costs $\mathcal{O}(\min\{pm^2, mp^2\})$ when using a direct solver. When both $p$ and $m$ are large, this cost per iteration is prohibitive. Here we develop a randomized NR method based on the sketch-and-project technique (Pilanci and Wainwright, 2015; Gower and Richtárik, 2015b) which can be applied in large scale, as we show in our experiments.

### 2.1.1 The sketched Newton-Raphson method

Our method relies on using *sketching matrices* to reduce the dimension of the Newton system.

**Definition 2.2.** *The sketching matrix $\mathbf{S} \in \mathbb{R}^{m\times\tau}$ is a random matrix sampled from a distribution $\mathcal{D}$, where $\tau \in \mathbb{N}$ is the sketch size. We use $\mathbf{S}_k \in \mathbb{R}^{m\times\tau}$ to denote a sketching matrix sampled from a distribution $\mathcal{D}_{x^k}$ that can depend on the iterate $x^k$.*

By sampling a sketching matrix $\mathbf{S}_k \sim \mathcal{D}_{x^k}$ at $k$th iteration, we *sketch* (row compress) NR update and compute an approximate *Sketched Newton-Raphson* (SNR) step, see line 4 in Algorithm 1. We use $\mathcal{D}_x$ to denote a distribution that depends on $x$, and allow the distribution of the sketching matrix to change from one iteration to the next.

**Algorithm 1:** SNR: Sketched Newton-Raphson

**Input:** $\mathcal{D}$ = distribution of sketching matrix; stepsize parameter $\gamma > 0$

1 Initialize $x^0 \in \mathbb{R}^p$

2 **for** $k = 0, 1, \cdots$ **do**

3 Sample a fresh sketching matrix: $\mathbf{S}_k \sim \mathcal{D}_{x^k}$

4 $x^{k+1} = x^k - \gamma DF(x^k)\mathbf{S}_k \left(\mathbf{S}_k^\top DF(x^k)^\top DF(x^k)\mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top F(x^k)$

**Output:** Last iterate $x^k$

Because the sketching matrix $\mathbf{S}_k$ has $\tau$ columns, the dominating costs of computing the SNR step (line 4 in Algorithm 1) are linear in $p$ and $m$. In particular, $DF(x^k)\mathbf{S}_k \in \mathbb{R}^{p\times\tau}$ can be computed by using $\tau$ directional derivatives of $F(x^k)$, one for each column of $\mathbf{S}_k$. Using automatic differentiation (Christianson, 1992), these directional derivatives cost $\tau$ evaluations of the function $F(x)$. Furthermore, it costs $\mathcal{O}(p\tau^2)$ to form the linear system in line 4 of Algorithm 1 by using the computed matrix $DF(x^k)\mathbf{S}_k$ and $\mathcal{O}(\tau^3)$ to solve it , respectively. Finally the matrix vector product $\mathbf{S}_k^\top F(x^k)$ costs $\mathcal{O}(m\tau)$. Thus, without making any further assumptions to the

structure of $F$ or the sketching matrix, the total cost in terms of operations of the update SNR (line 4 in Algorithm 1) is given by

$$\text{Cost(SNR)} = \mathcal{O}\left((\text{eval}(F) + m) \times \tau + p\tau^2 + \tau^3\right). \tag{2.3}$$

Thus Algorithm 1 can be applied when both $p$ and $m$ are large and $\tau$ is relatively small.

The rest of the chapter is organized as follows. In the next section, we provide some background and contrast it with our contributions. After introducing some notations in Section 2.1.3 and presenting alternative sketching techniques in Section 2.1.4, we show that SNR can be viewed as a sketch-and-project type method in Section 2.2. This is the viewpoint that first motivated the development of this method. After which, we provide another crucial equivalent viewpoint of SNR in Section 2.3, where we show that SNR can be seen as *Stochastic Gradient Descent* (SGD) applied to an equivalent reformulation of (2.1). We then provide a global convergence theory by leveraging this insight in Section 2.4. As a special case, our theory also provides a new global convergence theory for the original NR method (2.2) under strictly weaker assumptions as compared to the monotone convergence theory in Section 2.5, albeit for different step sizes. For the other extreme where the sketching matrix samples a single row, we present the new nonlinear Kaczmarz method as a variant of SNR and its global convergence theory in Section 2.6. We then show how to design several stochastic second order optimization methods by re-writing the optimization problem of interest as a system of nonlinear equations and applying SNR. For instance, using a variable splitting trick, we show that the *Stochastic Newton method* (SNM) (Rodomanov and Kropotov, 2016; Kovalev et al., 2019) is a special case of SNR, and use this connection to establish the first global convergence theory of SNM in Section 2.7. In Section 2.8, by applying SNR to find a stationary point of a GLM, we derive completely new and scalable stochastic second order methods. We show that the resulting method is very competitive as compared to state-of-the-art variance reduced methods.

### 2.1.2 Background and contributions

**a) Stochastic second-order methods.** There is now a concerted effort to develop efficient second-order methods for solving high dimensional and stochastic optimization problems in machine learning. Most recently developed Newton methods fall into one of two categories: *subsampling* and *dimension reduction*. The subsampling methods (Erdogdu and Montanari, 2015; Roosta-Khorasani and Mahoney, 2019; Kohler and Lucchi, 2017; Bollapragada et al., 2018; Zhou et al., 2018), and (Agarwal et al., 2017; Pilanci and Wainwright, 2017)[3] use mini-batches to

[3]Newton sketch (Pilanci and Wainwright, 2017) and LiSSa (Agarwal et al., 2017) use subsampling to build an estimate of the Hessian but require a full gradient evaluation. As such, these methods are not efficient for very large $n$.

compute an approximate Newton direction. Though these methods can handle a large number of *data points* ($n$), they do not scale well in the number of *features* ($d$). On the other hand, second-order methods based on dimension reduction techniques such as Gower et al. (2019a) apply Newton's method over a subspace of the features, and as such, do not scale well in the number of data points. Sketching has also been used to develop second-order methods in the online learning setting (Gürbüzbalaban et al., 2015; Luo et al., 2016; Calandriello et al., 2017) and quasi-Newton methods (Gower et al., 2016).

*Contributions.* We propose a new family of stochastic second-order method called SNR. Each choice of the sketching distribution and nonlinear equations used to describe the stationarity conditions, leads to a particular algorithm. For instance, we show that a nonlinear variant of the Kaczmarz method is a special case of SNR. We also show that the subsampling based SNM (Rodomanov and Kropotov, 2016; Kovalev et al., 2019) is a special case of SNR. By using a different norm in the sketch-and-project viewpoint, we show that the dimension reduced method *Randomized Subspace Newton* (RSN) (Gower et al., 2019a) is also a special case of SNR. We provide a concise global convergence theory, that when specialized to SNM gives its first global convergence result. Furthermore, the convergence theory of SNR allows for any sketch size, which translates to any mini-batch size for the nonlinear Kaczmarz and SNM. In contrast, excluding SNM, the subsampled based Newton methods (Erdogdu and Montanari, 2015; Roosta-Khorasani and Mahoney, 2019; Kohler and Lucchi, 2017; Bollapragada et al., 2018; Zhou et al., 2018; Agarwal et al., 2017; Pilanci and Wainwright, 2017) rely on high probability bounds that in turn require large mini-batch sizes[4]. We detail the nonlinear Kaczmarz method in Section 2.6, the connection with SNM in Section 2.7 and RSN in Appendix A.7.

**b) New method for GLMs.** There exist several specialized methods for solving GLMs, including variance reduced gradient methods such as SAG/SAGA (Schmidt et al., 2017; Defazio et al., 2014) and SVRG (Johnson and Zhang, 2013), and methods based on dual coordinate ascent like SDCA (Shalev-Shwartz and Zhang, 2013), dual free SDCA (dfSDCA) (Shalev-Shwartz, 2016) and Quartz (Qu et al., 2015).

*Contributions.* We develop a specialized variant of SNR for GLMs in Section 2.8. Our resulting method scales linearly in the number of dimensions $d$ and the number of data points $n$, has the same cost as SGD per iteration in average. We show in experiments that our method is very competitive as compared to state-of-the-art variance reduced methods for GLMs.

**c) Viewpoints of (Sketched) Newton-Raphson.** We show in Section 2.3 that SNR can be seen as SGD applied to an equivalent reformulation of our original problem. We will show that this reformulation is *always* a smooth and interpolated function (Ma et al., 2018; Vaswani

[4]The batch sizes in these methods scale proportional to a condition number (Agarwal et al., 2017) or $\epsilon^{-1}$ where $\epsilon$ is the desired tolerance.

et al., 2019a). These gratuitous properties allow us to establish a simple global convergence theory by only assuming that the reformulation is a *star-convex* function: a class of nonconvex functions that include convexity as a special case (Nesterov and Polyak, 2006; Lee and Valiant, 2016; Zhou et al., 2019; Hinder et al., 2020). The details of the SGD interpretation can be found in Section 2.3. In addition, we also show in Appendix A.1 that SNR can be seen as a type of stochastic Gauss-Newton method or as a type of stochastic fixed point method.

**d) Classic convergence theory of Newton-Raphson.** The better known convergence theorems for NR (the Newton-Kantorovich-Mysovskikh Theorems) only guarantee local or semi-local convergence (Kantorovitch, 1939; Ortega and Rheinboldt, 2000). To guarantee global convergence of NR, we often need an additional globalization strategy, such as damping sequences or adaptive trust-region methods (Conn et al., 2000; Lu et al., 2010; Deuflhard, 2011; Kelley, 2018), continuation schemes such as interior point methods (Nesterov and Nemirovskii, 1994; Wright and Nocedal, 2006), and more recently cubic regularization (Kovalev et al., 2019; Nesterov and Polyak, 2006; Cartis et al., 2009). Globalization strategies are used in conjunction with other second-order methods, such as inexact Newton backtracking type methods (Bellavia and Morini, 2001; An and Bai, 2007), Gauss-Newton or Levenberg-Marquardt type methods (Zhou and Chen, 2010; Zhou, 2013; Yuan, 2011) and quasi-Newton methods (Yuan, 2011)[5]. The only global convergence theory that does not rely on such a globalization strategy, requires strong assumptions on $F(x)$, such as in the monotone convergence theory (MCT) (Deuflhard, 2011).

*Contributions.* We show in Section 2.5.3 that our main theorem specialized to the standard NR method guarantees a global convergence under *strictly* less assumptions as compared to the MCT, albeit under a different stepsize. Indeed, MCT holds for step size equal to one ($\gamma = 1$) and our theory holds for step sizes less than one ($\gamma < 1$).

Furthermore, we give an explicit sublinear $\mathcal{O}(1/k)$ convergence rate, as opposed to only an asymptotic convergence in MCT. This appears to not be known before since, as stated by (Deuflhard, 2011) w.r.t. the NR method "*Not even an a-priori estimation for the number of iterations needed to achieve a prescribed accuracy* **may** *be possible*". We show that it is possible by monitoring which iterate achieves the best loss (suboptimality).

**e) Sketch-and-project.** The sketch-and-project method was originally introduced for solving linear systems in Gower and Richtárik (2015b) and Gower and Richtárik (2015a), where it was also proven to converge linearly and globally. In Richtárik and Takáč (2020), the authors then go on to show that the sketch-and-project method is in fact SGD applied to a particular reformulation of the linear system.

[5] A recent paper (Gao and Goldfarb, 2019) shows that quasi-Newton converges globally for self-concordant functions without globalization strategy.

*Contributions.* It is this SGD viewpoint in the linear setting (Richtárik and Takáč, 2020) that we extend to the nonlinear setting. Thus the SNR algorithm and our theory are generalizations of the original sketch-and-project method for solving linear equations to solving nonlinear equations, thus greatly expanding the scope of applications of these techniques.

### 2.1.3 Notations

In calculating an update of SNR and analyzing SNR, the following random matrix is key

$$\mathbf{H}_{\mathbf{S}}(x) \overset{\text{def}}{=} \mathbf{S}\left(\mathbf{S}^\top DF(x)^\top DF(x)\mathbf{S}\right)^\dagger \mathbf{S}^\top. \tag{2.4}$$

The sketching matrix $\mathbf{S}$ in (2.4) is sampled from a distribution $\mathcal{D}_x$ and $\mathbf{H}_{\mathbf{S}}(x) \in \mathbb{R}^{m\times m}$ is a random matrix that depends on $x$. We use $\mathbf{I}_p \in \mathbb{R}^{p\times p}$ to denote the identity matrix of dimension $p$ and use $\|x\|_{\mathbf{M}} \overset{\text{def}}{=} \sqrt{x^\top \mathbf{M} x}$ to denote the seminorm of $x \in \mathbb{R}^p$ induced by a symmetric positive semi-definite matrix $\mathbf{M} \in \mathbb{R}^{p\times p}$. Notice that $\|x\|_{\mathbf{M}}$ is not necessarily a norm as $\mathbf{M}$ is allowed to be non invertible. We handle this with care in our forthcoming analysis. We also define the following sets: $F(U) = \{F(x) \mid x \in U\}$ for a given set $U \subset \mathbb{R}^p$; $W^\perp = \{v \mid \langle u, v\rangle = 0,\ \forall u \in W\}$ to denote the orthogonal complement of a subspace $W$; $\mathbf{Im}(\mathbf{M}) = \{y \in \mathbb{R}^m \mid \exists x \in \mathbb{R}^p \text{ s.t. } \mathbf{M}x = y\}$ to denote the image space and $\mathbf{Ker}(\mathbf{M}) = \{x \in \mathbb{R}^p \mid \mathbf{M}x = 0\}$ to denote the null space of a matrix $\mathbf{M} \in \mathbb{R}^{m\times p}$. If $\mathbf{M}$ is a random matrix sampled from a certain distribution $\mathcal{D}$, we use $\mathbb{E}_{\mathbf{S}\sim\mathcal{D}}[\mathbf{M}] = \int_{\mathbf{M}} \mathbf{M} d\mathbb{P}_{\mathcal{D}}(\mathbf{M})$ to denote the expectation of the random matrix. We omit the notation of the distribution $\mathcal{D}$, i.e. $\mathbb{E}[\mathbf{M}]$, when the random source is clear. In particular, when $\mathbf{M}$ is sampled from a discrete distribution with $r \in \mathbb{N}$ s.t. $\Pr[\mathbf{M} = \mathbf{M}_i] = p_i > 0$, for $i = 1, \cdots, r$ and $\sum_{i=1}^r p_i = 1$, then $\mathbb{E}[\mathbf{M}] = \sum_{i=1}^r p_i \mathbf{M}_i$.

### 2.1.4 Sketching matrices

Here we provide examples of sketching matrices that can be used in conjunction with SNR. We point the reader to Woodruff (2014) for a detailed exposure and introduction. The most straightforward sketch is given by the Gaussian sketch where every coordinate $\mathbf{S}_{ij}$ of the sketch $\mathbf{S} \in \mathbb{R}^{m\times\tau}$ is sampled i.i.d according to a Gaussian distribution with $\mathbf{S}_{ij} \sim \mathcal{N}(0, \frac{1}{\tau})$ for $i = 1, \cdots, m$ and $j = 1, \cdots, \tau$. The sketch we mostly use here is the uniform subsampling sketch, whereby

$$\Pr[\mathbf{S} = \mathbf{I}_C] = \frac{1}{\binom{m}{\tau}}, \quad \text{for all set } C \subset \{1, \cdots, m\} \text{ s.t. } |C| = \tau, \tag{2.5}$$

where $\mathbf{I}_C \in \mathbb{R}^{m\times\tau}$ denotes the concatenation of the columns of the identity matrix $\mathbf{I}_m$ indexed in the set $C$. More sophisticated sketches that are able to make use of fast Fourier type routines include the random orthogonal sketches (ROS) (Pilanci and Wainwright, 2017; Ailon and

Chazelle, 2009). We will not cover ROS sketches here since these sketches are fast when applied only once to a fixed matrix $\mathbf{M}$, as opposed to being re-sampled at every iteration .

## 2.2 The sketch-and-project viewpoint

The viewpoint that motivated the development of Algorithm 1 was the following iterative *sketch-and-project* method applied to the Newton system. For this viewpoint, we assume that

**Assumption 2.3.** $F(x) \in \mathbf{Im}\left(DF(x)^\top\right) \forall x \in \mathbb{R}^p$.

This assumption guarantees that there exists a solution to the Newton system in (2.2). Indeed, we can now re-write the NR method (2.2) as a projection of the previous iterate $x^k$ onto the solution space of a Newton system

$$x^{k+1} = \operatorname{argmin}_{x\in\mathbb{R}^p} \left\|x - x^k\right\|^2 \quad \text{s. t.} \quad DF(x^k)^\top(x - x^k) = -\gamma F(x^k). \tag{2.6}$$

Since this is costly to solve when $DF(x^k)$ has many rows and columns, we *sketch* the Newton system. That is, we apply a random row compression to the Newton system using the sketching matrix $\mathbf{S}_k^\top \in \mathbb{R}^{\tau\times m}$ and then project the previous iterates $x^k$ onto this *sketched* system as follows

$$x^{k+1} = \operatorname{argmin}_{x\in\mathbb{R}^p} \left\|x - x^k\right\|^2 \quad \text{s. t.} \quad \mathbf{S}_k^\top DF(x^k)^\top(x - x^k) = -\gamma \mathbf{S}_k^\top F(x^k). \tag{2.7}$$

That is, $x^{k+1}$ is the projection of $x^k$ onto the solution space of the sketched Newton system. This viewpoint was our motivation for developing the SNR method. Next we establish our core theory. The theory does not rely on the assumption $F(x) \in \mathbf{Im}\left(DF(x)^\top\right)$, though this assumption will appear again in several specialized corollaries. Without this assumption, we can still interpret the Newton step (2.2) as the least squares solution of the linear system (2.6), as we show next.

## 2.3 Reformulation as stochastic gradient descent

Our insight into interpreting and analyzing the SNR in Algorithm 1 is through its connection to the SGD. Next, we show how SNR can be seen as SGD applied to a sequence of equivalent reformulations of (2.1). Each reformulation is given by a vector $y \in \mathbb{R}^p$ and the following minimization problem

$$\min_{x\in\mathbb{R}^p} \mathbb{E}_{\mathbf{S}\sim\mathcal{D}_y}\left[\tfrac{1}{2}\left\|F(x)\right\|^2_{\mathbf{H}_\mathbf{S}(y)}\right], \tag{2.8}$$

where $\mathbf{H}_{\mathbf{S}}(y)$ is defined in (2.4). To abbreviate notations, let

$$f_{\mathbf{S},y}(x) \overset{\text{def}}{=} \frac{1}{2}\|F(x)\|^2_{\mathbf{H}_{\mathbf{S}}(y)} \quad \text{and} \quad f_y(x) \overset{\text{def}}{=} \mathbb{E}\left[f_{\mathbf{S},y}(x)\right] = \frac{1}{2}\|F(x)\|^2_{\mathbb{E}[\mathbf{H}_{\mathbf{S}}(y)]}. \tag{2.9}$$

Every solution $x^* \in \mathbb{R}^p$ to (2.1) is a solution to (2.8), since $f_y(x)$ is non-negative for every $x \in \mathbb{R}^p$ and $f_y(x^*) = 0$ is thus a global minima. With an extra assumption, we can show that every solution to (2.8) is also a solution to (2.1) in the following lemma.

**Lemma 2.4.** *If Assumption 2.1 holds and the following* reformulation *assumption*

$$F(\mathbb{R}^p) \cap \mathbf{Ker}\left(\mathbb{E}_{\mathbf{S}\sim\mathcal{D}_y}\left[\mathbf{H}_{\mathbf{S}}(y)\right]\right) = \{0\}, \quad \forall y \in \mathbb{R}^p \tag{2.10}$$

*holds, then* $\operatorname{argmin}_{x\in\mathbb{R}^p} f_y(x) = \{x \mid F(x) = 0\}$ *for every* $y \in \mathbb{R}^p$.

*Proof.* Let $y \in \mathbb{R}^p$. Previously, we show that $\{x \mid F(x) = 0\} \subset \operatorname{argmin}_{x\in\mathbb{R}^p} f_y(x)$. Now let $x^* \in \operatorname{argmin}_{x\in\mathbb{R}^p} f_y(x)$. By Assumption 2.1, we know that any global minimizer $x^*$ of $f_y(x)$ must be s.t. $f_y(x^*) = 0$. This implies that $F(x^*) \in \mathbf{Ker}\left(\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(y)\right]\right)$ since $\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(y)\right]$ is symmetric. However $F(x^*) \in F(\mathbb{R}^p)$ and thus from (2.10), we have that $F(x^*) \in F(\mathbb{R}^p) \cap \mathbf{Ker}\left(\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(y)\right]\right) = \{0\}$, which implies $F(x^*) = 0$. Thus, we have $\operatorname{argmin}_{x\in\mathbb{R}^p} f_y(x) \subset \{x \mid F(x) = 0\}$ which concludes the proof. □

Thus with the extra reformulation assumption in (2.10), we can now use any viable optimization method to solve (2.8) for any fixed $y \in \mathbb{R}^p$ and arrive at a solution to (2.1). In Lemma A.8, we give sufficient conditions on the sketching matrix and on the function $F(x)$ that guarantee (2.10) hold. We also show how (2.10) holds for our forthcoming examples in Appendix A.6 as a direct consequence of Lemma A.8. However, (2.10) imposes for all $y \in \mathbb{R}^p$ which can be sometimes restrictive. In fact, we do *not need* for (2.8) to be equivalent to solving (2.1) for *every* $y \in \mathbb{R}^p$. Indeed, by carefully and iteratively updating $y$, we can solve (2.8) and obtain a solution to (2.1) *without* relying on (2.10). The trick here is to use an *online SGD* method for solving (2.8).

Since (2.8) is a stochastic optimization problem, SGD is a natural choice for solving (2.8). Let $\nabla f_{\mathbf{S},y}(x)$ denote the gradient of the function $f_{\mathbf{S},y}(\cdot)$ which is

$$\nabla f_{\mathbf{S},y}(x) \;=\; DF(x)\mathbf{H}_{\mathbf{S}}(y)F(x). \tag{2.11}$$

Since we are free to choose $y$, we allow $y$ to *change* from one iteration to the next by setting $y = x^k$ at the start of the $k$th iteration. We can now take a SGD step by sampling $\mathbf{S}_k \sim \mathcal{D}_{x^k}$ at

$k$th iteration and updating

$$x^{k+1} = x^k - \gamma \nabla f_{\mathbf{S}_k, x^k}(x^k). \tag{2.12}$$

It is straightforward to verify that the SGD update (2.12) is exactly the same as the SNR update in line 4 in Algorithm 1.

The objective function $f_{\mathbf{S},y}(x)$ has many properties that makes it very favourable for optimization including the interpolation condition and a gratuitous smoothness property. Indeed, for any $x^* \in \mathbb{R}^p$ s.t. $F(x^*) = 0$, we have that the stochastic gradient is zero, i.e. $\nabla f_{\mathbf{S},y}(x^*) = 0$. This is known as the *interpolation condition*. When it occurs together with strong convexity, it is possible to shows that SGD converges linearly (Ma et al., 2018; Vaswani et al., 2019a). We will also give a linear convergence result in Section 2.4 by assuming that $f_y(x)$ is quasi-strongly convex. We detail the smoothness property next.

However, we need to be careful, since (2.12) is not a classic SGD method. In fact, from the $k$th iteration to the $(k+1)$th iteration, we change our objective function from $f_{x^k}(x)$ to $f_{x^{k+1}}(x)$ and the distribution from $\mathcal{D}_{x^k}$ to $\mathcal{D}_{x^{k+1}}$. Thus it is an online SGD. We handle this with care in our forthcoming convergence proofs.

## 2.4 Convergence theory

Using the viewpoint of SNR in Section 2.3, we adapt proof techniques of SGD to establish the global convergence of SNR.

### 2.4.1 Smoothness property

In our upcoming proof, we rely on the following type of smoothness property thanks to our SGD reformulation (2.8).

**Lemma 2.5.** *For every $x \in \mathbb{R}^p$ and any realization $\mathbf{S} \sim \mathcal{D}_x$ associated with any distribution $\mathcal{D}_x$,*

$$\frac{1}{2}\|\nabla f_{\mathbf{S},x}(x)\|^2 = f_{\mathbf{S},x}(x). \tag{2.13}$$

This is not a standard smoothness property. Indeed, since $\nabla f_{\mathbf{S},x}(x^*) = 0$ and $f_x(x^*) = 0$, we have that (2.13) implies that

$$\|\nabla f_{\mathbf{S},x}(x) - \nabla f_{\mathbf{S},x}(x^*)\|^2 \leq 2(f_{\mathbf{S},x}(x) - f_{\mathbf{S},x}(x^*)),$$

which is usually a consequence of assuming that $f_{\mathbf{S},x}(x)$ is convex and 1–smooth (see Theorem 2.1.5 and Equation 2.1.7 in Nesterov (2014)). Yet in our case, equation (2.13) is a direct consequence of the definition of $f_{\mathbf{S},x}$ as opposed to being an extra assumption. This gratuitous property will be key in establishing a global convergence result.

### 2.4.2 Convergence for star-convex

We use the shorthand $f_k(x) \overset{\text{def}}{=} f_{x^k}(x)$, $f_{\mathbf{S}_k,k} \overset{\text{def}}{=} f_{\mathbf{S}_k,x^k}$ and $\mathbb{E}_k[\cdot] \overset{\text{def}}{=} \mathbb{E}\left[\cdot \mid x^k\right]$. Here we establish the global convergence of SNR by supposing that $f_k$ is *star-convex* which is a large class of nonconvex functions that includes convexity as a special case (Nesterov and Polyak, 2006; Lee and Valiant, 2016; Zhou et al., 2019; Hinder et al., 2020).

**Assumption 2.6** (Star-Convexity). *Let $x^*$ satisfy Assumption 2.1, i.e. let $x^*$ be a solution to (2.1). For every $x^k$ given by Algorithm 1 with $k \in \mathbb{N}$, we have that*

$$f_k(x^*) \;\geq\; f_k(x^k) + \left\langle \nabla f_k(x^k), x^* - x^k \right\rangle. \tag{2.14}$$

We now state our main theorem.

**Theorem 2.7.** *Let $x^*$ satisfy Assumption 2.6. If $0 < \gamma < 1$, then*

$$\mathbb{E}\left[\min_{t=0,\ldots,k-1} f_t(x^t)\right] \leq \frac{1}{k}\sum_{t=0}^{k-1} \mathbb{E}\left[f_t(x^t)\right] \leq \frac{1}{k}\frac{\left\|x^0 - x^*\right\|^2}{2\gamma\left(1-\gamma\right)}. \tag{2.15}$$

*Written in terms of $F$ and for $\gamma = 1/2$ the above gives*

$$\mathbb{E}\left[\min_{t=0,\ldots,k-1} \left\|F(x^t)\right\|^2_{\mathbb{E}[\mathbf{H}_{\mathbf{S}}(x^t)]}\right] \quad\leq\quad \frac{4\left\|x^0 - x^*\right\|^2}{k}.$$

*Besides, if the stochastic function $f_{\mathbf{S},x}(x)$ is star-convex along the iterates $x^k$ , i.e.*

$$f_{\mathbf{S}_k,x^k}(x^*) \;\geq\; f_{\mathbf{S}_k,x^k}(x^k) + \left\langle \nabla f_{\mathbf{S}_k,x^k}(x^k), x^* - x^k \right\rangle \tag{2.16}$$

*for all $\mathbf{S}_k \sim \mathcal{D}_{x^k}$, then the iterates $x^k$ of SNR (line 4 in Algorithm 1) are bounded with*

$$\left\|x^k - x^*\right\| \;\leq\; \left\|x^0 - x^*\right\|. \tag{2.17}$$

Theorem 2.7 is an unusual result for SGD methods. Currently, to get a $\mathcal{O}(1/k)$ convergence rate for SGD, one has to assume smoothness and strong convexity (Gower et al., 2019b) or convexity, smoothness and interpolation (Vaswani et al., 2019a). Here we get a $\mathcal{O}(1/k)$ rate by *only* assuming star-convexity. This is because we have smoothness and interpolation properties as a by-product due to our reformulation (2.8). However, the star-convexity assumption of $f_k(\cdot)$ for all $k \in \mathbb{N}$ is hard to interpret in terms of assumptions on $F$ in general. But, we are able to interpret it in many important extremes. That is, for the full NR method, we show that it suffices for the Newton direction to be 2–co-coercive (see (2.34) in Section 2.5). For the other extreme where the sketching matrix samples a single row, then the star-convexity assumption is even easier to check, and is guaranteed to hold so long as $F_i(x)^2$ is convex for all $i = 1, \cdots, m$ (see Section 2.6).

Next, we will show the convergence of $F(x^k)$ instead of $f_k(x^k)$ via Theorem 2.7.

#### 2.4.2.1 Sublinear convergence of the Euclidean norm $\|F\|$

If $\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x)\right]$ is invertible for all $x \in \mathbb{R}^p$, we can use Theorem 2.7 with the bound (2.17) to guarantee that $\|F\|$ converges sublinearly. Indeed, when $\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x)\right]$ is invertible, $\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x)\right]$ is symmetric positive definite. Thus there exists $\lambda > 0$ that bounds the smallest eigenvalue away from zero in any closed bounded set (e.g. $\{x \in \mathbb{R}^p \mid \|x - x^*\| \leq \|x^0 - x^*\|\}$[6]):

$$\min_{x \in \{x \mid \|x - x^*\| \leq \|x^0 - x^*\|\}} \lambda_{\min}\left(\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x)\right]\right) = \lambda > 0, \tag{2.18}$$

where $\lambda_{\min}(\cdot)$ is the smallest eigenvalue operator. Consequently, under the assumption of Theorem 2.7 with the condition (2.16), from (2.17) and (2.18), we have

$$\lambda \mathbb{E}\left[\min_{t=0,\ldots,k-1} \left\|F(x^t)\right\|^2\right] \leq \mathbb{E}\left[\min_{t=0,\ldots,k-1} \left\|F(x^t)\right\|^2_{\mathbb{E}[\mathbf{H}_{\mathbf{S}}(x^t)]}\right] \overset{(2.15)}{\leq} \frac{1}{k} \frac{\left\|x^0 - x^*\right\|^2}{\gamma\left(1 - \gamma\right)}. \tag{2.19}$$

It turns out that using the smallest eigenvalue of $\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x)\right]$ in the above bound is overly pessimistic. To improve it, first note that we do *not need* that $\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x)\right]$ is invertible. Instead, we only need that $F(x) \in \mathbf{Im}(DF(x)^\top) \subset \mathbf{Im}(\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x)\right])$, as we show in Corollary 2.8.

Note $L \overset{\text{def}}{=} \sup_{x \in \{x \mid \|x - x^*\| \leq \|x^0 - x^*\|\}} \|DF(x)\| > 0$. Such $L$ exists because $x$ is in a closed bounded convex set and because we have assumed that $DF(\cdot)$ is continuous. A continuous mapping over a closed bounded convex set is bounded. Now we can state the sublinear convergence results for $\|F\|$.

[6] We can re-write the set as the closure of the ball $\{x \in \mathbb{R}^p \mid x \in \overline{\mathcal{B}(x^*, \|x^0 - x^*\|)}\}$.

**Corollary 2.8.** *Let*

$$\rho(x) \overset{def}{=} \min_{v \in \mathbf{Im}(DF(x))/\{0\}} \frac{v^\top DF(x) \mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x)\right] DF(x)^\top v}{\|v\|^2}, \tag{2.20}$$

$$\rho \overset{def}{=} \min_{x \in \{x \mid \|x - x^*\| \leq \|x^0 - x^*\|\}} \rho(x). \tag{2.21}$$

*It follows that* $0 \leq \rho(x) \leq 1$. *If*

$$F(x) \in \mathbf{Im}(DF(x)^\top) \subset \mathbf{Im}(\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x)\right]) \quad \text{for all } x \in \mathbb{R}^p, \tag{2.22}$$

*then* $\rho(x) = \lambda_{\min}^+\left(DF(x)\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x)\right] DF(x)^\top\right) > 0 \quad \forall x \in \mathbb{R}^p$, *and* $\rho > 0$, *where* $\lambda_{\min}^+$ *is the smallest* non-zero *eigenvalue. Furthermore, if the star-convexity for each sketching matrix* (2.16) *holds , then*

$$\mathbb{E}\left[\min_{t=0,\ldots,k-1} \left\|F(x^t)\right\|^2\right] \leq \frac{1}{k} \cdot \frac{L^2 \left\|x^0 - x^*\right\|^2}{\rho\gamma\left(1-\gamma\right)}. \tag{2.23}$$

Thus with Corollary 2.8, we show that $F(x^t)$ converges to zero. This lemma relies on the inclusion (2.22), which in turn imposes some restrictions on the sketching matrix and $F(x)$. In our forthcoming examples in Section 2.5 and 2.6, we can directly verify the inclusion of (2.22). For other examples in Section 2.7 and 2.8, we provide the following Lemma 2.9 where we give sufficient conditions for (2.22) to hold.

**Lemma 2.9.** *Let* $F(x) \in \mathbf{Im}(DF(x)^\top)$. *Furthermore, we suppose that* $\mathbf{S} \sim \mathcal{D}_x$ *is* adapted *to* $DF(x)$ *by which we mean*

$$\mathbf{Ker}\left(\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right]\right) \subset \mathbf{Ker}(DF(x)) \subset \mathbf{Ker}\left(\mathbf{S}^\top\right), \quad \text{for all } \mathbf{S} \sim \mathcal{D}_x. \tag{2.24}$$

*Then it follows that* (2.22) *holds for all* $x \in \mathbb{R}^p$.

We refer to a sketching matrix $\mathbf{S} \sim \mathcal{D}_x$ that satisfies (2.24) as a sketch that is *adapted to* $DF(x)$. One easy way to design such adapted sketches is the following.

**Lemma 2.10.** *Let* $\hat{\mathbf{S}} \in \mathbb{R}^{p \times \tau}$ *s.t.* $\hat{\mathbf{S}} \sim \mathcal{D}$ *a fixed distribution independent to* $x$ *and* $\mathbf{Ker}(\mathbb{E}[\hat{\mathbf{S}}\hat{\mathbf{S}}^\top]) \subset \mathbf{Ker}(DF(x)^\top)$. *Thus,* $\mathbf{S} = DF(x)^\top \hat{\mathbf{S}} \in \mathbb{R}^{m \times \tau}$ *is adapted to* $DF(x)$.

The condition $\mathbf{Ker}(\mathbb{E}[\hat{\mathbf{S}}\hat{\mathbf{S}}^\top]) \subset \mathbf{Ker}(DF(x)^\top)$ in Lemma 2.10 holds for many standard sketches including Gaussian and subsampling sketches presented as follows.

**Lemma 2.11.** *For Gaussian and uniform subsampling sketches defined in Section 2.1.4, we have that* $\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right] = c\mathbf{I}_m$ *with* $c > 0$ *a fixed constant depending on the sketch.*

From Lemma 2.11, we know that $\mathbb{E}[\hat{\mathbf{S}}\hat{\mathbf{S}}^\top] = c\mathbf{I}_p$ invertible with $c > 0$. Thus $\mathbf{Ker}(\mathbb{E}[\hat{\mathbf{S}}\hat{\mathbf{S}}^\top]) = \{0\} \subset \mathbf{Ker}(DF(x)^\top)$ holds for *any* sketch size $\tau$.

### 2.4.3 Convergence for strongly convex

Here we establish a global linear convergence of SNR when assuming that $f_y$ is strongly quasi-convex.

**Assumption 2.12** ($\mu$-Strongly Quasi-Convexity)**.** *Let* $x^*$ *satisfy Assumption 2.1 and*

$$\exists\, \mu > 0 \text{ s.t. } f_y(x^*) \geq f_y(x) + \langle \nabla f_y(x), x^* - x\rangle + \frac{\mu}{2}\|x^* - x\|^2 \quad \forall\, x, y \in \mathbb{R}^p. \tag{2.25}$$

This Assumption 2.12 is strong, so much so, we have the following lemma.

**Lemma 2.13.** *Assumption 2.12 implies (2.10) and that the solution to (2.1) is unique.*

Under Assumption 2.12 , choosing $\gamma = 1$ guarantees a fast global linear convergence.

**Theorem 2.14.** *If* $x^*$ *satisfies Assumption 2.12 and* $\gamma \leq 1$*, then SNR converges linearly:*

$$\mathbb{E}\left[\left\|x^{k+1} - x^*\right\|^2\right] \leq (1 - \gamma\mu)^{k+1}\left\|x^0 - x^*\right\|^2 \qquad \text{with } \mu \leq 1. \tag{2.26}$$

## 2.5 New global convergence theory of the NR method

As a direct consequence of our general convergence theorems, in this section we develop a new global convergence theory for the original NR method. We first provide the results in 1-dimension in Section 2.5.1, then a general result in higher dimensions in the subsequent Section 2.5.2 and compare this result to the classic monotone convergence theory in Section 2.5.3.

### 2.5.1 A single nonlinear equation

Consider the case where $F(x) = \phi(x) \in \mathbb{R}$ is a one dimensional function and $x \in \mathbb{R}$. This includes common applications of the NR method such as calculating square roots of their reciprocal[7] and finding roots of polynomials. Even in this simple one dimension case, we find that our assumptions of global convergence given in Corollary 2.8 are *strictly weaker* than the standard assumptions used to guarantee NR convergence, as we explain next.

The NR method in one dimension at every iteration $k$ is given by

$$x^{k+1} = x^k - \frac{\phi(x^k)}{\phi'(x^k)} \stackrel{\text{def}}{=} g(x^k).$$

To guarantee that this is well defined, we assume that $\phi'(x^k) \neq 0$ for all $k$. A sufficient condition for this procedure to converge locally is that $|g'(x)| < 1$ with $x \in I$ where $I$ is a given interval containing the solution $x^*$. See for example Section 1.1 in (Deuflhard, 2011) or Chapter 12 in (Ortega and Rheinboldt, 2000). We can extend this to a global convergence by requiring that $|g'(x)| < 1$ globally. In the case of NR, since $g'(x) = 1 - \frac{\phi'(x)^2 - \phi(x)\phi''(x)}{\phi'(x)^2} = \frac{\phi(x)\phi''(x)}{\phi'(x)^2}$, this condition amounts to requiring

$$\frac{|\phi(x)\phi''(x)|}{\phi'(x)^2} < 1. \tag{2.27}$$

Curiously, this condition (2.27) has an interesting connection to convexity. In fact, condition (2.27) implies that $\phi^2(x)$ is convex and twice continuously differentiable. To see this, note that $\frac{d^2}{dx^2}\phi^2(x) \geq 0$ is equivalent to

$$\frac{d^2}{dx^2}\phi^2(x) = 2\frac{d}{dx}\phi'(x)\phi(x) = 2\left(\phi(x)\phi''(x) + \phi'(x)^2\right) \geq 0. \tag{2.28}$$

Now it is easy to see that (2.27) implies (2.28). Finally (2.28) also implies that $\phi^2(x)$ is *star-convex*, which is exactly what is required by our convergence theory in Corollary 2.8.

Indeed, in this one dimensional setting, Assumption 2.6 is equivalent to (2.16) and our reformulation in (2.8) boils down to minimizing $f_y(x) = (\phi(x)/\phi'(y))^2$. Thus by Corollary 2.8, the NR method converges globally if $f_{x^k}(x)$, or simply if $\phi(x)^2$ is star-convex and $\phi'(x^k) \neq 0$ for all iterates of $NR$, which shows that our condition is strictly weaker than the other conditions, because there exists functions that are star-convex but not convex, e.g. $\phi(x)^2 = |x|(1-\exp(-|x|))$ from Nesterov and Polyak (2006) and Lee and Valiant (2016).

[7] Used in particular to compute angles of incidence and reflection in games such as quake https://en.wikipedia.org/wiki/Fast_inverse_square_root.

For future reference and convenience, we can re-write the star-convexity of each $\phi(x)^2$ as

$$0 \;=\; \phi(x^*)^2 \;\geq\; \phi^2(x) + 2\phi(x)\phi'(x)(x^* - x),$$

where $x^*$ is the global minimum of $\phi(x)^2$, i.e. $\phi(x^*) = 0$. This can be re-written as

$$0 \;\geq\; \phi(x)\left(\phi(x) + 2\phi'(x)(x^* - x)\right). \tag{2.29}$$

By verifying (2.29) and that $\phi'(x^k) \neq 0$ on the iterates of NR, we can guarantee that the method converges globally.

### 2.5.2 The full NR

Now let $F(x) \in \mathbb{R}^m$ and consider the full NR method (2.2). Similarly, since $\mathbf{S} = \mathbf{I}_m$, Assumption 2.6 is equivalent to (2.16). Corollary 2.8 sheds some new light on the convergence of NR. In this case, our reformulation (2.8) is given by

$$f_y(x) \;=\; \tfrac{1}{2}F(x)^\top (DF(y)^\top DF(y))^\dagger F(x) \;=\; \tfrac{1}{2}\left\| \left(DF(y)^\top\right)^\dagger F(x)\right\|^2 \tag{2.30}$$

and Corollary 2.8 states that NR converges if $f_{x^k}(x)$ is star-convex for all the iterates $x^k \in \mathbb{R}^p$. This has a curious re-interpretation in this setting. Indeed, let

$$n(x) \;\overset{\text{def}}{=}\; -(DF(x)^\top)^\dagger F(x) \tag{2.31}$$

be the Newton direction. From (2.30) and (2.31), we have that

$$f_x(x) \;=\; \frac{1}{2}\left\|n(x)\right\|^2. \tag{2.32}$$

Using (2.32), Corollary 2.8 can be stated in this special case as the following corollary.

**Corollary 2.15.** *Consider $x^k$ given by the NR (2.2) with $\gamma < 1$. If we have*

$$F(x) \in \mathbf{Im}(DF(x)^\top), \tag{2.33}$$

$$\frac{1}{2}\left\|n(x)\right\|^2 \leq \langle n(x), x^* - x\rangle \tag{2.34}$$

*hold for every $x = x^k$ with solution $x^*$, then it exists $L > 0$ s.t. $\|DF(x^k)\| \leq L$ and*

$$\min_{t=0,\ldots,k-1} \left\|F(x^t)\right\|^2 \;\leq\; \frac{1}{k}\cdot\frac{L^2\left\|x^0 - x^*\right\|^2}{\gamma\,(1-\gamma)}. \tag{2.35}$$

First, in our proof of Corollary 2.15 in Appendix A.3.1, condition (2.33) implies (2.22).

Furthermore, condition (2.34) can be seen as a *co-coercivity* property of the Newton direction. This co-coercivity establishes a curious link with the modern proofs of convergence of gradient descent which rely on the co-coercivity of the gradient direction. That is, if $f(x)$ is convex and $L$–smooth, then we have that the gradient is $L$–co-coercive with

$$\frac{1}{L}\left\|\nabla f(x)\right\|^2 \;\leq\; \langle \nabla f(x), x - x^* \rangle \,.$$

This is the key property for proving convergence of gradient descent, see e.g. Section 5.2.4 in Bach (2021). To the best of our knowledge, this is the first time that the co-coercivity of the Newton direction has been identified as a key property for proving convergence of the Newton's method. In particular, global convergence results for the NR method such as the monotone convergence theories (MCT) only hold for functions $F : \mathbb{R}^p \to \mathbb{R}^m$ with $p = m$ and rely on a stepsize $\gamma = 1$, see Ortega and Rheinboldt (2000) and Deuflhard (2011). Corollary 2.15 accommodates "non-square" functions $F : \mathbb{R}^p \to \mathbb{R}^m$. Excluding the difference in stepsizes and focusing on "square" functions $F : \mathbb{R}^p \to \mathbb{R}^m$ with $p = m$, next we show in Theorem 2.16 that our assumptions are strictly weaker than those used for establishing the global convergence of NRwith constant stepsizes through the MCT.

### 2.5.3 Comparing to the classic monotone convergence theory of NR

Consider $m = p$. Here we show that our Assumption 2.1, (2.33) and (2.34) are strictly weaker than the classic assumptions used for establishing the global convergence of NR with constant stepsize. To show this, we take the assumptions used in the MCT in Section 13.3.4 in Ortega and Rheinboldt (2000) and compare with our assumptions in the following theorem.

**Theorem 2.16.** *Let $F : \mathbb{R}^p \to \mathbb{R}^p$ and let $x^k$ be the iterates of the NR method with stepsize $\gamma = 1$, that is*

$$x^{k+1} \;=\; x^k - \left(DF(x^k)^\top\right)^\dagger F(x^k). \tag{2.36}$$

*Consider the following two sets of assumptions*

*(I) $F(x)$ is component wise convex, $(DF(x)^\top)^{-1}$ exists and is element-wise positive $\forall x \in \mathbb{R}^p$. There exist $x$ and $y$ s.t. $F(x) \leq 0 \leq F(y)$ element-wise.*

*(II) There exists a unique $x^* \in \mathbb{R}^p$ s.t. $F(x^*) = 0$, (2.33) and (2.34) hold for $k \geq 1$.*

*If (I) holds, then (II) always holds. Furthermore, there exist problems for which (II) holds and (I) does not hold.*

We observe that our assumptions are also strictly weaker than the affine covariates formulations of convex functions given in Lemma 3.1 in Deuflhard (2011). The proof is verbatim to the above.

Theorem 2.16 only considers the case that the stepsize $\gamma = 1$. We also investigate the case where the stepsize $\gamma < 1$ in particular in 1-dimension and show that MCT does not hold in this case in Appendix A.3.3. Thus we claim that our assumptions are strictly weaker than the assumptions used in MCT (Ortega and Rheinboldt, 2000; Deuflhard, 2011) for establishing the global convergence of NR, albeit for different step sizes.

## 2.6 Single row sampling: the nonlinear Kaczmarz method

The SNR enjoys many interesting instantiations. Among which, we have chosen three to present in the main text: the nonlinear Kaczmarz method in this section, the Stochastic Newton method (Rodomanov and Kropotov, 2016; Kovalev et al., 2019) in Section 2.7 and a new specialized variant for solving GLMs in Section 2.8.

Here we present the new nonlinear Kaczmarz method as a variant of SNR. Consider the original problem (2.1). We use a single row importance weighted subsampling sketch to sample rows of $F(x) = 0$. That is, let $\Pr[\mathbf{S} = e_i] = p_i$ with the $i$th unit coordinate vector $e_i \in \mathbb{R}^m$ for $i = 1, \cdots, m$. Then the SNR update (line 4 in Algorithm 1) is given by

$$x^{k+1} = x^k - \gamma \frac{F_i(x^k)}{\|\nabla F_i(x^k)\|^2} \nabla F_i(x^k). \tag{2.37}$$

We dub (2.37) the *nonlinear Kaczmarz method*, as it can be seen as an extension of the randomized Kaczmarz method (Kaczmarz, 1937; Strohmer and Vershynin, 2009; Needell, 2010; Needell et al., 2016) for solving linear systems to the nonlinear case[8]. By (2.9), this nonlinear Kaczmarz method is simply SGD applied to minimizing

$$f_{x^k}(x) \;=\; \sum_{i=1}^{m} \Pr[\mathbf{S} = e_i] \, f_{e_i, x^k}(x) \;\overset{(2.9)+(2.4)}{=}\; \frac{1}{2} \sum_{i=1}^{m} p_i \frac{F_i(x)^2}{\|\nabla F_i(x^k)\|^2}.$$

A sufficient condition for (2.10) to hold is that the diagonal matrix

$$\mathbb{E}_{e_i}\left[\mathbf{H}_{e_i}(x^k)\right] \overset{(2.4)}{=} \sum_{i=1}^{m} p_i \frac{e_i e_i^\top}{\|\nabla F_i(x^k)\|^2} = \mathbf{Diag}\left(\frac{p_i}{\|\nabla F_i(x^k)\|^2}\right) \tag{2.38}$$

[8]We note that there exists a nonlinear variant of the Kaczmarz method which is referred to as the Landweber–Kaczmarz method (Leitão and Svaiter, 2016). Though the Landweber–Kaczmarz is very similar to Kaczmarz, it is not truly an extension since it does not adaptively re-weight the stepsize by $\left\|\nabla F_i(x^k)\right\|^2$.

is invertible. Thus $\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x^k)\right]$ is invertible if $\nabla F_i(x^k) \neq 0$ for all $i \in \{1, \cdots, m\}$ and $x^k \in \mathbb{R}^p$. In which case $\mathbf{Ker}\left(\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(y)\right]\right) = \{0\}$ for all $y \in \mathbb{R}^p$ and (2.10) holds.

Finally, to guarantee that (2.37) converges through Theorem 2.7, we need $f_{x^k}(x)$ to be star-convex on $x^k$ at every iteration. In this case, it suffices for each $F_i(x)^2$ to be star-convex, since any conic combination of star-convex functions is star-convex (Lee and Valiant, 2016). This is a straightforward abstraction of the one dimension case, in that, if (2.29) holds for every $F_i$ in the place of $\phi$, we can guarantee the convergence of (2.37). This is also equivalent to assuming the star-convexity for each sketching matrix (2.16). Furthermore, if we have $F(x) \in \mathbf{Im}(DF(x)^\top)$ hold for all $x$, then (2.22) holds, as $\mathbf{Ker}\left(\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(y)\right]\right) = \{0\}$. We can guarantee the convergence of (2.37) through Corollary 2.8.

## 2.7 The Stochastic Newton method

We now show that the Stochastic Newton method (SNM) (Rodomanov and Kropotov, 2016; Kovalev et al., 2019) is a special case of SNR. This connection combined with the global convergence theory of SNR, gives us the first global convergence theory of SNM, which we detail in Section 2.7.2.

SNM (Kovalev et al., 2019) is a stochastic second order method that takes a Newton-type step at each iteration to solve optimization problems with a finite-sum structure

$$\min_{w \in \mathbb{R}^d} \left[ P(w) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \phi_i(w) \right], \tag{2.39}$$

where each $\phi_i : \mathbb{R}^d \to \mathbb{R}$ is twice differentiable and strictly convex. In brevity, the updates in SNM at the $k$th iteration are given by

$$w^{k+1} = \left( \frac{1}{n} \sum_{i=1}^{n} \nabla^2 \phi_i(\alpha_i^k) \right)^{-1} \left( \frac{1}{n} \sum_{i=1}^{n} \nabla^2 \phi_i(\alpha_i^k) \alpha_i^k - \frac{1}{n} \sum_{i=1}^{n} \nabla \phi_i(\alpha_i^k) \right), \tag{2.40}$$

$$\alpha_i^{k+1} = \begin{cases} w^{k+1} & \text{if } i \in B_n \\ \alpha_i^k & \text{if } i \notin B_n \end{cases}, \tag{2.41}$$

where $\alpha_1^k, \cdots, \alpha_n^k$ are auxiliary variables, initialized in SNM, and $B_n \subset \{1, \ldots, n\}$ is a subset of size $\tau$ chosen uniformly on average from all subsets of size $\tau$.

### 2.7.1 Rewrite SNM as a special case of SNR

Since $P(w)$ is strictly convex, every minimizer of $P$ satisfies $\nabla P(w) = \frac{1}{n} \sum_{i=1}^{n} \nabla \phi_i(w) = 0$. Our main insight to deducing SNM is that we can re-write this stationarity condition using

a *variable splitting trick*. That is, by introducing a new variable $\alpha_i \in \mathbb{R}^d$ for each gradient $\nabla\phi_i$, and let $p := (n+1)d$ and $x = \left[w\,;\alpha_1\,;\cdots\,;\alpha_n\right] \in \mathbb{R}^p$ be the stacking[9] of the $w$ and $\alpha_i$ variables, we have that solving $\nabla P(w) = 0$ is equivalent to finding the *roots* of the following nonlinear equations

$$F(x) \;=\; F(w\,;\alpha_1\,;\cdots\,;\alpha_n) \;\overset{\text{def}}{=}\; \left[\frac{1}{n}\sum_{i=1}^{n}\nabla\phi_i(\alpha_i)\;;\; w-\alpha_1\;;\;\cdots\;;\; w-\alpha_n\right], \tag{2.42}$$

where $F : \mathbb{R}^{(n+1)d} \to \mathbb{R}^{(n+1)d}$. Our objective now becomes solving $F(x) = 0$ with $p = m = (n+1)d$. To apply SNR to (2.42), we are going to use a structured sketching matrix. But first, we need some additional notations.

Divide $\mathbf{I}_{nd} \in \mathbb{R}^{nd\times nd}$ into $n$ contiguous blocks of size $nd \times d$ as follows

$$\mathbf{I}_{nd} \;\overset{\text{def}}{=}\; [\,\mathbf{I}_{nd,1}\,\mathbf{I}_{nd,2}\cdots\mathbf{I}_{nd,n}\,]$$

where $\mathbf{I}_{nd,i}$ is the $i$th block of $\mathbf{I}_{nd}$. Let $B_n \subset \{1,\ldots,n\}$ with $|B_n| = \tau$ chosen uniformly at average. Let $\mathbf{I}_{B_n} \in \mathbb{R}^{nd\times\tau d}$ denote the concatenation of the blocks $\mathbf{I}_{nd,i}$ such that the indices $i \in B_n$.

At the $k$th iteration of SNR, denote $x^k = [w^k;\,\alpha_1^k;\,\cdots\,;\,\alpha_n^k]$, we define our sketching matrix $\mathbf{S} \sim \mathcal{D}_{x^k}$ as

$$\mathbf{S} \;=\; \begin{bmatrix} \mathbf{I}_d & 0 \\ \frac{1}{n}\nabla^2\phi_1(\alpha_1^k) & \\ \vdots & \mathbf{I}_{B_n} \\ \frac{1}{n}\nabla^2\phi_n(\alpha_n^k) & \end{bmatrix} \in \mathbb{R}^{(n+1)d\times(\tau+1)d}. \tag{2.43}$$

Here the distribution $\mathcal{D}_{x^k}$ depends on the iterates $x^k$. The sketch size of $\mathbf{S}$ is $(\tau+1)d$ with *any* $\tau \in \{1,\cdots,n\}$. Now we can state the following lemma.

**Lemma 2.17.** *Let $\phi_i$ be strictly convex for $i = 1,\ldots,n$. At each iteration $k$, the updates of SNR (line 4 in Algorithm 1) with $F$ defined in (2.42), the sketching matrix $\mathbf{S}_k$ defined in (2.43) , and stepsize $\gamma = 1$, are equal to the updates (2.40) and (2.41) of SNM.*

Thus we conclude that SNM is a special case of SNR. However, in practice for solving GLMs, instead of sampling $\mathbf{S} \sim \mathcal{D}_{x^k}$ provided in (2.43), we only sample $B_n$ and we execute the efficient updates as suggested in Kovalev et al. (2019).

[9] In this chapter, vectors are columns by default, and given $x_1,\ldots,x_n \in \mathbb{R}^q$, we note $[x_1;\ldots;x_n] \in \mathbb{R}^{qn}$ the (column) vector stacking the $x_i$'s on top of each other with $q \in \mathbb{N}$.

### 2.7.2 Global convergence theory of SNM

Let $x' \overset{\text{def}}{=} (w'; \alpha_1'; \cdots; \alpha_n') \in \mathbb{R}^{(n+1)d}$ and $\mathbf{S} \sim \mathcal{D}_x$ defined in (2.43). By applying the global convergence theory of SNR, we can now provide the first global convergence theory for SNM.

**Corollary 2.18.** *Let $w^*$ be a solution to $\nabla P(w) = 0$. Consider the iterate $x^k = (w^k; \alpha_1^k; \cdots; \alpha_n^k)$ given by SNM (2.40) and (2.41) and note $x^* \overset{def}{=} (w^*; w^*; \cdots; w^*) \in \mathbb{R}^{(n+1)d}$. If there exists $\mu > 0$ such that for all $x, x' \in \mathbb{R}^{(n+1)d}$,*

$$\begin{aligned} f_{x'}(x^*) &\geq f_{x'}(x) + \langle \nabla f_{x'}(x), x^* - x \rangle + \frac{\mu}{2} \|x^* - x\|^2 \\ &= f_{x'}(x) + \langle \nabla f_{x'}(x), x^* - x \rangle + \frac{\mu}{2} \left( \|w^* - w\|^2 + \sum_{i=1}^{n} \|w^* - \alpha_i\|^2 \right), \end{aligned} \tag{2.44}$$

*then the iterates $\{x^k\}$ of SNM converge linearly according to*

$$\mathbb{E}\left[ \left\| x^{k+1} - x^* \right\|^2 \right] \leq (1 - \mu)^{k+1} \left\| x^0 - x^* \right\|^2. \tag{2.45}$$

*Proof.* As $\nabla P(w^*) = 0$, this implies immediately that $x^*$ is a solution of $F$. Besides, (2.44) satisfies Assumption 2.12. Thus by Theorem 2.14, we get (2.45). □

Even though (2.44) is a strong assumption, this is the first global convergence theory of SNM, since only local convergence results of SNM are addressed in Kovalev et al. (2019).

As a by-product, we find that the function $F(x)$ in (2.42) and the sketching matrix $\mathbf{S}$ defined in (2.43) actually satisfy (2.22) through Lemma 2.9, namely as the following lemma.

**Lemma 2.19.** *Consider the function $F$ defined in (2.42) and the sketching matrix $\mathbf{S}$ defined in (2.43), then we have the condition (2.22) hold.*

From Lemma 2.19, we know that for *any* size of the subset sampling $|B_n| = \tau \in \{1, \cdots, n\}$, the condition (2.22) holds. The corresponding sketch size of $\mathbf{S}$ is $(\tau + 1)d$.

Furthermore, using Lemma 2.17, we can also provide the global convergence of SNM with stepsizes $\gamma < 1$ (SNM with relaxation) by using the weaker star-convexity assumption and Theorem 2.7. We expand on this comment in Appendix A.4.3.

## 2.8 Applications to GLMs – *tossing-coin-sketch* method

Consider the problem of training a generalized linear model

$$w^* = \arg\min_{w\in\mathbb{R}^d} P(w) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n}\phi_i(a_i^\top w) + \frac{\lambda}{2}\|w\|^2, \tag{2.46}$$

where $\phi_i : \mathbb{R} \to \mathbb{R}^+$ is a convex and continuously twice differentiable loss function, $a_i \in \mathbb{R}^d$ are data samples and $w \in \mathbb{R}^d$ is the parameter to optimize. As the objective function is strongly convex, the unique minimizer satisfies $\nabla P(w) = 0$, that is

$$\nabla P(w) = \frac{1}{n}\sum_{i=1}^{n}\phi_i'(a_i^\top w)a_i + \lambda w = 0. \tag{2.47}$$

Let $\Phi(w) \overset{\text{def}}{=} \left[\phi_1'(a_1^\top w) \ \cdots \ \phi_n'(a_n^\top w)\right]^\top \in \mathbb{R}^n$ and $\mathbf{A} \overset{\text{def}}{=} \left[a_1 \ \cdots \ a_n\right] \in \mathbb{R}^{d\times n}$. By introducing auxiliary variables $\alpha_i \in \mathbb{R}$ s.t. $\alpha_i \overset{\text{def}}{=} -\phi_i'(a_i^\top w)$, we can re-write (2.47) as

$$w = \frac{1}{\lambda n}\mathbf{A}\alpha, \qquad \text{and} \qquad \alpha = -\Phi(w). \tag{2.48}$$

Note $x = [\alpha; w] \in \mathbb{R}^{n+d}$. The objective of finding the minimum of (2.46) is now equivalent to finding *zeros* for the function

$$F(x) = F(\alpha; w) \overset{\text{def}}{=} \begin{bmatrix} \frac{1}{\lambda n}\mathbf{A}\alpha - w \\ \alpha + \Phi(w) \end{bmatrix}, \tag{2.49}$$

where $F : \mathbb{R}^{n+d} \to \mathbb{R}^{n+d}$. Our objective now becomes solving $F(x) = 0$ with $p = m = n + d$. For this, we will use a variant of the SNR. The advantage in representing (2.47) as the nonlinear system (2.49) is that we now have one row per data point (see the second equation in (2.48)). This allows us to use sketching to *subsample* the data.

Since the function $F$ has a block structure, we will use a structured sketching matrix which we refer to as a *Tossing-coin-sketch*. But first, we need the following definition of a block sketch.

**Definition 2.20** ($(n,\tau)$–block sketch)**.** *Let $B_n \subset \{1,\dots,n\}$ be a subset of size $\tau$ uniformly sampling at random. We say that $\mathbf{S} \in \mathbb{R}^{n\times\tau}$ is a $(n,\tau)$–block sketch if $\mathbf{S} = \mathbf{I}_{B_n}$ where $\mathbf{I}_{B_n}$ denotes the column concatenation of the columns of the identity matrix $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ whose indices are in $B_n$.*

Our Tossing-coin-sketch is a sketch that alternates between two blocks depending on the result of a coin toss.

**Definition 2.21** (Tossing-coin-sketch). *Let* $\mathbf{S}_d \in \mathbb{R}^{d\times\tau_d}$ *and* $\mathbf{S}_n \in \mathbb{R}^{n\times\tau_n}$ *be a* $(d,\tau_d)$*–block sketch and a* $(n,\tau_n)$*–block sketch, respectively. Let* $b \in (0;1)$*. Now each time we sample* $\mathbf{S}$*, we "toss a coin" to determine the structure of* $\mathbf{S} \in \mathbb{R}^{(d+n)\times(\tau_d+\tau_n)}$*. That is,* $\mathbf{S} = \begin{bmatrix} \mathbf{S}_d & 0 \\ 0 & 0 \end{bmatrix}$ *with probability* $1-b$ *and* $\mathbf{S} = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{S}_n \end{bmatrix}$ *with probability* $b$*.*

By applying the SNR method with a tossing-coin-sketch for solving (2.49), we arrive at an efficient method for solving (2.46) that we call the *TCS* method. By using a tossing-coin-sketch, we can alternate between solving a linear system based on the first $d$ rows of (2.49) and a nonlinear system based on the last $n$ rows of (2.49).

TCS is inspired by the first-order stochastic dual ascent methods (Shalev-Shwartz and Zhang, 2013; Shalev-Shwartz, 2016; Qu et al., 2015). Indeed, equation (2.48) can be seen as primal-dual systems with primal variables $w$ and dual variables $\alpha$ induced by the Legendre–Fenchel transformation. Stochastic dual ascent methods are efficient to solve (2.48). At each iteration, they update alternatively the primal and the dual variables $w$ and $\alpha$ with the first-order information. Thus, by sketching alternatively the primal and the dual systems and updating accordingly with the Newton-type steps, TCS's updates can be seen as the second-order stochastic dual ascent methods.

Next we show that the TCS method verifies (2.22) through Lemma 2.9 in the following.

**Lemma 2.22.** *Consider the function* $F$ *defined in* (2.49) *and the tossing-coin-sketch* $\mathbf{S}$ *defined in Definition 2.21, then* (2.22) *holds.*

From Lemma 2.22, we know that for *any* size of the block sketch $\tau_d \in \{1,\cdots,d\}$ and $\tau_n \in \{1,\cdots,n\}$, (2.22) holds. The corresponding sketch size of $\mathbf{S}$ is $\tau_d + \tau_n$.

Using sketch sizes s.t. $\tau_n \ll n$, the TCS method has the same cost as SGD in the case $d \ll n$. The low computational cost per iteration is thus another advantage of the TCS method. See Appendix A.10 the cost per iteration analysis. For a detailed derivation of the TCS method, see Appendix A.8 and a detailed implementation in Algorithm 7 in the appendix.

**Table 2.1** – Details of the data sets for binary classification

| dataset | dimension ($d$) | samples ($n$) | C.N. of the model | $L$ |
|---|---|---|---|---|
| covetype | 54 | 581012 | $7.45 \times 10^{12}$ | $1.28 \times 10^{7}$ |
| a9a | 123 | 32561 | $5.12 \times 10^{4}$ | 1.57 |
| fourclass | 2 | 862 | $4.86 \times 10^{6}$ | $5.66 \times 10^{3}$ |
| artificial | 50 | 10000 | $3.91 \times 10^{4}$ | 3.91 |
| ijcnn1 | 22 | 49990 | $2.88 \times 10^{3}$ | $5.77 \times 10^{-2}$ |
| webspam | 254 | 350000 | $7.47 \times 10^{4}$ | $2.13 \times 10^{-1}$ |
| epsilon | 2000 | 400000 | $3.51 \times 10^{4}$ | $8.76 \times 10^{-2}$ |
| phishing | 68 | 11055 | $1.04 \times 10^{3}$ | $9.40 \times 10^{-2}$ |

### 2.8.1 Experiments for TCS method applied for GLM

We consider the logistic regression problem with 8 datasets[10] taken from LibSVM (Chang and Lin, 2011), except for one artificial dataset. Table 2.1 provides the details of these datasets, including the *condition number* (C.N.) of the model and the smoothness constant $L$ of the model. C.N. of the logistic regression problem is given by

$$\text{C.N.} \overset{\text{def}}{=} \frac{\lambda_{\max}(\mathbf{A}\mathbf{A}^\top)}{4n\lambda} + 1,$$

where $\lambda_{\max}(\cdot)$ is the largest eigenvalue operator. The smoothness constant $L$ is given by

$$L \overset{\text{def}}{=} \frac{\lambda_{\max}(\mathbf{A}\mathbf{A}^\top)}{4n} + \lambda.$$

As for the logistic regression problem, we consider the loss function $\phi_i$ in (2.46) in the form

$$\phi_i(t) = \ln(1 + \mathrm{e}^{-y_i t})$$

where $y_i$ are the target values for $i = 1, \cdots, n$.

**The artificial dataset.** The artificial dataset $\mathbf{A}^\top \in \mathbb{R}^{n \times d}$ in Table 2.1 is of size $10000 \times 50$ and generated by a Gaussian distribution whose mean is zero and covariance is a Toeplitz matrix. Toeplitz matrices are completely determined by their diagonal. We set the diagonal of our Toeplitz matrix as

$$[c^0; c^1; \cdots; c^{d-1}] \in \mathbb{R}^d$$

where $c \in \mathbb{R}^+$ is a parameter. We choose $c = 0.9$ (closed to 1) which results in $\mathbf{A}$ having highly correlated columns, which in turn makes $\mathbf{A}$ an ill-conditioned data set. We set the ground

[10] All datasets except for the artificial dataset can be found downloaded on https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/. Some of the datasets can be found originally in Kohavi (1996), Mohammad et al. (2012), Chang and Lin (2001), Blackard and Dean (1999), Wang et al. (2012), and Dua and Graff (2017).

truth coefficients of the model

$$\mathbf{w} = [(-1)^0 \cdot \mathrm{e}^{-\frac{0}{10}}; \cdots; (-1)^{d-1} \cdot \mathrm{e}^{-\frac{d-1}{10}}] \in \mathbb{R}^d$$

and the target values of the dataset

$$\mathbf{y} = \operatorname{sgn}\left(\mathbf{A}^\top \mathbf{w} + \mathbf{r}\right) \in \mathbb{R}^n$$

where $\mathbf{r} \in \mathbb{R}^n$ is the noise generated from a standard normal distribution.

We compare the TCS method with SAG (Schmidt et al., 2017), SVRG (Johnson and Zhang, 2013), dfSDCA (Shalev-Shwartz, 2016) and Quartz (Qu et al., 2015). All experiments were initialized at $w^0 = 0 \in \mathbb{R}^d$ (and/or $\alpha^0 = 0 \in \mathbb{R}^n$ for TCS/dfSDCA methods) and were performed in Python 3.7.3 on a laptop with an Intel Core i9-9980HK CPU and 32 Gigabyte of DDR4 RAM running OSX 10.14.5. For all methods, we used the stepsize that was shown to work well in practice. For instance, the common rule of thumb for SAG and SVRG is to use a stepsize $\frac{1}{L}$, where $L$ is the smoothness constant. This rule of thumb stepsize is not supported by theory. Indeed for SAG, the theoretical stepsize is $\frac{1}{16L}$ and it should be even smaller for SVRG depending on the condition number. For dfSDCA and Quartz's, we used the stepsize suggested in the experiments in Shalev-Shwartz (2016) and Qu et al. (2015) respectively. For TCS, we used two types of stepsize, related to the C.N. of the model. If the condition number is big (Figure 2.1 top row), we used $\gamma = 1$ except for a9a with $\gamma = 1.5$. If the condition number is small (Figure 2.1 bottom row), we used $\gamma = 1.8$. We also set the Bernoulli parameter $b$ (probability of the coin toss) depending on the size of the dataset (see Table A.1 in Appendix A.10), and $\tau_d = d$. We tested three different sketch sizes $\tau_n = 50, 150, 300$. More details of the parameter settings are presented in Appendix A.10.

We used $\lambda = \frac{1}{n}$ regularization parameter, evaluated each method 10 times and stopped once the gradient norm[11] was below $10^{-5}$ or some maximum time had been reached. In Figure 2.1, we plotted the central tendency as a solid line and all other executions as a shaded region for the wall-clock time vs gradient norm.

From Figure 2.1, TCS outperforms all other methods on ill-conditioned problems (Figure 2.1 top row), but not always the case on well-conditioned problems (Figure 2.1 bottom row). This is because in ill-conditioned problems, the curvature of the optimization function is not uniform over directions and varies in the input space. Second-order methods effectively exploit information of the local curvature to converge faster than first-order methods. To further illustrate the performance of TCS on ill-conditioned problems, we compared the performance of TCS on the artificial dataset in the top right of Figure 2.1. Note as well that for reaching an approximate solution at early stage (i.e. $tol = 10^{-3}, 10^{-4}$), TCS is very competitive on all

[11] We evaluated the true gradient norm every 1000 iterations. We also paused the timing when computing the performance evaluation of the gradient norm.

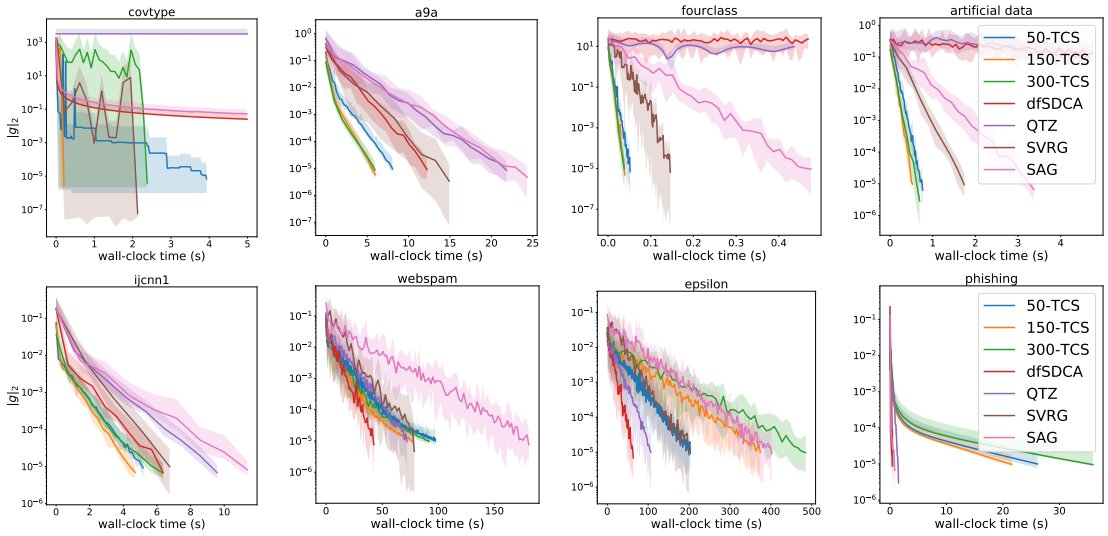


**Figure 2.1** – Experiments for TCS method applied for generalized linear model.

problems. TCS also has the smallest variance compared to the first-order methods based on eye-balling the shaded error bars in Figure 2.1, especially compared to SVRG. Among the three tested sketch sizes, 150 performed the best except on the *epsilon* dataset.

## 2.9 Discussion and bibliographical remarks

In this chapter, we introduced the SNR method, for which we provided strong convergence guarantees. We also developed several promising applications of SNR to show that SNR is very flexible and tested one of these specialized variants for training GLMs. SNR is flexible by the fact that its primitive goal is to solve efficiently nonlinear equations. Since there are many ways to re-write an optimization problem as nonlinear equations, each re-write leads to a distinct method, thus leads to a specific implementation in practice (e.g. SNM, TCS methods) when using SNR. Besides, the convergence theories presented in Section 2.4 guarantee a large variety of choices for the sketch.

A natural question opened by our work was whether this flexibility would allow us to discover other applications of SNR. This was answered positively by the work of Gower et al. (2021a), which devised a variant of Stochastic Polyak (SP) method (Berrada et al., 2020; Loizou et al., 2021) based on the Polyak step size (Polyak, 1987). In particular, Gower et al. (2021a) show that SP under the interpolation condition (Crammer et al., 2006; Vaswani et al., 2019a) is the nonlinear Kaczmarz methods (2.37). Thus SP is a special case of SNR. Using this viewpoint, and leveraging our convergence results, Gower et al. (2021a) further show that SP can also be viewed as a type of online SGD method, which facilitates the analysis of SP. On the experimental side, Gower et al. (2021a) show that the SP method is very competitive as compared to ADAM

(Kingma and Ba, 2015) for the training of deep neural networks on computer vision tasks, CIFAR10 (Krizhevsky, 2009) and SVHN (Netzer et al., 2011), and on NLP benchmarks, the IWSLT14 English-German translation task (Cettolo et al., 2014).

As a follow-up work of SNR, the nonlinear Kaczmarz methods (2.37) and their convergence were also recently investigated by Zeng and Ye (2020) and Wang et al. (2022) with different choices of sampling. Faster nonlinear Kaczmarz methods with some greedy sampling rules were further developed by Zhang et al. (2022c) and Zhang and Li (2022).

There are also many possible ways to extend SNR. Indeed, recall the sketch-and-project viewpoint of SNR in (2.7)

$$x^{k+1} = \operatorname{argmin}_{x\in\mathbb{R}^p} \left\|x - x^k\right\|^2 \tag{2.50}$$

$$\text{s. t.} \quad \mathbf{S}_k^\top DF(x^k)^\top (x - x^k) = -\gamma \mathbf{S}_k^\top F(x^k). \tag{2.51}$$

One extension of SNR is when the updates are relaxed to use inequality constraints instead of equality constraints in (2.51), which was recently proposed by Gower et al. (2022). Second possible extension of SNR is that, instead of projecting onto the linearization of $\mathbf{S}_k^\top F(x^k)$, one can use the local second-order expansion. That is, as a proxy of setting $\mathbf{S}_k^\top F(x) = 0$, we set the second-order expansion of $\mathbf{S}_k^\top F(x)$ around $x^k$ to be zero

$$\mathbf{S}_k^\top F(x^k) + \mathbf{S}_k^\top DF(x^k)^\top (x - x^k) + \frac{1}{2}(x - x^k)^\top \mathbf{S}_k^\top D^2 F(x^k)(x - x^k) = 0, \tag{2.52}$$

where $D^2F(x^k)$ is the second-order derivatives of $F(x^k)$. Hence, we replace the linear constraints (2.51) by the second-order constraints (2.52). If we choose the sketching matrix $\mathbf{S}_k$ as a single row subsampling sketch, this becomes SP2 which was recently proposed by Li et al. (2022a).

Another extension of SNR is to use variable metric for the projection rather than L2 norm projection in (2.50). This was known as *Sketched Newton-Raphson with Variable Metric* (SNRVM) developed in our later work Chen et al. (2022a), which will be presented in Section 3.4 in the next Chapter 3.

# Chapter 3

# SAN: Stochastic Average Newton Algorithm for Minimizing Finite Sums

In this chapter, we present a principled approach for designing stochastic Newton methods for solving finite sum optimization problems. Our approach has two steps. First, we re-write the stationarity conditions as a system of nonlinear equations that associates each data point to a new row. Second, we apply a Subsampled Newton Raphson method to solve this system of nonlinear equations. Using our approach, we develop a new Stochastic Average Newton (SAN) method, which is incremental by design, in that it requires only a single data point per iteration. It is also cheap to implement when solving regularized generalized linear models, with a cost per iteration of the order of the number of the parameters. We show through numerical experiments that SAN requires no knowledge about the problem, neither parameter tuning, while remaining competitive as compared to classical variance reduced gradient methods (e.g. SAG and SVRG), incremental Newton and quasi-Newton methods (e.g. SNM, IQN). [1]

## Contents



[1] This chapter is based on an article published in the proceedings of the $25^{th}$ International Conference on Artificial Intelligence and Statistics (AISTATS 2022) (Chen et al., 2022a).

## 3.1 Introduction

In contrast to the previous chapter, where we aim to solve the general nonlinear equation, in this chapter we consider more precisely the problem of minimizing a sum of terms

$$w^* \in \operatorname*{argmin}_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(w) \stackrel{\text{def}}{=} f(w), \tag{3.1}$$

where $f_i$ is a convex twice differentiable loss over a given $i$-th data point. Because of the specific finite sum structure, we can design more adaptive algorithm, like TCS method in the previous Section 2.8 (Algorithm 7), than the SNR method (Algorithm 1), which is for solving nonlinear equation in general.

When the number of *data points* $n$ and *features* $d$ are large, first order methods such as Stochastic Gradient Descent (Robbins and Monro, 1951, SGD), SAG (Schmidt et al., 2017), SVRG (Johnson and Zhang, 2013) and ADAM (Kingma and Ba, 2015) are the methods of choice for solving (3.1) because of their low cost per iteration. The issue with first order methods is that they can require extensive parameter tuning, and/or knowledge of the parameters of the problem. Consequently, to make a first order method work well requires careful tweaking and tuning from an expert, and a careful choice of the model itself. Indeed, neural networks have evolved in such a way that allows for SGD to converge, such as the introduction of batch norm (Ioffe and Szegedy, 2015) and the push for more over-parametrized networks which greatly speed-up the convergence of SGD (Ma et al., 2018; Vaswani et al., 2019a; Gower et al., 2021c). Thus the reliance on first order methods ultimately restricts the choice and development of alternative models.

There is now a concerted effort to develop efficient stochastic second order methods that can exploit the sum of terms structured in (3.1). The hope for second order methods for solving (3.1) is that they require less parameter tuning and converge for wider variety of models and datasets. In particular, here we set out to develop stochastic second order methods that achieve the following objective.

**Objective 3.1.** *Develop a second order method for solving* (3.1) *that is* incremental, efficient, *scales well with the dimension* $d$*, and that requires no* knowledge from the problem, *neither* parameter tuning.

Most stochastic second order methods are not incremental, and thus fall short of our first criteria. This is due to the fact that most of these methods are only guaranteed to work in a large mini-batch size regime, and not with a single sample. For instance, the subsampled Newton methods (Roosta-Khorasani and Mahoney, 2019; Bollapragada et al., 2018; Liu and

Roosta, 2021; Erdogdu and Montanari, 2015; Kohler and Lucchi, 2017) require potentially large mini-batch sizes in order to guarantee that the subsampled Newton direction closely matches the full Newton direction in high probability. Stochastic quasi-Newton methods (Byrd et al., 2011; Mokhtari and Ribeiro, 2015; Moritz et al., 2016; Gower et al., 2016), SDNA (Qu et al., 2016), the Newton sketch (Pilanci and Wainwright, 2017) and Lissa (Agarwal et al., 2017), suffer from the same drawback: the need for large mini-batches or full gradient evaluation to work, which makes them all not incremental.

The two existing methods that we are aware of that are truly incremental are *IQN* (Incremental Quasi-Newton) (Mokhtari et al., 2018; Gao et al., 2020) and *SNM* (Stochastic Newton Method) (Kovalev et al., 2019; Rodomanov and Kropotov, 2016). Both methods also enjoy a fast local convergence rate. Their only drawback is their computational and memory costs per iteration are at least $\mathcal{O}(d^2)$ (see Table 3.1 and Section B.3.4 for more details). This is prohibitive in a setting where the number of parameters for the model is large. Our goal is to develop a method that is not only incremental, but also has a cost per iteration of $\mathcal{O}(d)$, as is the case for first-order methods like SGD.

In this chapter, we develop two new Newton methods for solving (3.2) that effectively make use of second order information, are incremental, and are governed by a single global convergence theory. Our starting point for developing these methods is to re-write the stationarity conditions

$$\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(w) = 0. \tag{3.2}$$

At this point, we could apply Newton's method for solving nonlinear equations, otherwise known as the Newton Raphson method. However, this approach would ultimately require a full pass over the data at each iteration.

To avoid taking full passes over the data, we re-write (3.2) by introducing $n$ auxiliary variables $\alpha_i \in \mathbb{R}^d$ and solving instead the nonlinear system given by

$$\frac{1}{n}\sum_{i=1}^{n}\alpha_i = 0, \tag{3.3}$$

$$\alpha_i = \nabla f_i(w), \quad \forall i \in \{1,\dots,n\}. \tag{3.4}$$

Clearly (3.3–3.4) have the same solutions in $w$. The advantage of (3.3–3.4) is that each gradient lies on a separate row. Consequently, applying a *subsampled* Newton Raphson method, that is sampling a row and then applying Newton Raphson, to (3.3–3.4) will result in an incremental method. We refer to (3.3–3.4) as the *function splitting* formulation, since it splits the gradient across different rows.

To solve (3.3–3.4) efficiently, we propose SAN (*Stochastic Average Newton*) in Section 3.2.1. SAN is a subsampled Newton Raphson method that is based on a new variable metric extension of *SNR* (Sketch Newton Raphson Method) (Yuan et al., 2022b) that we present in Section 3.4, which is itself a nonlinear extension of the Sketch-and-Project method for solving linear systems (Gower and Richtárik, 2015b). By using a different subsampling of the rows (3.3–3.4), we also derive SANA in Section 3.2.2, which is a variant of SAN that uses unbiased estimates of the gradient.

Note that the idea of applying a subsampled Newton method to a well-chosen system of optimality conditions is not new. Indeed, it was recently shown in Yuan et al. (2022b) that the SNM method (Kovalev et al., 2019) can be seen as the application of a subsampled Newton method to the equations

$$\begin{aligned} \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\alpha_i) = 0, & \\ w = \alpha_i, \quad \text{for } i = 1, \ldots, n, & \end{aligned} \tag{3.5}$$

which clearly are equivalent to (3.2). Consequently, the two methods SAN and SNM are both subsampled Newton Raphson methods applied to either a function or a *variable splitting* formulation of (3.2).

The contributions of this chapter are the following:

- We propose combining the function splitting reformulation (3.3–3.4) with a subsampled Newton Raphson as a tool for designing stochastic Newton methods, all of which are *variance reducing* in the sense that they are incremental and they converge with a constant step size.
- We introduce SAN (Stochastic Average Newton method) by using this tool, which is incremental and parameter-free, in that, SAN works well with a step size $\gamma = 1$ independently of the underlying dataset or the objective function.
- By specializing to GLMs (Generalized Linear Models), we develop an efficient implementation of SAN that has the same cost per iteration as the first-order methods. We perform extensive numerical experiments and show that SAN is competitive as compared to SAG and SVRG.
- To provide a convergence theory of our methods, we extend the class of Sketched Newton Raphson (Yuan et al., 2022b) methods to allow for a variable metric that includes SAN as a special case.

In Section 3.2, we show how to derive the SAN and SANA methods. We then present two different experimental settings comparing SAN to variance reduced gradients methods

in Section 3.3. In Section 3.4, we study SAN/SANA as instantiations of a new *variable metric* Sketch Newton Raphson method and present a convergence theory for this class of method.

The following will be assumed throughout the chapter.

**Assumption 3.2.** *For all $i \in \{1, \dots, n\}$, the function $f_i : \mathbb{R}^d \longrightarrow \mathbb{R}$ is of class $C^2$ and verifies $\nabla^2 f_i(w) \succ 0$ for every $w \in \mathbb{R}^d$.*

## 3.2 Function splitting methods

The advantage of the function splitting formulation given by (3.3) and (3.4) is that there is a separate row for each data point. We will now take advantage of this, and develop new incremental Newton methods based on subsampling the rows of (3.3–3.4).

The reformulation given in (3.3–3.4) is a large system of nonlinear equations. For brevity, let $p := (n+1)d$ and $x = \left[w\,;\alpha_1\,;\cdots\,;\alpha_n\right] \in \mathbb{R}^p$ be the stacking[2] of the $w$ and $\alpha_i$ variables. Thus solving (3.3–3.4) is equivalent to solve $F(x) = 0$, where

$$\begin{aligned} F : \mathbb{R}^p &\to \mathbb{R}^p \\ x &\mapsto \left[\tfrac{1}{n}\textstyle\sum \alpha_i; \nabla f_1(w) - \alpha_1; \cdots; \nabla f_n(w) - \alpha_n\right]. \end{aligned} \tag{3.6}$$

As seen in Chapter 1, solving nonlinear equations has long been one of the core problems in numerical analysis, with variants of the Newton Raphson method (Ortega and Rheinboldt, 2000) being one of the core techniques. From a given iterate $x^k \in \mathbb{R}^p$, the Newton Raphson method computes the next iterate $x^{k+1}$ by linearizing $F$ around $x^k$ and solving the *Newton system*

$$\nabla F(x^k)^\top (x^{k+1} - x^k) = -F(x^k). \tag{3.7}$$

Here $\nabla F(x) \in \mathbb{R}^{p \times p}$ denotes the Jacobian matrix of $F$ at $x$, and it is assumed that (3.7) has a solution. The least norm solution of the Newton system is given by

$$x^{k+1} = x^k - \nabla F(x^k)^{\top\dagger} F(x^k), \tag{3.8}$$

where $^\dagger$ denotes the Moore-Penrose pseudoinverse.

This update can also be written as a projection step:

$$x^{k+1} = \operatorname{argmin} \|x - x^k\|^2$$

[2] In this chapter, vectors are columns by default, and given $x_1, \dots, x_n \in \mathbb{R}^q$ we note $[x_1; \dots; x_n] \in \mathbb{R}^{qn}$ the (column) vector stacking the $x_i$'s on top of each other.

$$\text{s.t. } \nabla F(x^k)^\top (x - x^k) = -F(x^k). \tag{3.9}$$

In our setting, (3.8) is prohibitively expensive because it requires access to all of the data at each step and the solution of a large $(n+1)d \times (n+1)d$ linear system. To bring down the cost of each iteration, and to have a resulting incremental method, at each iteration we will *subsample* the rows of the Newton system before taking a projection step. Next, we present two methods based on subsampling. Later on Section 3.4, we generalize this subsampling approach to make use of *sketches* of the system.

### 3.2.1 SAN: the Stochastic Average Newton method

The SAN method is a subsampled Newton Raphson method that alternates between sampling equation (3.3) or sampling one of the equations in (3.4). After sampling, we then apply a step of Newton Raphson to the sampled equation.

To detail the SAN method, let $\pi \in (0, 1)$ be a fixed probability, and let $x^k = [w^k; \alpha_1^k; \ldots; \alpha_n^k] \in \mathbb{R}^p$ be a given $k$-th iterate. With probability $\pi$ the SAN method samples equation (3.3) and focuses on finding a solution to this equation. Since (3.3) is a linear equation, it is equal to its own Newton equation. Furthermore, this linear equation (3.3) has $n$ variables and only one equation, thus it has infinite solutions. We choose a single one of these infinite solution by using a projection step

$$\begin{aligned} \alpha_1^{k+1}, \ldots, \alpha_n^{k+1} = \underset{\alpha_1, \ldots, \alpha_n \in \mathbb{R}^d}{\operatorname{argmin}} & \textstyle\sum_{i=1}^n \left\| \alpha_i - \alpha_i^k \right\|^2 \\ \text{s.t. } & \tfrac{1}{n} \textstyle\sum_{i=1}^n \alpha_i = 0. \end{aligned} \tag{3.10}$$

The solution to this projection is given in line 4 in Algorithm 2 when $\gamma = 1$. We have added the step size $\gamma \in (0,\ 1]$ to act as relaxation.

Alternatively, with probability $(1 - \pi)$ the SAN method then samples the $j$-th equation in (3.4) uniformly among the $n$ equations. To get the Newton system of $\nabla f_j(w) = \alpha_j$, we linearize around $w^k \in \mathbb{R}^d$ and $\alpha_j^k \in \mathbb{R}^d$ and set the linearization to zero giving

$$\nabla f_j(w^k) + \nabla^2 f_j(w^k)(w - w^k) = \alpha_j.$$

This linear equation has $2d$ unknowns, and thus also has infinite solutions. Again, we use a projection step to pick a unique solution as follows

$$\begin{aligned} \alpha_j^{k+1}, w^{k+1} = \underset{\alpha_j, w \in \mathbb{R}^d}{\operatorname{argmin}} & \left\| \alpha_j - \alpha_j^k \right\|^2 + \left\| w - w^k \right\|^2_{\nabla^2 f_j(w^k)} \\ \text{s.t. } & \nabla f_j(w^k) + \nabla^2 f_j(w^k)(w - w^k) = \alpha_j. \end{aligned} \tag{3.11}$$

Here we have introduced a projection under a norm $\|w\|_{\nabla^2 f_j(w^k)} \overset{\text{def}}{=} \left\langle \nabla^2 f_j(w^k) w, w \right\rangle$ which is based on the Hessian matrix $\nabla^2 f_j(w^k)$. Performing a projection step with respect to the metric induced by the Hessian is often used in Newton type methods such as interior point methods (Renegar, 2001) and quasi-Newton methods (Goldfarb, 1970). Moreover, we observed that this choice of metric resulted in a much faster algorithm (see Section B.3.5 for experiments that highlight this). The closed form solution to the above is given in lines 8-10 in Algorithm 2 when $\gamma = 1$ (see Lemma B.2 for the details).

We gather all these updates in Algorithm 2 and call the resulting method the *Stochastic Average Newton* method, or *SAN* for short.

**Algorithm 2:** SAN: Stochastic Average Newton

**Input:** $\{f_i\}_{i=1}^n$, step size $\gamma \in (0, 1]$, probability $\pi \in (0, 1)$, max iteration $T$

1 Initialize $\alpha_1^0, \cdots, \alpha_n^0, w^0 \in \mathbb{R}^d$

2 **for** $k = 1, \ldots, T$ **do**

3 **With probability** $\pi$ **update:**

4 $\alpha_i^{k+1} = \alpha_i^k - \frac{\gamma}{n} \sum_{j=1}^n \alpha_j^k, \ \forall i \in \{1, \cdots, n\}$

5 **Otherwise with probability** $(1 - \pi)$**:**

6 Sample uniformly $j \in \{1, \cdots, n\}$

7 $\mathbf{H}_k = \mathbf{I}_d + \nabla^2 f_j(w^k)$

8 $d^k = -\mathbf{H}_k^{-1} \left( \nabla f_j(w^k) - \alpha_j^k \right)$

9 $w^{k+1} = w^k + \gamma d^k$

10 $\alpha_j^{k+1} = \alpha_j^k - \gamma d^k$

**Output:** Last iterate $w_{T+1}$

The SAN method is incremental, since it can be applied with as little as one data point per iteration. SAN can also be implemented in such a way that the cost per iteration is $\mathcal{O}(d)$ in expectation. Indeed, the *averaging step* on line 4 contributes with a $\pi \times \mathcal{O}(nd)$ cost to the total cost in expectation, since all of the vectors $\alpha_i \in \mathbb{R}^d$ for $i = 1, \ldots, n$, are updated. But as we found through expensive testing in Section B.3.3, SAN converges quickly if $\pi$ is of the order of $\mathcal{O}(1/n)$, reducing the cost in expectation to $\mathcal{O}(d)$. Further, the average of the $\alpha_i$'s can be efficiently implemented by maintaining and updating a variable $\overline{\alpha}^k = \frac{1}{n} \sum_{j=1}^n \alpha_j^k$. The main cost for SAN is in solving the linear system $(\mathbf{I}_d + \nabla^2 f_j(w^k))d = \alpha_j^k - \nabla f_j(w^k)$. Solving this system with a direct solver would cost $\mathcal{O}(d^3)$. Alternatively, the solution can be approximated using an iterative Krylov method for which each iteration costs $\mathcal{O}(d)$ by using backpropagation (Freund et al., 1992; Christianson, 1992) to compute Hessian-vector products. For regularized generalized linear models (GLMs), the total cost of this matrix inversion is only $\mathcal{O}(d)$ operations, as we show next.

**Generalized Linear Models.** Regularized GLMs are models for which we have

$$f_i(w) \;=\; \phi_i(a_i^\top w) + \lambda R(w), \tag{3.12}$$

where $\phi_i : \mathbb{R} \to \mathbb{R}$ is a loss function associated with the $i$-th data point $a_i \in \mathbb{R}^d$, $\lambda > 0$ is a regularization parameter and $R$ is a regularizer that is twice differentiable and separable, i.e. $R(w) = \sum_{i=1}^d R_i(w_i)$ with $R_i : \mathbb{R} \to \mathbb{R}$. The inversion on line 8 of Algorithm 2 can be efficiently computed using the Woodbury identity because the Hessian $\nabla^2 f_j(w) \;=\; \phi_j''(a_j^\top w)(a_j a_j^\top) + \lambda \nabla^2 R(w)$ is a rank-one perturbation of a diagonal matrix, which costs $\mathcal{O}(d)$ to invert (see Lemma B.7 for an explicit formula).

**Remark 3.3** (SAN vs. SNM for GLMs)**.** *SAN can be implemented efficiently for all GLMs with separable regularizers. This is not the case for SNM (Kovalev et al., 2019), which can only be implemented efficiently when the regularizer is the L2 norm. For other separable regularizers, the cost per iteration for SNM is to $\mathcal{O}(d^3)$ instead of $\mathcal{O}(d^2)$. See Appendix B.3.4 for details.*

### 3.2.2 SANA: alternative with simultaneous projections

Here we present SANA, an alternative version of the SAN method. Instead of alternating between projecting onto linearizations of (3.3) and (3.4), the SANA method projects onto the intersection of (3.3) and the linearization of a subsampled equation (3.4). In other words, the next iterate $x^{k+1} = [w^{k+1}; \alpha_1^{k+1}; \ldots; \alpha_n^{k+1}]$ is defined as the unique solution of

$$\begin{aligned} \operatorname*{argmin}_{w,\alpha_1,\ldots,\alpha_n \in \mathbb{R}^d} & \sum_{i=1}^n \left\|\alpha_i - \alpha_i^k\right\|^2 + \left\|w - w^k\right\|^2_{\nabla^2 f_j(w^k)}, \\ \text{s.t. } & \nabla f_j(w^k) + \nabla^2 f_j(w^k)(w - w^k) = \alpha_j, \\ & \tfrac{1}{n}\textstyle\sum_{i=1}^n \alpha_i \;=\; 0. \end{aligned} \tag{3.13}$$

The closed form solution of (3.13) corresponds to lines 4–8 in Algorithm 3 when the relaxation parameter is $\gamma = 1$ (see Lemma B.4 for a proof).

Computing one step of this method requires access to only one function $f_j$ (through its gradient and Hessian evaluated at $w^k$). In terms of computational cost, each step requires inverting the $d \times d$ matrix $(1 - \frac{1}{n})I_d + \nabla^2 f_j(w^k)$. As with SAN, this cost reduces to $\mathcal{O}(d)$ in the context of generalized linear models. See Algorithm 10 in the appendix for the resulting implementation for GLMs. Yet even in the case of GLMs, the SANA method costs $\mathcal{O}(nd)$ per iteration because it updates all the $\alpha_i$ vectors at every iteration. Thus the SANA method has complexity which is $\mathcal{O}(n)$ times larger than SAN in expectation.

**Algorithm 3:** SANA

**Input:** $\{f_i\}_{i=1}^n$, step size $\gamma \in (0, 1]$, max iteration $T$

Initialize $w^0, \alpha_1^0, \cdots, \alpha_n^0 \in \mathbb{R}^d$ s.t. $\sum_{i=1}^n \alpha_i^0 = 0$

**for** $k = 1, \ldots, T$ **do**

  Sample uniformly $j \in \{1, \ldots, n\}$

  $\mathbf{H}_k = (1 - \frac{1}{n})\mathbf{I}_d + \nabla^2 f_j(w^k)$

  $d^k = -\mathbf{H}_k^{-1}(\nabla f_j(w^k) - \alpha_j^k)$

  $w^{k+1} = w^k + \gamma d^k$

  $\alpha_j^{k+1} = \alpha_j^k - \gamma(1 - \frac{1}{n})d^k$

  $\alpha_i^{k+1} = \alpha_i^k + \frac{\gamma}{n} d^k, \quad$ for $i \neq j$

**Output:** Last iterate $w_{T+1}$

Both SAN and SANA can be interpreted as a stochastic relaxed Newton method that uses estimates of the gradient. Indeed, computing $d^k$ in Algorithms 2 and 3 requires solving a relaxed Newton system

$$\left(\delta \mathbf{I}_d + \nabla^2 f_j(w^k)\right) d^k = \alpha_j^k - \nabla f_j(w^k), \tag{3.14}$$

where $\delta = 1$ and $\delta = 1 - \frac{1}{n}$, respectively. The right hand side of this Newton system is a biased estimate of the gradient for SAN and an unbiased estimate for SANA. To see this, for simplicity, let $\gamma = 1$. Taking expectation conditioned on time $k$ over the right-hand side of (3.14) gives

$$\mathbb{E}\left[\alpha_j^k - \nabla f_j(w^k)\right] = \frac{1}{n}\sum_{i=1}^n \alpha_i^k - \nabla f(w^k).$$

For SANA, because the averaging constraint is always enforced in (3.13), we have that $\frac{1}{n}\sum_{i=1}^n \alpha_i^k = 0$, thus the right hand side of (3.14) is always an unbiased estimate of the negative gradient. As for SAN, the averaging constraint is only enforced every so often with the update on line 4 in Algorithm 2. Thus for SAN, the right hand is a biased estimate of the negative gradient until the averaging constraint is enforced. In this sense, SAN and SANA are analogous to SAG (Schmidt et al., 2017) and SAGA (Defazio et al., 2014). We found in practice that this biased estimate of the gradient did not hurt the empirical performance of SAN, and thus we focus on experiments on SAN in Section 3.3.

## 3.3 Experiments for SAN applied for GLMs

Here we compare SAN in Algorithm 2 against two variance reduced gradient methods SAG (Schmidt et al., 2017) and SVRG (Johnson and Zhang, 2013) for solving regularized GLMs (3.12), where

$\phi_i(t) = \log\left(1 + e^{-y_i t}\right)$ is the logistic loss, $y_i \in \{-1, 1\}$ is the $i$-th target value, and $R$ is the regularizer.

We use eight datasets in our experiments taken from LibSVM (Chang and Lin, 2011),[3] with disparate properties (see details of the datasets in Table B.1). We fixed an initial random seed, evaluated each method 10 times, and stopped when the gradient norm was below $10^{-6}$ or a maximum of 50 effective passes over data had been reached. In all of our experiments, we plot effective data passes[4] vs gradient norm, and plot the central tendency as a solid line and all other executions as a shaded region. Plots with function sub-optimality are also provided in Figure B.1 Section B.3.2, and show much the same relative rankings amongst the methods as the gradient norm plots.

For all methods, we used the default step size. For instance, for SAG and SVRG we use the step size $\frac{1}{L_{\max}}$ where $L_{\max}$ is the largest smoothness constant of $f_i$, for $i = 1, \ldots, n$. This step size is significantly larger than what has been proven to work for SAG and SVRG.[5] Yet despite this, it is the default setting in sklearn's logistic regression solver (Pedregosa et al., 2011), and we also found that it worked well in practice. The other hyperparameter of SVRG is the inner loop size which is set to $n$ throughout our experiments. As for SAN, we set the probability $\pi = \frac{1}{n+1}$ and step size $\gamma = 1$. More details of the experiments are in Section B.3.1. [6]

**Logistic regression with L2 regularization.** We consider L2-regularized logistic regression, i.e. the regularizer is $R(w) = \frac{\lambda}{2}\|w\|^2$ with $\lambda = 1/n$. From Figure 3.1, SAN outperforms SAG and SVRG in all eight datasets, except for `mushrooms`, `ijcnn1` and `covtype`, where SAN remains competitive with SAG or SVRG. Note as well that for reaching an approximate solution at early stage, SAN outperforms SAG and SVRG in all datasets, except for `covtype`. Furthermore, SAN often has a smaller variance compared to SAG and SVRG based on eye-balling the shaded error bars in Figure 3.1 which was produced by multiple executions.

**Logistic regression with pseudo-Huber regularization.** We also tested logistic regression with pseudo-Huber regularizer. The pseudo-Huber regularizer is defined as $R(w) = \lambda \sum_{i=1}^{d} R_i(w_i)$ with $R_i(w_i) = \delta^2\left(\sqrt{1 + \left(\frac{w_i}{\delta}\right)^2} - 1\right)$ and is used to promote the sparsity of the solution (Fountoulakis and Gondzio, 2016). We set $\delta = 1$ and $\lambda = 1/n$. See Section B.3.1 for more properties and interpretations of the pseudo-Huber regularizer. From Figure 3.2, SAN is competitive

[3] All datasets can be found downloaded on https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/. Some of the datasets can be found originally in (Mohammad et al., 2012; Chang and Lin, 2001; Blackard and Dean, 1999; Wang et al., 2012; Lewis et al., 2004; Dua and Graff, 2017).

[4] By effective data passes we mean the number of data access divided by $n$.

[5] SAG has been proven to converge with a step size of $1/16L_{\max}$ (Schmidt et al., 2017) and SVRG provably converges with a step size of $1/10L_{\max}$ and loop size of $m = 10L_{\max}/\mu$ where $\mu$ is the strong convexity parameter $f(w)$ (Johnson and Zhang, 2013).

[6] The code is available on https://github.com/nathansiae/Stochastic-Average-Newton.

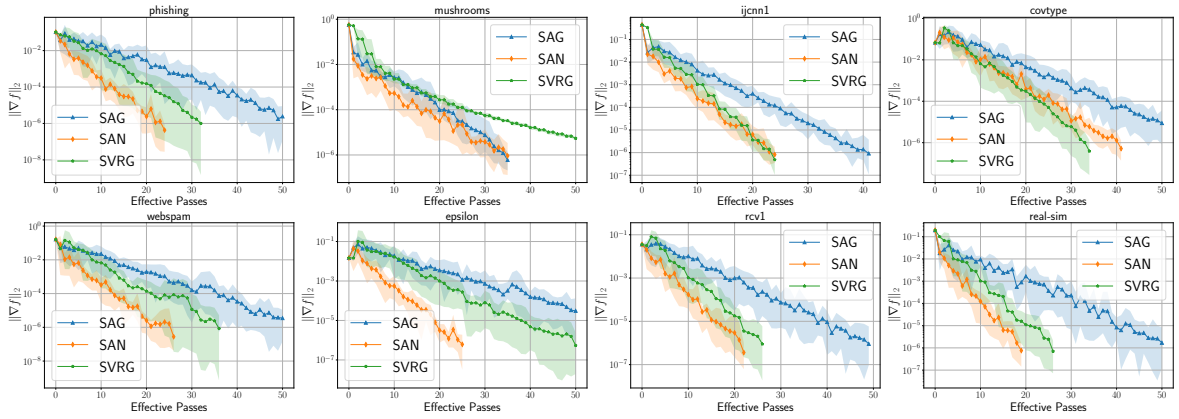


**Figure 3.1** – Logistic regression with L2 regularization.

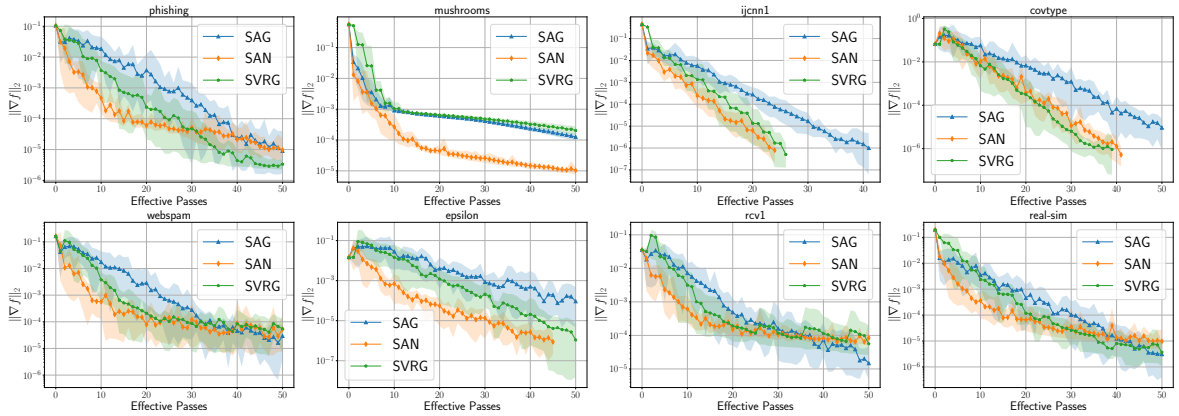


**Figure 3.2** – Logistic regression with pseudo-Huber regularization.

with SAG and SVRG. Changing the regularizer from an L2 to pseudo-Huber has resulted in a slower convergence for all methods, except on the datasets `ijcnn1` and `covtype`. SVRG is notably slower when using the pseudo-Huber regularizer, while SAG is the least affected, and SAN is in between. Besides, SAN again outperforms SAG and SVRG in all datasets, except for `covtype`, for reaching an approximate solution at early stage and has a smaller variance compared to SAG and SVRG.

Overall, these tests confirm that SAN is efficient for a wide variety of datasets and problems. SAN is efficient in terms of effective passes and cost per iteration. It benefits from both using second order information yet still has the same cost $\mathcal{O}(d)$ as the stochastic first order methods. SNM and IQN, the only other methods that fit our stated objective, have a $\mathcal{O}(d^2)$ cost per iteration for L2-regularized GLMs. SNM costs even more for other regularizers and IQN has a $\mathcal{O}(nd^2)$ memory cost. In Section B.3.4, we present experiments comparing SAN/SANA to the SNM and IQN algorithms.

Another advantage of SAN is that it requires no prior knowledge of the datasets nor tuning of the hyperparameters. To show this, we did a grid search over $\pi$ and the step size $\gamma$ of SAN, see

**Table 3.1** – Average cost of one iteration of various stochastic methods applied to GLMs.

| | memory | memory access | data access | computational cost |
|---|---|---|---|---|
| SAN | $\mathcal{O}(nd)$ | $\boldsymbol{\mathcal{O}(d)}$ | $\boldsymbol{\mathcal{O}(1)}$ | $\boldsymbol{\mathcal{O}(d)}^*$ |
| SANA | $\mathcal{O}(nd)$ | $\mathcal{O}(nd)$ | $\boldsymbol{\mathcal{O}(1)}$ | $\mathcal{O}(nd)$ |
| SAG | $\mathcal{O}(nd)$ | $\boldsymbol{\mathcal{O}(d)}$ | $\boldsymbol{\mathcal{O}(1)}$ | $\boldsymbol{\mathcal{O}(d)}$ |
| SVRG | $\boldsymbol{\mathcal{O}(d)}$ | $\boldsymbol{\mathcal{O}(d)}$ | $\boldsymbol{\mathcal{O}(1)}$ | $\boldsymbol{\mathcal{O}(d)}$ |
| SNM | $\mathcal{O}(n+d^2)$ | $\mathcal{O}(d^2)$ | $\boldsymbol{\mathcal{O}(1)}$ | $\mathcal{O}(d^2)^{**}$ |
| IQN | $\mathcal{O}(nd^2)$ | $\mathcal{O}(d^2)$ | $\boldsymbol{\mathcal{O}(1)}$ | $\mathcal{O}(d^2)$ |

*For SAN this $\mathcal{O}(d)$ computational cost is derived when $\pi = \mathcal{O}(1/n)$.

**For SNM this $\mathcal{O}(d^2)$ computational cost only holds for a L2 regularizer.

Tables B.2 and B.3 in Section B.3.3, where we found that SAN performs equally well for a wide range of values of $\pi$ and $\gamma$. Thus for simplicity we set $\pi = \frac{1}{n+1}$ and $\gamma = 1$ in the experiments. In contrast, SAG and SVRG require the computation of $L_{\max}$ and the performance is highly affected by the step size (see Table B.4 and B.5 in Section B.3.3).

However, the downside of SAN is that it stores $nd$ scalars much like SAG/SAGA (Defazio et al., 2014). See Table 3.1 the comparison among different algorithms.

## 3.4 Sketched Newton Raphson with a variable metric

### 3.4.1 Presentation of the SNRVM algorithm

Though our main focus is in solving the function splitting reformulation (3.3) and (3.4), we find that our forthcoming theory holds for a large class of variable metric *Sketched Newton Raphson* methods (Yuan et al., 2022b), of which SAN/SANA are special cases. All proofs are given in Section B.5.

In order to design such method, we first reformulated our original problem (3.1) as a system of nonlinear equations $F(x) = 0$ for a given choice of a smooth map $F : \mathbb{R}^p \longrightarrow \mathbb{R}^m$ and where $p, m \in \mathbb{N}$ are appropriately chosen dimensions, e.g., $p = m = (n+1)d$ in (3.6) for SAN/SANA. We then proposed using a subsampled Newton Raphson method for solving these nonlinear equations. Here we extend this subsampling to make use of any randomized *sketch* of the system. That is, consider a random *sketching* matrix $\mathbf{S}_k \in \mathbb{R}^{m\times\tau}$ sampled from some distribution, where $\tau \in \mathbb{N}$ is significantly smaller than $p$ or $m$. We use this sketching matrix to compress the rows of the Newton system (3.7) at each iteration by left multiplying as follows

$$\mathbf{S}_k^\top F(x^k) + \mathbf{S}_k^\top \nabla F(x^k)^\top (x - x^k) = 0. \tag{3.15}$$

The resulting system has $\tau$ rows and is under-determined. To pick a solution, we use the projection

$$x^{k+1} = \arg\min_{x \in \mathbb{R}^p} \left\| x - x^k \right\|^2 \quad \text{s.t. } \mathbf{S}_k^\top F(x^k) + \mathbf{S}_k^\top \nabla F(x^k)^\top (x - x^k) = 0. \tag{3.16}$$

The method in (3.16) is known as the SNR (*Sketched Newton Raphson*) method (Yuan et al., 2022b). The SNR method affords a lot of flexibility by choosing different distributions for the sketching matrices. Yet it is not flexible enough to include SAN/SANA, since these require projections under a variable metric. To allow for projections under norms other than the L2 norm, we introduce a random positive-definite *metric* matrix $\mathbf{W}_k \in \mathbb{R}^{p \times p}$ that defines the norm under which we project. Introducing as well a damping parameter $\gamma > 0$, as we did for SAN and SANA, we obtain the following method

$$\begin{aligned} \bar{x}^{k+1} &= \operatorname{argmin} \|x - x^k\|^2_{\mathbf{W}_k} \\ &\quad \text{s.t. } \mathbf{S}_k^\top \nabla F(x^k)^\top (x - x^k) = -\mathbf{S}_k^\top F(x^k), \\ x^{k+1} &= (1-\gamma) x^k + \gamma \bar{x}^{k+1}. \end{aligned} \tag{3.17}$$

We call this method the *Sketched Newton Raphson with Variable Metric* (SNRVM for short). The closed form expression[7] for the iterates (3.17) is given by

$$x^{k+1} = x^k - \gamma \mathbf{W}_k^{-1} \nabla F(x^k) \mathbf{S}_k \cdot \left( \mathbf{S}_k^\top \nabla F(x^k)^\top \mathbf{W}_k^{-1} \nabla F(x^k) \mathbf{S}_k \right)^\dagger \mathbf{S}_k^\top F(x^k). \tag{3.18}$$

In Chapter 2 and Appendix A.7, we already mention and show that RSN (Gower et al., 2019a) is a special case of SNRVM. Here, SAN/SANA are also both instances of the SNRVM method by choosing $\mathbf{S}_k$ as a subsampling matrix and $\mathbf{W}_k$ depending on the stochastic Hessian matrices. In Section B.4 we provide a detailed derivation of SAN/SANA as an instance of the SNRVM method.

We assume that at each iteration, the random matrices $(\mathbf{S}_k, \mathbf{W}_k)$ are sampled according to a proper finite distribution $\mathcal{D}_{x^k}$ defined in the following.

**Assumption 3.4** (Proper finite distribution)**.** *For every $x \in \mathbb{R}^p$, there exists $r \in \mathbb{N}$, probabilities $\pi_1, \ldots, \pi_r > 0$ with $\sum_{i=1}^r \pi_i = 1$, and matrices $\{\mathbf{S}_i(x), \mathbf{W}_i(x)\}_{i=1}^r$ s.t. for $i = 1, \ldots, r$, we have*

$$\mathbb{P}_{(\mathbf{S},\mathbf{W}) \sim \mathcal{D}_x} \left[ (\mathbf{S}, \mathbf{W}) = (\mathbf{S}_i(x), \mathbf{W}_i(x)) \right] \; = \; \pi_i.$$

[7] See Lemma B.14 in the appendix for a proof.

The addition of a variable metric to SNR has proven to be very challenging in terms of establishing a convergence theory. The current convergence theory and proofs techniques for SNR in Yuan et al. (2022b) all fail with the addition of a variable metric. This is not so surprising, considering the historic difficulty in developing theory for variable metric methods such as the quasi-Newton methods. Despite the immense practical success of quasi-Newton methods, a meaningful non-asymptotic convergence rate has eluded the optimization community for 70 years, with the first results having only just appeared last year (Rodomanov and Nesterov, 2021a; Rodomanov and Nesterov, 2021c; Rodomanov and Nesterov, 2021b; Jin and Mokhtari, 2022).

In the following sections, we provide a general linear convergence theory for SNRVM in Section 3.4.2 and a more explicit linear convergence rate for SAN and SANA in Section 3.4.3.

### 3.4.2 Linear convergence rates for SNRVM

We start by introducing a technical assumption which guarantees that (3.17) is well defined, which is always true for SAN and SANA.

**Assumption 3.5.** *For every $x \in \mathbb{R}^p$, the matrices $\nabla F(x)^\top \nabla F(x)$ and $\mathbb{E}_{\mathcal{D}_x}[\mathbf{S}\mathbf{S}^\top]$ are invertible, and every matrix $\mathbf{W} \sim \mathcal{D}_x$ is symmetric positive definite.*

**Proposition 3.6.** *Assumptions 3.4 and 3.5 are verified for SAN and SANA, under Assumption 3.2.*

Let us now introduce the surrogate function

$$\hat{f}_k(x) \overset{\text{def}}{=} \frac{1}{2} \|F(x)\|^2_{(\nabla F(x^k)^\top \mathbf{W}_k^{-1} \nabla F(x^k))^\dagger}, \tag{3.19}$$

where $\mathbf{W}_k \sim \mathcal{D}_{x^k}$. This function is closely related to the SNRVM algorithm. Indeed, it is possible to show that $x^{k+1}$ is obtained by minimizing a quadratic approximation of $\hat{f}_k$ along a random subspace (see Lemma B.18). More precisely, $x^{k+1}$ is the solution of

$$\begin{aligned} &\operatorname*{argmin}_{x \in \mathbb{R}^p} \hat{f}_k(x^k) + \langle \nabla \hat{f}_k(x^k), x - x^k \rangle + \frac{1}{2\gamma} \left\| x - x^k \right\|^2_{\mathbf{W}_k} \\ &\text{s.t. } x \in x^k + \mathbf{Im}\left( \mathbf{W}_k^{-1} \nabla F(x^k) \mathbf{S}_k \right) \end{aligned}$$

Our forthcoming Theorem 3.8 shows that $\hat{f}_k(x^k)$ converges linearly to zero in expectation. To achieve this, we need to make an assumption that controls the evolution of $\hat{f}_k$ along the iterations.

**Assumption 3.7.** *There exists $L > 0$ such that, for every $k \in \mathbb{N}$ and every $x \in \mathbb{R}^p$:*

$$\hat{f}_{k+1}(x) \leq \hat{f}_k(x^k) + \langle \nabla \hat{f}_k(x^k), x - x^k \rangle + \frac{L}{2} \left\| x - x^k \right\|_{\mathbf{W}_k}^2 .$$

We now state our core convergence result, which we prove in Appendix B.5.4.

**Theorem 3.8.** *Let Assumptions 3.4, 3.5 and 3.7 hold, and let $\gamma = 1/L$. Let $(\mathbf{S}, \mathbf{W}) \sim \mathcal{D}_x$ and let*

$$\mathbf{H}(x) \overset{def}{=} \mathbb{E}\left[\mathbf{S}\left(\mathbf{S}^\top \nabla F(x)^\top \mathbf{W}^{-1} \nabla F(x) \mathbf{S}\right)^\dagger \mathbf{S}^\top\right],$$
$$\rho(x) \overset{def}{=} \min_{i=1,\ldots,r} \lambda_{\min}^+\left(\mathbf{M}_i \mathbf{H}(x) \mathbf{M}_i^\top\right),$$

*where $\mathbf{M}_i \overset{def}{=} \mathbf{W}_i (x^k)^{-\frac{1}{2}} \nabla F(x)$. Assume that there exists $\rho > 0$ such that $\inf_{k \in \mathbb{N}} \rho(x^k) \geq \rho$ almost surely. It follows that*

$$\mathbb{E}\left[\hat{f}_k(x^k)\right] \;\leq\; (1 - \rho\gamma)^k \, \mathbb{E}\left[\hat{f}_0(x^0)\right] \text{ a.s.}$$

When the metric is constant along iterations and $F(x)$ is a linear function, or equivalently our original problem (3.1) is a quadratic problem, then the SNRVM method (3.18) is known as the sketch-and-project method (Gower and Richtárik, 2015b). In Section B.5.5, we show that Theorem 3.8 when specialized to this case allows us to recover the well known convergence rates for solving linear systems using sketch-and-project.

The convergence result of SNRVM in Theorem 3.8 is for the surrogate function $\hat{f}_k(x^k)$ and the convergence rate $\rho$ is not explicit. Next, we develop a linear convergence theory of SNRVM for $\left\| F(x^k) \right\|^2$ with the explicit linear convergence rate $\rho$.

Indeed, the existence of a lower bound $\rho > 0$ in Theorem 3.8 can be guaranteed, provided that we can uniformly control the matrices $\mathbf{S}$, $\mathbf{W}$ and $\nabla F(x)$. Let us make this more precise:

**Assumption 3.9.** *Assumption 3.5 holds, $m = p$ and $F$ is injective. We assume that there exists a set $\Omega \subset \mathbb{R}^p$ and constants $\mu_W, L_W, \bar{\mu}_S, L_S, \mu_{\nabla F}, L_{\nabla F}$ in $(0, +\infty)$ such that, for all $x \in \Omega$, for all $(\mathbf{S}, \mathbf{W}) \sim \mathcal{D}_x$,*

$$\text{spec}(\mathbf{W}) \subset [\mu_W, L_W], \quad \sigma(\nabla F(x)) \subset [\mu_{\nabla F}, L_{\nabla F}],$$

$$\bar{\mu}_S \leq \lambda_{min}\left(\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right]\right), \quad \|\mathbf{S}\mathbf{S}^\top\| \leq L_S.$$

*where* $\mathrm{spec}(M)$ *(resp.* $\sigma(M)$*) denote the set of eigenvalues (resp. singular values) of a square matrix* $M$*.*

This assumption is typically verified on bounded sets, if the matrices $\mathbf{W}, \mathbf{S}, \nabla F(x)$ enjoy some sort of continuity with respect to $x$ (we detail this argument for SAN in Section B.5.7 in the Appendix). Now we can state our general linear convergence theory of SNRVM in terms of $F(x^k)$ itself, instead of the surrogate $\hat{f}_k(x^k)$, with explicit rates.

**Theorem 3.10.** *Let Assumptions 3.4, 3.5, 3.7, and 3.9 hold, and let* $\gamma = 1/L$*. Let* $\{x^k\}_{k\in\mathbb{N}}$ *be generated by the SNRVM algorithm, and suppose that* $x^k \in \Omega$ *almost surely. Then for all* $k \in \mathbb{N}$ *we have that*

$$\mathbb{E}\left[\|F(x^k)\|^2\right] \leq C(1-\gamma\rho)^k \quad \textit{almost surely,}$$

*with* $\rho = \frac{\mu^2_{\nabla F}}{L^2_{\nabla F}} \frac{\mu_W}{L_W} \frac{\bar{\mu}_S}{L_S}$*, and* $C = 2\mathbb{E}\left[\hat{f}_0(x^0)\right] \frac{L^2_{\nabla F}}{\mu_W}$*.*

### 3.4.3 Linear convergence rates for SAN and SANA

Theorem 3.10 provides a general convergence theory for SNRVM. In this section, when specialized to SAN and SANA, we are able to get more insightful convergence results.

Indeed, Assumption 3.9 can be ensured for SAN and SANA under reasonable assumptions on the regularity of the functions $f_i$, in the following sense:

**Assumption 3.11.** *There exists* $0 < \mu_f \leq L_f$ *such that for all* $i \in \{1, \ldots, n\}$*, the function* $f_i : \mathbb{R}^d \longrightarrow \mathbb{R}$ *is of class* $C^2$*,* $\mu_f$*-strongly convex and has a* $L_f$*-Lipschitz continuous gradient.*

The next Theorem, which is our main convergence result for SAN and SANA, describes linear rates for the iterates $(w^k, \alpha_1^k, \ldots, \alpha_n^k)$ themselves:

**Theorem 3.12.** *Let Assumptions 3.7 and 3.11 hold. Let* $\{x^k\}_{k\in\mathbb{N}}$ *be a sequence generated by SAN with* $\pi = 1/(n+1)$*, or by SANA, with* $\gamma = 1/L$*. Let* $w^* = \mathrm{argmin}\ f$*. Then for every* $k \in \mathbb{N}$*:*

$$\mathbb{E}\left[\|w^k - w^*\|^2\right] + \sum_{i=1}^n \mathbb{E}\left[\|\alpha_i^k - \nabla f_i(w^*)\|^2\right] \leq C(1-\gamma\rho)^k$$

*holds almost surely, where we can take*

$$\rho = \frac{\min\{1, \mu_f^3\}}{14n^3(2+L_f^2)^2 \max\{1, L_f^3\}}, \qquad C = 18n^2 \frac{\max\{1, L_f^2\}(2+L_f^2)^2}{\min\{1, \mu_f^3\}} \mathbb{E}\left[\hat{f}_0(x^0)\right].$$

The resulting linear rate of convergence $\rho$ in Theorem 3.12 depends on $n$ and converges to 1 as $n$ goes to infinity. This is not surprising, since it is also the case for variance reduced methods such as SAGA (Defazio et al., 2014) and SVRG (Johnson and Zhang, 2013). We also note that our theoretical rate has a much worse dependence in $\mu_f$, $L_f$ and $n$ than the rates of SVRG and SAGA, because of the presence of exponents greater than 1. This might suggest that our analysis is not tight: indeed we observed empirically that SAN performs as well as SVRG and SAG, even in a regime where $n$ is large and the problem is severely ill-conditioned (see Table B.1 for more details).

## 3.5 Discussion

In this chapter, we introduced the use of a subsampled Newton Raphson method applied to a specific function splitting problem as a tool for designing new incremental Newton methods. We showcase this by developing SAN, an average Newton method that is empirically highly competitive as compared to variance reduced gradient methods, and does not require parameter tuning. Further venues of investigation include:

- Improving our theoretical analysis, to obtain rates that better matches the ones of usual variance reduced methods, motivated by our numerical experiments.
- Leveraging SNRVM's structure to design a more efficient variant of SAN including mini-batching. This should be simple thanks to our function splitting point of view, as we would simply sample many rows at once in (3.5).
- Exploring different sketching techniques together with original and alternative splitting point of views to design methods that have not been discovered yet.

# Part II

# Finite Time Analysis of Policy Gradient Methods in Reinforcement Learning

# Chapter 4

# A General Sample Complexity Analysis of Vanilla Policy Gradient

In this chapter, we adapt recent tools developed for the analysis of Stochastic Gradient Descent (SGD) in non-convex optimization to obtain convergence and sample complexity guarantees for the vanilla policy gradient (PG). Our only assumptions are that the expected return is smooth w.r.t. the policy parameters, that its $H$-step truncated gradient is close to the exact gradient, and a certain *ABC assumption*. This assumption requires the second moment of the estimated gradient to be bounded by $A \geq 0$ times the suboptimality gap, $B \geq 0$ times the norm of the full batch gradient and an additive constant $C \geq 0$, or any combination of aforementioned. We show that the ABC assumption is more general than the commonly used assumptions on the policy space to prove convergence to a stationary point. We provide a single convergence theorem that recovers the $\widetilde{\mathcal{O}}(\epsilon^{-4})$ sample complexity of PG to a stationary point. Our results also affords greater flexibility in the choice of hyper parameters such as the step size and the batch size $m$, including the single trajectory case (i.e., $m = 1$). When an additional *relaxed weak gradient domination* assumption is available, we establish a novel global optimum convergence theory of PG with $\widetilde{\mathcal{O}}(\epsilon^{-3})$ sample complexity. We then instantiate our theorems in different settings, where we both recover existing results and obtain improved sample complexity, e.g., $\widetilde{\mathcal{O}}(\epsilon^{-3})$ sample complexity for the convergence to the global optimum for Fisher-non-degenerated parametrized policies. [1]

## Contents



[1] This chapter is based on an article published in the proceedings of the $25^{th}$ International Conference on Artificial Intelligence and Statistics (AISTATS 2022) (Yuan et al., 2022a).

## 4.1 Introduction

Policy gradient (PG) is one of the most popular reinforcement learning (RL) methods for computing policies that maximize long-term rewards (Williams, 1992; Sutton et al., 2000; Baxter and Bartlett, 2001). The success of PG methods is due to their simplicity and versatility, as they can be readily implemented to solve a wide range of problems (including non-Markov and partially-observable environments) and they can be effectively paired with other techniques to obtain more sophisticated algorithms such as the actor-critic (Konda and Tsitsiklis, 2000; Mnih et al., 2016), natural PG (Kakade, 2001), natural actor-critic (Peters and Schaal, 2008a; Bhatnagar et al., 2009), policy mirror descent (Tomar et al., 2022; Vaswani et al., 2022), trust-region based variants (Schulman et al., 2015; Schulman et al., 2017; Shani et al., 2020), and variance-reduced methods (Papini et al., 2018; Shen et al., 2019; Xu et al., 2020b; Yuan et al., 2020; Huang et al., 2020; Pham et al., 2020; Yang et al., 2022; Huang et al., 2022). Unlike value-based methods, a solid theoretical understanding of even the "vanilla" PG has long been elusive. Recently, a more complete theory of PG has been derived by leveraging the RL structure of the problem together with tools from convex and non-convex optimization (see Appendix C.1 for a thorough review).

In this chapter, we first focus on the sample complexity of PG for reaching a FOSP (first-order stationary point). We show how PG can be analysed under a very general assumption on the second moment of the estimated gradient called the *ABC* assumption, which includes most of the bounded gradient type assumptions as a special case. Our first contribution is convergence guarantees and sample complexity for both REINFORCE (Williams, 1992) and GPOMDP (Sutton et al., 2000; Baxter and Bartlett, 2001) under the ABC and assumptions on the smoothness of the expected return and on its truncated gradient. Our sample complexity analysis recovers both the well known $\mathcal{O}(\epsilon^{-2})$ iteration complexity of exact PG and the $\widetilde{\mathcal{O}}(\epsilon^{-4})$ sample complexity of REINFORCE and GPOMDP under weaker assumptions than had previously been explored (Zhang et al., 2020b; Liu et al., 2020; Xiong et al., 2021). Furthermore, our analysis is less restrictive when it comes to the hyper-parameter choices. In fact, our results allow for a wide range of step sizes and place almost no restriction on the batch size $m$, even allowing for single trajectory sampling ($m = 1$), which is uncommon in the literature. The generality of our assumption allows us to unify much of the fragmented results in the literature under one guise. Indeed, we show that the analysis of Lipschitz and smooth policies, Gaussian polices, softmax tabular polices with or without a log barrier or an entropy regularizer are all special cases of our general analysis (see hierarchy diagram further down in Figure 4.1).

Recently, there has also been much work on establishing the convergence of PG to a global optimum (i.e., the best-in-class policy). This usually requires more restrictive assumptions (Zhang et al., 2020a; Zhang et al., 2021a), specific RL settings (e.g., linear-quadratic regulator (Fazel et al., 2018), tabular (Agarwal et al., 2021) and softmax tabular policy (Mei et al., 2020)), and it

is often limited to exact PG. Inspired by the sample complexity analysis of the stochastic PG for the global optimum in Liu et al. (2020) and Ding et al. (2022), our second contribution is to establish a novel global optimum convergence theory of PG when an additional *relaxed weak gradient domination* assumption is available. Our sample complexity analysis recovers the well known $\mathcal{O}(\epsilon^{-1})$ iteration complexity of the exact PG with the softmax tabular policy (Mei et al., 2020) as a special case and obtains a new improved $\widetilde{\mathcal{O}}(\epsilon^{-3})$ sample complexity compared to $\widetilde{\mathcal{O}}(\epsilon^{-4})$ in Liu et al. (2020), with the Fisher-non-degenerate parametrized policy (Liu et al., 2020; Ding et al., 2022) as a special case. We also establish even faster global optimum convergence theory when replacing the relaxed weak gradient domination assumption by gradient domination in Appendix C.8. As a special case, we recover the well known linear convergence rate of the exact PG with the softmax tabular policy with entropy regularization (Mei et al., 2019) in Appendix C.8. Table 4.1 provides a complete overview of our results.

## 4.2 Preliminaries

**Markov decision process (MDP).** We consider a MDP $M = \{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \rho\}$, where $\mathcal{S}$ is a state space; $\mathcal{A}$ is an action space; $\mathcal{P}$ is a Markovian transition model, where $\mathcal{P}(s' \mid s, a)$ is the transition density from state $s$ to $s'$ under action $a$; $\mathcal{R}$ is the reward function, where $\mathcal{R}(s, a) \in [-\mathcal{R}_{\max}, \mathcal{R}_{\max}]$ is the bounded reward for state-action pair $(s, a)$ ; $\gamma \in [0, 1)$ is the discounted factor; and $\rho$ is the initial state distribution. The agent's behaviour is modelled as a policy $\pi \in \Delta(\mathcal{A})^{\mathcal{S}}$, where $\pi(a \mid s)$ is the density of the distribution over actions at state $s \in \mathcal{S}$. We consider the infinite-horizon discounted setting.

Let $p(\tau \mid \pi)$ be the probability density of a single trajectory $\tau$ being sampled from $\pi$, that is

$$p(\tau \mid \pi) = \rho(s_0) \prod_{t=0}^{\infty} \pi(a_t \mid s_t) \mathcal{P}(s_{t+1} \mid s_t, a_t). \tag{4.1}$$

With a slight abuse of notation, let $\mathcal{R}(\tau) = \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t)$ be the total discounted reward accumulated along trajectory $\tau$. We define the expected return of $\pi$ as

$$J(\pi) \stackrel{\text{def}}{=} \mathbb{E}_{\tau \sim p(\cdot \mid \pi)} \left[ \mathcal{R}(\tau) \right]. \tag{4.2}$$

**Policy gradient.** We introduce a set of parametrized policies $\{\pi_\theta : \theta \in \mathbb{R}^d\}$, with the assumption that $\pi_\theta$ is differentiable w.r.t. $\theta$. We denote $J(\theta) = J(\pi_\theta)$ and $p(\tau \mid \theta) = p_\theta(\tau) = p(\tau \mid \pi_\theta)$. In general, $J(\theta)$ is a non-convex function. The PG methods use gradient ascent in the space of $\theta$ to find the policy that maximizes the expected return, i.e., $\theta^* \in \arg\sup_{\theta \in \mathbb{R}^d} J(\theta)$. We denote the *optimal expected return* as $J^* \stackrel{\text{def}}{=} J(\theta^*)$.

**Table 4.1** – Overview of different convergence results for vanilla PG methods. The darker cells contain our new results. The light cells contain previously known results that we recover as special cases of our analysis, and extend the permitted parameter settings. White cells contain existing results that we could not recover under our general analysis.

| Guarantee* | Setting** | Reference (our results in bold) | Bound | Remarks |
|---|---|---|---|---|
| Sample complexity of stochastic PG for FOSP | ABC | **Theorem 4.4** | $\widetilde{\mathcal{O}}(\epsilon^{-4})$ | Weakest assumption |
| | E-LS | Papini (2020) **Corollary 4.14** | $\widetilde{\mathcal{O}}(\epsilon^{-4})$ | Weaker assumption; Wider range of parameters; Recover $\mathcal{O}(\epsilon^{-2})$ for exact PG; Improved smoothness constant |
| Sample complexity of stochastic PG for GO | ABC + PL | **Theorem C.22** | $\widetilde{\mathcal{O}}(\epsilon^{-1})$ | Recover linear convergence for the exact PG |
| | ABC + (4.17) | **Theorem C.8** | $\widetilde{\mathcal{O}}(\epsilon^{-3})$ | Recover $\mathcal{O}(\epsilon^{-1})$ for the exact PG |
| | E-LS + FI + compatible | **Corollary 4.21** | $\widetilde{\mathcal{O}}(\epsilon^{-3})$ | Improved by $\epsilon$ compared to Corollary 4.14 |
| Sample complexity of stochastic PG for AR | ABC + (4.17) | **Corollary C.7** | $\widetilde{\mathcal{O}}(\epsilon^{-4})$ | Weakest assumption |
| | E-LS + FI + compatible | Liu et al. (2020) **Corollary C.17** | $\widetilde{\mathcal{O}}(\epsilon^{-4})$ | Weaker assumption; Wider range of parameters |
| | Softmax + log barrier (4.31) | Zhang et al. (2021b) **Corollary 4.18** | $\widetilde{\mathcal{O}}(\epsilon^{-6})$ | Constant step size; Wider range of parameters; Extra phased learning step unnecessary |
| Iteration complexity of the exact PG for GO | Softmax + log barrier (4.31) | Agarwal et al. (2021) **Corollary C.14** | $\mathcal{O}(\epsilon^{-2})$ | Improved by $1-\gamma$ |
| | Softmax (4.28) | Mei et al. (2020) **Theorem C.8** | $\mathcal{O}(\epsilon^{-1})$ | |
| | Softmax + entropy (C.84) | Mei et al. (2020) **Theorem C.22** | linear | |
| | LS + bijection + PPG | Zhang et al. (2020a) | $\mathcal{O}(\epsilon^{-1})$ | |
| | Tabular + PPG | Xiao (2022) | $\mathcal{O}(\epsilon^{-1})$ | |
| | LQR | Fazel et al. (2018) | linear | |

* **Type of convergence.** *PG*: policy gradient; *FOSP*: first-order stationary point; *GO*: global optimum; *AR*: average regret to the global optimum.

** **Setting.** *bijection*: Assumption 1 in Zhang et al. (2020a) about occupancy distribution; *PPG*: analysis also holds for the projected PG; *Tabular*: direct parametrized policy; *LQR*: linear-quadratic regulator.

The gradient $\nabla J(\theta)$ of the expected return has the following structure

$$\begin{aligned}\nabla J(\theta) &= \int \mathcal{R}(\tau)\nabla p(\tau \mid \theta)d\tau \\ &= \int \mathcal{R}(\tau)\left(\nabla p(\tau \mid \theta)/p(\tau \mid \theta)\right)p(\tau \mid \theta)d\tau \\ &= \mathbb{E}_{\tau\sim p(\cdot|\theta)}\left[\mathcal{R}(\tau)\nabla \log p(\tau \mid \theta)\right]\end{aligned}$$

$$\overset{(4.1)}{=} \mathbb{E}_\tau \left[ \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \sum_{t'=0}^{\infty} \nabla_\theta \log \pi_\theta(a_{t'} \mid s_{t'}) \right]. \tag{4.3}$$

In practice, we cannot compute this full gradient, since computing the above expectation requires averaging over all possible trajectories $\tau \sim p(\cdot \mid \theta)$. We resort to an empirical estimate of the gradient by sampling $m$ truncated trajectories $\tau_i = \left(s_0^i, a_0^i, r_0^i, s_1^i, \cdots, s_{H-1}^i, a_{H-1}^i, r_{H-1}^i\right)$ with $r_t^i = \mathcal{R}(s_t^i, a_t^i)$ obtained by executing $\pi_\theta$ for a given fixed horizon $H \in \mathbb{N}$. The resulting gradient estimator is

$$\widehat{\nabla}_m J(\theta) \;=\; \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{H-1} \gamma^t \mathcal{R}(s_t^i, a_t^i) \cdot \sum_{t'=0}^{H-1} \nabla_\theta \log \pi_\theta(a_{t'}^i \mid s_{t'}^i). \tag{4.4}$$

The estimator (4.4) is known as the REINFORCE gradient estimator (Williams, 1992).

The REINFORCE estimator can be simplified by leveraging the fact that future actions do not depend on past rewards. This leads to the alternative formulation of the full gradient

$$\nabla J(\theta) \;=\; \mathbb{E}_\tau \left[ \sum_{t=0}^{\infty} \left( \sum_{k=0}^{t} \nabla_\theta \log \pi_\theta(a_k \mid s_k) \right) \gamma^t \mathcal{R}(s_t, a_t) \right], \tag{4.5}$$

which leads to the following estimate of the gradient known as GPOMDP (Baxter and Bartlett, 2001)

$$\widehat{\nabla}_m J(\theta) \;=\; \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{H-1} \left( \sum_{k=0}^{t} \nabla_\theta \log \pi_\theta(a_k^i \mid s_k^i) \right) \gamma^t \mathcal{R}(s_t^i, a_t^i). \tag{4.6}$$

Following the same argument of that future actions do not depend on past rewards, we can also simplify the REINFORCE estimator by removing the rewards from previous states. This leads to the third alternative formulation of the full gradient

$$\nabla J(\theta) \;=\; \mathbb{E}_\tau \left[ \sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t \mid s_t) \sum_{t'=t}^{\infty} \gamma^{t'} \mathcal{R}(s_{t'}, a_{t'}) \right]. \tag{4.7}$$

From (4.7), one can suggest the gradient estimator $\widehat{\nabla}_m J(\theta)$ as

$$\widehat{\nabla}_m J(\theta) \;=\; \frac{1}{m} \sum_{i=1}^{m} \sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(a_t^i \mid s_t^i) \cdot \sum_{t'=t}^{H-1} \gamma^{t'} \mathcal{R}(s_{t'}^i, a_{t'}^i), \tag{4.8}$$

known as the policy gradient theorem (PGT) (Sutton et al., 2000). It has been shown by Peters and Schaal (2008b) that PGT (4.8) is equivalent to GPOMDP (4.6). Due to their equivalence, we refer to them interchangeably. A derivation of (4.5) and (4.7) is provided in Appendix C.2

(Lemma C.3) for completeness. Notice that PGT has also an action value expression (5.8) which will be presented in Chapter 5.

Both REINFORCE and GPOMDP are the truncated versions of unbiased gradient estimators and they are unbiased estimates of the gradient of the truncated expected return

$$J_H(\theta) \overset{\text{def}}{=} \mathbb{E}_\tau \left[ \sum_{t=0}^{H-1} \gamma^t \mathcal{R}(s_t, a_t) \right].^{2} \tag{4.9}$$

Equipped with gradient estimators defined as either the exact full gradient (4.3), (4.5) and (4.7) or the stochastic PG estimator (4.4), (4.6) or (4.8), vanilla policy gradient updates the policy parameters as follows

$$\theta_{t+1} = \theta_t + \eta_t \widehat{\nabla}_m J(\theta_t) \tag{4.10}$$

where $\eta_t > 0$ is the step size at the $t$-th iteration. See also Algorithm 4.

**Algorithm 4:** Vanilla policy gradient

**Input:** Mini-batch size $m$, step size $\eta_0 > 0$

Initialize $\theta_0 \in \mathbb{R}^d$ **for** $t = 0$ ***to*** $T - 1$ **do**

- Sample $m$ trajectories following policy $\pi_{\theta_t}$ from the MDP
- Compute the policy gradient estimator $\widehat{\nabla}_m J(\theta_t)$
- Update $\theta_{t+1} = \theta_t + \eta_t \widehat{\nabla}_m J(\theta_t)$ and $\eta_t$

## 4.3 Non-convex optimization under ABC assumption

### 4.3.1 First-order stationary point convergence

We use $\widehat{\nabla}_m J(\theta)$ to denote the unbiased policy gradient estimator of $\nabla J_H(\theta)$ used in (4.10). It can be the exact gradient $\nabla J(\theta)$ when $H = m = \infty$, or the truncated gradient estimators in (4.4) or (4.6). All our forthcoming analysis relies on the following common assumptions.

**Assumption 4.1** (Smoothness). *There exists $L > 0$ such that, for all $\theta, \theta' \in \mathbb{R}^d$, we have*

$$\left| J(\theta') - J(\theta) - \langle \nabla J(\theta), \theta' - \theta \rangle \right| \leq \frac{L}{2} \left\| \theta' - \theta \right\|^2. \tag{4.11}$$

[2] We allow $H$ to be infinity so that $J_\infty(\cdot) = J(\cdot)$.

**Assumption 4.2** (Truncation). *There exists $D, D' > 0$ such that, for all $\theta \in \mathbb{R}^d$, we have*

$$|\langle \nabla J_H(\theta), \nabla J_H(\theta) - \nabla J(\theta) \rangle| \leq D\gamma^H, \tag{4.12}$$

$$\|\nabla J_H(\theta) - \nabla J(\theta)\| \leq D'\gamma^H. \tag{4.13}$$

We recall that given the boundedness of the reward function, we have $|J(\theta) - J_H(\theta)| \leq \frac{\mathcal{R}_{\max}}{1-\gamma}\gamma^H$ by the definition of $J(\cdot)$ and $J_H(\cdot)$. As such, when $H$ is large, the difference between $J(\theta)$ and $J_H(\theta)$ is negligible. However, Assumption 4.2 is still necessary, since in our analysis we first prove that $\|\nabla J_H(\theta)\|^2$ is small, and then rely on (4.13) to show that $\|\nabla J(\theta)\|^2$ is also small.

**Remark.** Assumption 4.2 might not be necessary if we replace the truncated estimator to the unbiased estimator where the sampled trajectories have a stochastic length of the horizon driven by the discounted factor. The alternative samplers are used later in Chapter 5 and presented in Algorithms 13 and 14 in Appendix D.3.

We also make use of the *ABC* assumption (Polyak and Tsypkin (1973, equation (3.1)) and Khaled and Richtárik (2023, Assumption 2)[3]) which bounds the second moment of the norm of the gradient estimators using the norm of the truncated full gradient, the suboptimality gap and an additive constant.

**Assumption 4.3** (ABC). *There exists $A, B, C \geq 0$ such that the policy gradient estimator satisfies*

$$\mathbb{E}\left[\left\|\widehat{\nabla}_m J(\theta)\right\|^2\right] \leq 2A(J^* - J(\theta)) + B\|\nabla J_H(\theta)\|^2 + C, \tag{ABC}$$

*for all $\theta \in \mathbb{R}^d$.*

The ABC assumption effectively summarizes a number of popular and more restrictive assumptions commonly used in non-convex optimization. Indeed, the bounded variance of the stochastic gradient assumption (Ghadimi and Lan, 2013), the gradient confusion assumption (Sankararaman et al., 2020), the sure-smoothness assumption (Lei et al., 2020), the convex expected smoothness assumption (Gower et al., 2019b; Gower et al., 2021b) and different variants of strong growth assumptions proposed by Schmidt and Roux (2013) and Vaswani et al. (2019a) and Bottou et al. (2018) can all be seen as specific cases of Assumption 4.3. The

[3] While Khaled and Richtárik (2023) refer to this assumption as *expected smoothness*, we prefer the alternative name ABC to avoid confusion with the smoothness of $J$.

ABC assumption has been shown to be the weakest among all existing assumptions to provide convergence guarantees for SGD for the minimization of non-convex smooth functions. A more detailed discussion of the assumption for non-convex optimization convergence theory can be found in Theorem 1 in Khaled and Richtárik (2023).

We state our main convergence theorem, that we will then develop into several corollaries.

**Theorem 4.4.** *Suppose that Assumption 4.1, 4.2 and 4.3 hold. Consider the iterates $\theta_t$ of the PG method (4.10) with stepsize $\eta_t = \eta \in \left(0, \frac{2}{LB}\right)$ where $B = 0$ means that $\eta \in (0, \infty)$. Let $\delta_0 \overset{def}{=} J^* - J(\theta_0)$. It follows that*

$$\min_{0 \leq t \leq T-1} \mathbb{E}\left[\|\nabla J(\theta_t)\|^2\right] \leq \frac{2\delta_0(1 + L\eta^2 A)^T}{\eta T(2 - LB\eta)} + \frac{LC\eta}{2 - LB\eta} + \left(\frac{2D(3 - LB\eta)}{2 - LB\eta} + D'^2\gamma^H\right)\gamma^H. \tag{4.14}$$

*In particular if $A = 0$, we have*

$$\mathbb{E}\left[\|\nabla J(\theta_U)\|^2\right] \leq \frac{2\delta_0}{\eta T(2 - LB\eta)} + \frac{LC\eta}{2 - LB\eta} + \left(\frac{2D(3 - LB\eta)}{2 - LB\eta} + D'^2\gamma^H\right)\gamma^H, \tag{4.15}$$

*where $\theta_U$ is uniformly sampled from $\{\theta_0, \cdots, \theta_{T-1}\}$.*

Theorem 4.4 provides a general characterization of the convergence of PG as a function of all the constants involved in the assumptions on the problem and the policy gradient estimator. Refer to Appendix C.1.1 for a discussion comparing the technical aspects of this result compared to Khaled and Richtárik (2023). From (4.14) we derive the sample complexity as follows.

**Corollary 4.5.** *Consider the setting of Theorem 4.4. Given $\epsilon > 0$, let $\eta = \min\left\{\frac{1}{\sqrt{LAT}}, \frac{1}{LB}, \frac{\epsilon}{2LC}\right\}$ and the horizon $H = \mathcal{O}(\log \epsilon^{-1})$. If the number of iterations $T$ satisfies*

$$T \geq \frac{12\delta_0 L}{\epsilon^2} \max\left\{B, \frac{12\delta_0 A}{\epsilon^2}, \frac{2C}{\epsilon^2}\right\}, \tag{4.16}$$

*then $\min_{0 \leq t \leq T-1} \mathbb{E}\left[\|\nabla J(\theta_t)\|^2\right] = \mathcal{O}(\epsilon^2)$.*

Despite the generality of the ABC assumption, Corollary 4.5 recovers the best known iteration complexity for vanilla PG in several well-known cases.

First, (4.16) recovers the $\mathcal{O}(\epsilon^{-2})$ iteration complexity of the exact gradient method as a special case. To see this, let $H = m = \infty$ and $\widehat{\nabla}_m J(\theta) = \nabla J(\theta)$ in (4.10), thus Assumption 4.2 and 4.3 hold automatically with $A = C = D = D' = 0$ and $B = 1$. By (4.16), this shows that

for any policy and MDP that satisfy the smoothness property (Assumption 4.1), the exact full PG converges to a $\epsilon$-FOSP in $T = \mathcal{O}(\epsilon^{-2})$ iterations. This is the state-of-the-art convergence rate for the exact gradient descent on non-convex objectives without any other assumptions (Beck, 2017).

Second, we recover sample complexity for stochastic vanilla PG. From Corollary 4.5, notice that there is no restriction on the batch size $m$. By choosing $m = \mathcal{O}(1)$, equation (4.16) shows that with $TH = \widetilde{\mathcal{O}}(\epsilon^{-4})$ samples (i.e., single-step interaction with the environment and single sampled trajectory per iteration), the vanilla PG either with updates (4.4) or (4.6) is guaranteed to converge to an $\epsilon$-stationary point. Our sample complexity matches the results of Papini (2020), Zhang et al. (2020b), Liu et al. (2020), and Xiong et al. (2021), but improve upon them in generality, i.e., by recovering the exact PG analysis, providing wider range of parameter choices and using the weaker ABC assumption (see Section 4.4.1 for more details).

### 4.3.2 Global optimum convergence under relaxed weak gradient domination

In this section, we present a global optimum convergence of the vanilla PG when the relaxed weak gradient domination assumption is available, in addition to the (ABC) assumption.

**Assumption 4.6** (Relaxed weak gradient domination). *We say that $J$ satisfies the relaxed weak gradient domination condition if for all $\theta \in \mathbb{R}^d$, there exists $\mu > 0$ and $\epsilon' \geq 0$ such that*

$$\epsilon' + \|\nabla J_H(\theta)\| \;\geq\; 2\sqrt{\mu}\,(J^* - J(\theta))\,. \tag{4.17}$$

The relaxed weak gradient domination is an extension of weak gradient domination[4] (Agarwal et al., 2021; Mei et al., 2020; Mei et al., 2021) where $\epsilon' = 0$. Equipped with this assumption, we obtain an average regret convergence as a direct consequence of Corollary 4.5 (see Corollary C.7 in Appendix C.3.3). With the same assumption, we also obtain a new global optimum convergence guarantee (see Theorem C.8 in Appendix C.3.4 for the full details).

**Corollary 4.7.** *Consider the setting of Theorem C.8. Given $\epsilon > 0$, let the horizon $H = \mathcal{O}(\log \epsilon^{-1})$. If $\epsilon' = 0$, we choose the number of iterations $T = \mathcal{O}(\epsilon^{-3})$; if $\epsilon' > 0$, we choose $T = \mathcal{O}((\epsilon')^{-2}\epsilon^{-1})$. Then $\min_{t\in\{0,1,\cdots,T\}} J^* - \mathbb{E}\left[J(\theta_t)\right] \leq \mathcal{O}(\epsilon) + \mathcal{O}(\epsilon')$.*

[4] The weak gradient domination is the special case of the Kurdyka-Łojasiewicz (KŁ) condition with KŁ exponent 1 (Kurdyka, 1998).

Consequently, when $\epsilon' = \Theta(\epsilon)$ we have that the complexity of PG to reach a global optimum is $\mathcal{O}(\epsilon^{-3})$. Thus the relaxed weak gradient domination has afforded us a factor of $\epsilon^{-1}$ improvement as compared to the $\mathcal{O}(\epsilon^{-4})$ complexity in Corollary 4.5. The relaxed weak gradient domination is an assumption that is unique to PG methods. In Section 4.4.3, we show that the Fisher-non-degenerate parametrized policy satisfies this assumption.

## 4.4 Applications

In this section we show how the ABC assumption can be used to unify many of the current assumptions used in the literature. In Figure 4.1 we collect all these special cases in a hierarchy tree. Then for each special case we give the sample complexity of PG as a corollary of Theorem 4.4. Each of our corollaries match the best known results in these special cases, while also providing a wider range of parameter choices and, in some cases, improving the dependency on some terms in the bound (e.g., the discount factor $\gamma$). Finally, we show that the relaxed weak gradient domination assumption holds for Fisher-non-degenerate parametrized policies, thus leading to new improved sample complexity result for this setting.

### 4.4.1 Expected Lipschitz and smooth policies

We consider the **expected Lipschitz and smooth policy** (E-LS) assumptions proposed by Papini et al. (2022)[5].

**Assumption 4.8** (E-LS)**.** *There exists constants $G, F > 0$ such that for every state $s \in \mathcal{S}$, the expected gradient and Hessian of $\log \pi_\theta(\cdot \mid s)$ satisfy*

$$\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\|\nabla_\theta \log \pi_\theta(a \mid s)\|^2\right] \leq G^2, \tag{4.18}$$

$$\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\left\|\nabla^2_\theta \log \pi_\theta(a \mid s)\right\|\right] \leq F. \tag{4.19}$$

We call the above *Expected* Lipschitz and Smooth (E-LS), due to the expectation of $a \sim \pi_\theta(\cdot \mid s)$, in contrast to the more restrictive **Lipschitz and smooth policy** (LS) assumption

$$\|\nabla_\theta \log \pi_\theta(a \mid s)\| \leq G \quad \text{and} \quad \left\|\nabla^2_\theta \log \pi_\theta(a \mid s)\right\| \leq F, \tag{LS}$$

[5]While Papini et al. (2022) refers to this assumption as *smoothing policy*, we prefer the alternative name expected Lipschitz and smooth policy, as they not only induce the smoothness of $J$ (see Lemma 4.11), but also the Lipschitzness (see Lemma C.9). In Papini et al. (2022), they also assume that $\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\|\nabla_\theta \log \pi_\theta(a \mid s)\|\right]$ is bounded, while it is a direct consequence of (4.18) by Cauchy-Schwarz inequality.

for all $(s, a) \in \mathcal{S} \times \mathcal{A}$. The (LS) assumption is widely adopted in the analysis of vanilla PG (Zhang et al., 2020b) and variance-reduced PG methods, e.g. Shen et al. (2019), Xu et al. (2020a), Xu et al. (2020b), Yuan et al. (2020), Huang et al. (2020), Pham et al. (2020), Liu et al. (2020), and Zhang et al. (2021a). It is also a relaxation of the element-wise boundness of $\left|\frac{\partial}{\partial\theta_i}\log\pi_\theta(a \mid s)\right|$ and $\left|\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log\pi_\theta(a \mid s)\right|$ assumed by Pirotta et al. (2015) and Papini et al. (2018)

#### 4.4.1.1 Expected Lipschitz and smooth policy is a special case of ABC

In the following lemma we show that (E-LS) implies the ABC assumption.

**Lemma 4.9.** *Under Assumption 4.8, consider a truncated gradient estimator defined either in (4.4) or (4.6). Assumption 4.3 holds with $A = 0, B = 1 - \frac{1}{m}$ and $C = \frac{\nu}{m}$, that is,*

$$\mathbb{E}\left[\left\|\widehat{\nabla}_m J(\theta)\right\|^2\right] \leq \left(1 - \frac{1}{m}\right)\|\nabla J_H(\theta)\|^2 + \frac{\nu}{m}, \tag{4.20}$$

*where $m$ is the mini-batch size, and $\nu = \frac{HG^2\mathcal{R}_{\max}^2}{(1-\gamma)^2}$ when using REINFORCE gradient estimator (4.4) or $\nu = \frac{G^2\mathcal{R}_{\max}^2}{(1-\gamma)^3}$ when using GPOMDP gradient estimator (4.6).*

**Bounded variance of the gradient estimator.** Interestingly, from (4.20) we immediately obtain

$$\mathbb{V}\mathrm{ar}\left[\widehat{\nabla}_m J(\theta)\right] = \mathbb{E}\left[\left\|\widehat{\nabla}_m J(\theta)\right\|^2\right] - \|\nabla J_H(\theta)\|^2 \overset{(4.20)}{\leq} \frac{\nu - \|\nabla J_H(\theta)\|^2}{m} \leq \frac{\nu}{m}, \tag{4.21}$$

which was used as an assumption by Papini et al. (2018), Xu et al. (2020a), Xu et al. (2020b), Yuan et al. (2020), Huang et al. (2020), and Liu et al. (2020). Yet (4.21) needs not to be an additional assumption since it is a direct consequence of Assumption 4.8.

The (LS) and (E-LS) form the backbone of our hierarchy of assumptions in Figure 4.1. In particular, (LS) implies (E-LS), and thus ABC is the weaker (and most general) assumption of the three.

**Corollary 4.10.** *The (ABC) assumption is the weakest condition compared to (LS) and (E-LS).*

#### 4.4.1.2 Sample complexity analysis for stationary point convergence

Of independent interest to the ABC assumption, Assumption 4.8 also implies the smoothness of $J(\cdot)$ and the truncated gradient assumptions as reported in the following lemmas.

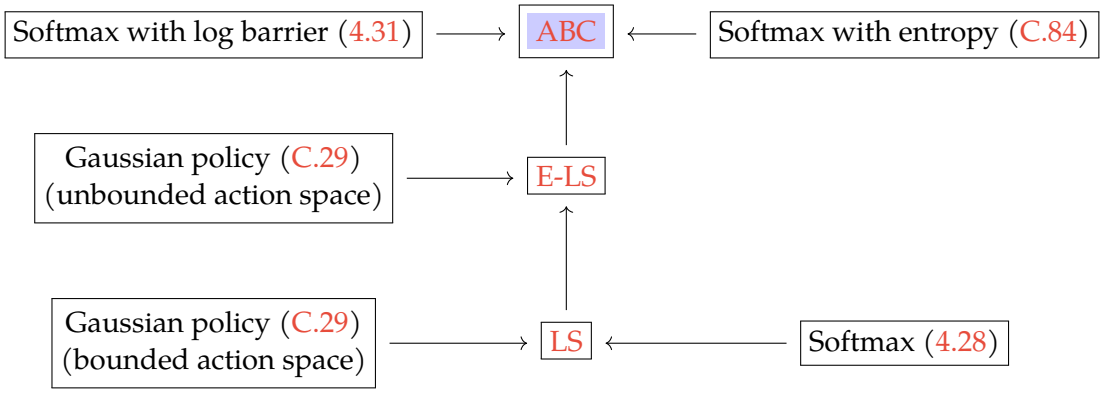


**Figure 4.1** – A hierarchy between the assumptions we present throughout the chapter. An arrow indicates an implication.

**Lemma 4.11.** *Under Assumption 4.8, $J(\cdot)$ is L-smooth, namely $\|\nabla^2 J(\theta)\| \leq L$ for all $\theta$ which is a sufficient condition of Assumption 4.1, with*

$$L = \frac{\mathcal{R}_{\max}}{(1-\gamma)^2}\left(G^2 + F\right). \tag{4.22}$$

The smoothness constant (4.22) is tighter by a factor of $1-\gamma$ as compared to the smoothness constant proposed in Papini et al. (2022). This is the tightest upper bound of $\nabla^2 J(\cdot)$ we are aware of in the existing literature. (see Appendix C.1.3 for more details).

**Lemma 4.12.** *Under Assumption 4.8, Assumption 4.2 holds with*

$$D = \frac{D' G \mathcal{R}_{\max}}{(1-\gamma)^{3/2}}, \tag{4.23}$$

$$D' = \frac{G\mathcal{R}_{\max}}{1-\gamma}\sqrt{\frac{1}{1-\gamma} + H}. \tag{4.24}$$

The coefficient $D'$ in (4.24) got improved and is tighter by a factor of $(1-\gamma)^{1/2}$ as compared to the same term analysed in Lemma B.1 in Liu et al. (2020).

As a by-product, in Lemma C.9 in the appendix, we also show that $J(\cdot)$ is Lipschitz under Assumption 4.8 with a tighter Lipschitzness constant, as compared to Papini et al. (2022), Xu et al. (2020b), and Yuan et al. (2020). See more details in Appendix C.4.5.

Now we can establish the sample complexity of vanilla PG for the expected Lipschitz and smooth policy assumptions as a corollary of Theorem 4.4 and Lemmas 4.9, 4.11, and 4.12.

**Corollary 4.13.** *Suppose that Assumption 4.8 is satisfied. Let* $\delta_0 \overset{def}{=} J^* - J(\theta_0)$. *The PG method applied in* (4.10) *with a mini-batch sampling of size* $m$ *and constant step size*

$$\eta \in \left(0, \frac{2}{L(1-1/m)}\right), \tag{4.25}$$

*satisfies*

$$\begin{aligned}\mathbb{E}\left[\|\nabla J(\theta_U)\|^2\right] \leq & \frac{2\delta_0}{\eta T\left(2-L\eta\left(1-\frac{1}{m}\right)\right)} + \frac{L\nu\eta}{m\left(2-L\eta\left(1-\frac{1}{m}\right)\right)} \\ & + \left(\frac{2D\left(3-L\eta\left(1-\frac{1}{m}\right)\right)}{2-L\eta\left(1-\frac{1}{m}\right)} + D'^2\gamma^H\right)\gamma^H,\end{aligned} \tag{4.26}$$

*where* $\nu, L$ *and* $D, D' > 0$ *are provided in Lemmas 4.9, 4.11 and 4.12, respectively.*

We first note that Corollary 4.13 imposes no restriction on the batch size, allowing us to analyse both exact full PG and its stochastic variants REINFORCE and GPOMDP. For exact PG, i.e., $H = m = \infty$, we recover the $\mathcal{O}(1/T)$ convergence. This translates to an iteration complexity $T = \mathcal{O}\left(\frac{1}{\epsilon^2}\right)$ with a constant step size $\eta = \frac{1}{L}$ to guarantee $\mathbb{E}\left[\|\nabla J(\theta_U)\|^2\right] = \mathcal{O}(\epsilon^2)$. On the other extreme, when $m = 1$, by (4.25) we have that $\eta \in (0, \infty)$, i.e., we place no restriction on the step size. In this case, we have that (4.26) reduces to

$$\mathbb{E}\left[\|\nabla J(\theta_U)\|^2\right] \leq \frac{\delta_0}{\eta T} + \frac{L\nu\eta}{2} + \left(3D + D'^2\gamma^H\right)\gamma^H.$$

Thus the stepsize $\eta$ controls the trade-off between the rate of convergence $\frac{1}{\eta T}$ and leading constant term $\frac{L\nu\eta}{2}$. Using Corollary 4.13, next we develop an explicit sample complexity for PG methods.

**Corollary 4.14.** *Consider the setting of Corollary 4.13. For a given* $\epsilon > 0$, *by choosing the mini-batch size* $m$ *such that* $1 \leq m \leq \frac{2\nu}{\epsilon^2}$, *the step size* $\eta = \frac{\epsilon^2 m}{2L\nu}$, *the number of iterations* $T$ *such that*

$$Tm \geq \frac{8\delta_0 L\nu}{\epsilon^4} = \begin{cases} \mathcal{O}\left(\frac{H}{(1-\gamma)^4\epsilon^4}\right) & \text{for REINFORCE} \\ \mathcal{O}\left(\frac{1}{(1-\gamma)^5\epsilon^4}\right) & \text{for GPOMDP} \end{cases} \tag{4.27}$$

*and the horizon* $H = \mathcal{O}\left((1-\gamma)^{-1}\log(1/\epsilon)\right)$, *then* $\mathbb{E}\left[\|\nabla J(\theta_U)\|^2\right] = \mathcal{O}(\epsilon^2)$.

**Remark.** Given the horizon $H = \mathcal{O}\left((1-\gamma)^{-1}\log(1/\epsilon)\right)$, we have that (4.27) shows that the sample complexity of GPOMDP is a factor of $\log(1/\epsilon)$ smaller than that of REINFORCE.

Corollary 4.14 greatly extends the range of parameters for which PG is guaranteed to converge within the existing literature. It shows that it is *possible* for vanilla policy gradient methods to converge with a mini-batch size per iteration from 1 to $\mathcal{O}(\epsilon^{-2})$ and a constant step size chosen accordingly between $\mathcal{O}(\epsilon^2)$ and $\mathcal{O}(1)$, while still achieving the $Tm \times H = \widetilde{\mathcal{O}}\left(\epsilon^{-4}\right)$ optimal complexity.

In particular, Corollary 4.4 in Zhang et al. (2020b), Proposition 1 in Xiong et al. (2021) and Theorem E.1 in Liu et al. (2020) establish $\widetilde{\mathcal{O}}\left(\epsilon^{-4}\right)$ for FOSP convergence by using the more restrictive assumption (LS). Papini (2020) obtain the same results with the weaker assumption (E-LS), which is also our case. However, we improve upon all of them by recovering the exact full PG analysis, allowing much wider range of choices for the batch size $m$ and the constant step size $\eta$ to achieve the same optimal sample complexity $\widetilde{\mathcal{O}}\left(\epsilon^{-4}\right)$. Indeed, to achieve the optimal sample complexity of FOSP, Papini (2020), Zhang et al. (2020b), Xiong et al. (2021), and Liu et al. (2020) do not allow a single trajectory sampled per iteration. They require the batch size $m$ to be either $\epsilon^{-1}$ or $\epsilon^{-2}$. The existing analysis for vanilla PG that allows $m = 1$ that we are aware of is Zhang et al. (2021b), which we compare with in Section 4.4.2.1 under the specific setting of softmax tabular policy with log barrier regularization for the average regret analysis.

### 4.4.2 Softmax tabular policy

In this section, we instantiate the FOSP convergence results of Corollary 4.13 and 4.14 in the case of the softmax tabular policy. Combined with the specific properties of the softmax, our general theory also recovers the average regret of the global optimum convergence analysis for the softmax with log barrier regularization (Zhang et al., 2021b) and brings new insights of the theory by leveraing the ABC assumption analysis.

Here the state space $\mathcal{S}$ and the action space $\mathcal{A}$ are finite. For all $\theta \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ and any state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$, consider the following softmax tabular policy

$$\pi_\theta(a \mid s) \overset{\text{def}}{=} \frac{\exp(\theta_{s,a})}{\sum_{a' \in \mathcal{A}} \exp(\theta_{s,a'})}. \tag{4.28}$$

We show that the softmax tabular policy satisfies (E-LS) as illustrated in the following lemma.

**Lemma 4.15.** *The softmax tabular policy satisfies Assumption 4.8 with* $G^2 = 1 - \frac{1}{|\mathcal{A}|}$ *and* $F = 1$*, that is, for all* $s \in \mathcal{S}$*, we have*

$$\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\|\nabla_\theta \log \pi_\theta(a \mid s)\|^2\right] \leq 1 - \frac{1}{|\mathcal{A}|}, \tag{4.29}$$

$$\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\left\|\nabla^2_\theta \log \pi_\theta(a \mid s)\right\|\right] \leq 1. \tag{4.30}$$

**Remark.** The softmax tabular policy also satisfies (LS) but with a bigger constant (see Appendix C.5.2).

Lemma 4.15 and the results in Section 4.4.1 immediately imply that all assumptions including the (ABC) assumption of Theorem 4.4 are verified. Thus, as a consequence of Corollary 4.13 and 4.14, we have the following sample complexity for the softmax tabular policy.[6]

**Corollary 4.16** (Informal)**.** *Given* $\epsilon > 0$*, there exists a range of parameter choices for the batch size* $m$ *s.t.* $1 \leq m \leq \mathcal{O}(\epsilon^{-2})$*, the step size* $\eta$ *s.t.* $\mathcal{O}(\epsilon^2) \leq \eta \leq \mathcal{O}(1)$*, the number of iterations* $T$ *and the horizon* $H$ *such that the sample complexity of the vanilla PG (either REINFORCE or GPOMDP) is* $Tm \times H = \widetilde{\mathcal{O}}\left(\frac{1}{(1-\gamma)^6\epsilon^4}\right)$ *to achieve* $\mathbb{E}\left[\|\nabla J(\theta_U)\|^2\right] = \mathcal{O}(\epsilon^2)$*.*

#### 4.4.2.1 Global optimum convergence of softmax with log barrier regularization

Leveraging the work of Agarwal et al. (2021) and our Theorem 4.4, we can establish a global optimum convergence analysis for softmax policies with log barrier regularization.

Log barrier regularization is often used to prevent the policy from becoming deterministic. Indeed, when optimizing the softmax by PG, policies can rapidly become near deterministic and the optimal policy is usually obtained by sending some parameters to infinity. This can result in an extremely slow convergence of PG. Li et al. (2021a) show that PG can even take exponential time to converge. To prevent the parameters from becoming too large and to ensure enough exploration, an entropy-based regularization term is commonly used to keep the probabilities from getting too small (Williams and Peng, 1991; Mnih et al., 2016; Nachum et al., 2017; Haarnoja et al., 2018; Mei et al., 2019) . Here we study stochastic gradient ascent on a relative entropy regularized objective, softmax with log barrier regularization, which is

[6] The exact statement is similar to Corollary 4.14. Thus, we report a more compact statement in Appendix C.5.2 (Corollary C.11).

defined as

$$L_\lambda(\theta) \overset{\text{def}}{=} J(\theta) - \lambda \mathbb{E}_{s\sim \text{Unif}_\mathcal{S}} \left[\text{KL}(\text{Unif}_\mathcal{A}, \pi_\theta(\cdot \mid s))\right]$$
$$= J(\theta) + \frac{\lambda}{|\mathcal{A}||\mathcal{S}|} \sum_{s,a} \log \pi_\theta(a \mid s) + \lambda \log |\mathcal{A}|, \tag{4.31}$$

where the relative entropy for distributions $p$ and $q$ is defined as $\text{KL}(p, q) \overset{\text{def}}{=} \mathbb{E}_{x\sim p}\left[-\frac{\log q(x)}{\log p(x)}\right]$, $\text{Unif}_\chi$ denotes the uniform distribution over a set $\chi$ and $\lambda > 0$ determines the strength of the penalty.

Let $\widehat{\nabla}_m L_\lambda(\theta)$ be the stochastic gradient estimator of $L_\lambda(\theta)$ using REINFORCE or GPOMDP with batch size $m$ (see the closed form of $\widehat{\nabla}_m L_\lambda(\theta)$ in (C.58)). Thus $\widehat{\nabla}_m L_\lambda(\theta)$ is an unbiased estimate of the gradient of the truncated function

$$L_{\lambda,H}(\theta) \overset{\text{def}}{=} J_H(\theta) + \frac{\lambda}{|\mathcal{A}||\mathcal{S}|} \sum_{s,a} \log \pi_\theta(a \mid s) + \lambda \log |\mathcal{A}|. \tag{4.32}$$

We show in the following that $\widehat{\nabla}_m L_\lambda(\theta)$ satisfies the (ABC).

**Lemma 4.17.** *Consider $\widehat{\nabla}_m L_\lambda(\theta)$ by using either RIENFORCE (4.4) or GPOMDP (4.6), Assumption 4.3 holds with $A = 0, B = 1 - \frac{1}{m}$ and $C = \frac{\nu}{m}$, that is,*

$$\mathbb{E}\left[\left\|\widehat{\nabla}_m L_\lambda(\theta)\right\|^2\right] \leq \left(1 - \frac{1}{m}\right) \|\nabla L_{\lambda,H}(\theta)\|^2 + \frac{\nu}{m}, \tag{4.33}$$

*where $\nu = 2\left(1 - \frac{1}{|\mathcal{A}|}\right)\left(\frac{H\mathcal{R}_{\max}^2}{(1-\gamma)^2} + \frac{\lambda^2}{|\mathcal{S}|}\right)$ when using REINFORCE or $\nu = 2\left(1 - \frac{1}{|\mathcal{A}|}\right)\left(\frac{\mathcal{R}_{\max}^2}{(1-\gamma)^3} + \frac{\lambda^2}{|\mathcal{S}|}\right)$ when using GPOMDP.*

Similar to the softmax case, we show in Appendix C.5.3 that $L_\lambda(\theta)$ is also smooth and verifies Assumption 4.2. Thus from Theorem 4.4, we have $\{\theta_t\}_{t\geq 0}$ converges to a FOSP of $L_\lambda(\cdot)$. See the formal statement of this result in Appendix C.5.3 (Corollary C.13).

Besides, thanks to Theorem 5.2 in Agarwal et al. (2021), the FOSP of $L_\lambda(\cdot)$ is approximately the global optimal solution of $J(\cdot)$ when the regularization parameter $\lambda$ is sufficiently small. As a by-product, we can also establish a high probability global optimum convergence analysis (Appendix C.5.4).

In the following corollary, we show that we can leverage the versatility of Theorem 4.4 to derive yet another type of result: a guarantee on the average regret w.r.t. the global optimum.

**Corollary 4.18.** *Given $\epsilon > 0$, consider the batch size $m$ such that $1 \leq m \leq \frac{1}{(1-\gamma)^6\epsilon^3}$, the step size $\mathcal{O}(\epsilon^3) \leq \eta = \frac{(1-\gamma)^3\epsilon^3 m}{2L\nu} \leq \mathcal{O}(1)$ with $L, \nu$ in the setting of Corollary C.13 . If the horizon $H = \mathcal{O}\left(\frac{\log(1/\epsilon)}{1-\gamma}\right)$ and the number of iterations $T$ is such that*

$$Tm \times H \;\geq\; \widetilde{\mathcal{O}}\left(\frac{1}{(1-\gamma)^{12}\epsilon^6}\right),$$

*we have $J^* - \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[J(\theta_t)\right] \;=\; \mathcal{O}(\epsilon)$.*

This result recovers the sample complexity $\widetilde{\mathcal{O}}(\epsilon^{-6})$ of Zhang et al. (2021b). However, Zhang et al. (2021b) do not study the vanilla policy gradient. Instead, they add an extra phased learning step to enforce the exploration of the MDP and use a decreasing step size. Our result shows that such extra phased learning step is unnecessary and the step size can be constant. We also provide a wider range of parameter choices for the batch size and the step size with the same sample complexity.

As Agarwal et al. (2021) mentioned, the regularizer (4.31) is more "aggressive" in penalizing small probabilities than the more commonly utilized entropy regularizer. We also show that entropy regularized softmax satisfies the (ABC) and provide its FOSP analysis in Appendix C.7, again thanks to the versatility of Theorem 4.4. Notice that for the FOSP convergence, only an asymptotic result was established in Lemma 4.4 in Ding et al. (2021). Thus all proofs and implications in Figure 4.1 are provided.

### 4.4.3 Fisher-non-degenerate parameterization

In this section, we study a general policy class that satisfies the following assumption.

**Assumption 4.19** (Fisher-non-degenerate, Assumption 2.1 in Ding et al. (2022))**.** *For all $\theta \in \mathbb{R}^d$, there exists $\mu_F > 0$ s.t. the Fisher information matrix $F_\rho(\theta)$ induced by policy $\pi_\theta$ and initial state distribution $\rho$ satisfies*

$$F_\rho(\theta) \overset{def}{=} \mathbb{E}_{(s,a)\sim v_\rho^{\pi_\theta}}\left[\nabla_\theta \log \pi_\theta(a \mid s) \nabla_\theta \log \pi_\theta(a \mid s)^\top\right] \geq \mu_F \mathbf{I}_d, \tag{FI}$$

*where $v_\rho^{\pi_\theta}$ is the state-action visitation measure defined as*

$$v_\rho^{\pi_\theta}(s,a) \overset{def}{=} (1-\gamma)\mathbb{E}_{s_0\sim\rho}\sum_{t=0}^{\infty}\gamma^t \mathbb{P}(s_t = s, a_t = a | s_0, \pi_\theta).$$

This assumption is commonly used in the literatures (Liu et al., 2020; Ding et al., 2022). Similar conditions of the Fisher-non-degeneracy is also required in other global optimum convergence framework (Assumption 6.5 in Agarwal et al. (2021) on the relative condition number). This assumption is satisfied by a wide families of policies, including the Gaussian policy (C.29) and certain neural policy. We refer to Section B.2 in Liu et al. (2020) and Section 8 in Ding et al. (2022) for more discussions on the Fisher-non-degenerate setting.

We also need the following *compatible function approximation error* assumption[7].

**Assumption 4.20** (Compatible, Assumption 4.6 in Ding et al. (2022))**.** *For all $\theta \in \mathbb{R}^d$, there exists $\epsilon_{bias} > 0$ s.t. the* transferred compatible function approximation error *with $(s,a) \sim v_\rho^{\pi_{\theta^*}}$ satisfies*

$$\mathbb{E}\left[\left(A^{\pi_\theta}(s,a) - (1-\gamma)u^{*\top}\nabla_\theta \pi_\theta(a \mid s)\right)^2\right] \le \epsilon_{bias}, \qquad \text{(compatible)}$$

*where $v_\rho^{\pi_{\theta^*}}$ is the state-action distribution induced by an optimal policy $\pi_{\theta^*}$, $u^* = (F_\rho(\theta))^\dagger \nabla J(\theta)$.*

This is also a common assumption (Wang et al., 2020; Agarwal et al., 2021; Liu et al., 2020; Ding et al., 2022). In particular, when $\pi_\theta$ is a softmax tabular policy (C.45), $\epsilon_{bias}$ is 0 (Ding et al., 2022); when $\pi_\theta$ is a rich neural policy, $\epsilon_{bias}$ is small (Wang et al., 2020).

Combining Assumption (FI), (compatible) with Assumption E-LS, by Lemma 4.7 in Ding et al. (2022), we know that $J(\cdot)$ satisfies the relaxed weak gradient domination property (4.17) with $\epsilon' = \frac{\mu_F\sqrt{\epsilon_{bias}}}{(1-\gamma)G}$ and $\mu = \frac{\mu_F^2}{4G^2}$. Consequently, we recover the average regret convergence result $\mathcal{O}(\epsilon^{-4})$ of Liu et al. (2020) in Corollary C.17 in Appendix C.6.1 with weaker assumption and allowing wider range of parameter choices. We also have the following new global optimum convergence result for the Fisher-non-degenerate parametrized policy.

**Corollary 4.21.** *If the policy $\pi_\theta$ satisfies Assumption 4.8, 4.19 and 4.20, consider the setting of Corollary 4.7 with $\epsilon' = \frac{\mu_F\sqrt{\epsilon_{bias}}}{(1-\gamma)G}$ and $\mu = \frac{\mu_F^2}{4G^2}$. Then $\min_{t\in\{0,1,\cdots,T\}} J^* - \mathbb{E}\left[J(\theta_t)\right] \le \mathcal{O}(\epsilon) + \mathcal{O}(\sqrt{\epsilon_{bias}})$ and the sample complexity $T \times H = \widetilde{\mathcal{O}}(\epsilon^{-3})$ when $\epsilon_{bias} = 0$ or $T \times H = \widetilde{\mathcal{O}}((\epsilon_{bias}\cdot\epsilon)^{-1})$ when $\epsilon_{bias} > 0$.*

## 4.5 Discussion and bibliographical remarks

In this chapter, thanks to the recent tools developed for the SGD analysis in the nonconvex optimization, we improved the convergence and sample complexity analysis for the vanilla PG

[7] We defer the definition of the advantage function $A^{\pi_\theta}$ in Appendix C.6.

under the general ABC assumption. This generality allowed us to unify much of the fragmented results in the RL literature and brought new insight for the hyperparameter choices of the PG algorithm. When an additional (weak) gradient domination condition is available, we established the global optimum convergence results of vanilla PG and improved the current best known sample complexity results for the stochastic vanilla PG. The results we have obtained open up several experimental questions related to parameter settings for PG. We leave such questions as an important future work to further support our theoretical findings.

One natural open question is whether the ABC assumption and the associated analysis can be extended to the projected PG. If the answer is positive, this might improve the sample complexity analysis of the direct policy parameterization setting in the stochastic case. Indeed, the direct policy parameterization satisfies a variant of weak gradient domination condition (4.17) (Agarwal et al., 2021; Xiao, 2022) under the proximal framework. If the ABC assumption and the associated analysis of Theorem C.8, which also uses the (4.17) condition, can be extended to the proximal framework, it might be possible to establish the $\widetilde{\mathcal{O}}(\epsilon^{-3})$ sample complexity as our Theorem C.8 for the global optimum convergence for the direct policy parameterization and allow for a wider range of hyperparameter choices.

Similarly, we wonder if the ABC assumption and the associated analysis can be extended to the LQR setting. The challenge here will be the smoothness assumption and whether the ABC assumption is satisfied by the LQR when doing the stochastic PG updates. Indeed, the LQR only has an "almost" smoothness property (Fazel et al., 2018). One needs to investigate how this will affect the current ABC analysis by extending the smoothness property to the "almost" smoothness property.

It is worth mentioning that although the main focus of this chapter is the theoretical analysis of vanilla variants of the PG method, Theorem C.8 under the relaxed weak gradient domination assumption (4.17) is new for nonconvex SGD analysis and of independent interest. Recently, Fatkhullin et al. (2022) extended the SGD analysis of Khaled and Richtárik (2023) and our Theorem C.8 under more general assumptions. In particular, they relax the ABC assumption by so called the expected smoothness of order $k$ assumption (Assumption 4 in their paper) and relax the weak gradient domination assumption (4.17) by the global Kurdyka-Łojasiewicz assumption (Assumption 2 in their paper). We refer to Fatkhullin et al. (2022, Section 2) for more details on their assumptions. Despite the generality of the assumptions they consider, they are able to recover our SGD convergence analysis as special cases. As a result, this might improve our vanilla PG analysis to a even more general setting.

# Chapter 5

# Linear Convergence of Natural Policy Gradient Methods with Log-Linear Policies

In this chapter, we consider infinite-horizon discounted Markov decision processes and study the convergence rates of the natural policy gradient (NPG) and the Q-NPG methods with the log-linear policy class. Using the compatible function approximation framework, both methods with log-linear policies can be written as inexact versions of the policy mirror descent (PMD) method. We show that both methods attain linear convergence rates and $\tilde{\mathcal{O}}(1/\epsilon^2)$ sample complexities using a simple, non-adaptive geometrically increasing step size, without resorting to entropy or other strongly convex regularization. Lastly, as a byproduct, we obtain sublinear convergence rates for both methods with arbitrary constant step size. [1]

## Contents



[1] This chapter is based on an article published in the proceedings of the $11^{th}$ International Conference on Learning Representations (ICLR 2023) (Yuan et al., 2023).

## 5.1 Introduction

Policy gradient (PG) methods have emerged as a popular class of algorithms for reinforcement learning. Unlike classical methods based on (approximate) dynamic programming (e.g., Puterman, 1994; De Farias and Van Roy, 2003; Bertsekas, 2012; Sutton and Barto, 2018), PG methods update directly the policy and its parametrization along the gradient direction of the value function (e.g., Williams, 1992; Sutton et al., 2000; Konda and Tsitsiklis, 2000; Baxter and Bartlett, 2001). An important variant of PG is the natural policy gradient (NPG) method (Kakade, 2001), which is a direct application of natural gradient method (Amari, 1998) for RL. NPG uses the Fisher information matrix of the policy distribution as a preconditioner to improve the policy gradient direction, similar to quasi-Newton methods in classical optimization (Martens, 2020). Variants of NPG with policy parametrization through deep neural networks were shown to have impressive empirical successes (Schulman et al., 2015; Lillicrap et al., 2016; Mnih et al., 2016; Schulman et al., 2017; Haarnoja et al., 2018; Tomar et al., 2022).

Motivated by the success of NPG in practice, there is now a concerted effort to develop convergence theories for the NPG method. Neu et al. (2017) provide the first interpretation of NPG as a mirror descent (MD) method (Nemirovski and Yudin, 1983; Beck and Teboulle, 2003). By leveraging different techniques for analyzing MD, it has been established that NPG converges to the global optimum in the tabular case (Agarwal et al., 2021; Khodadadian et al., 2021b; Xiao, 2022) and some more general settings (Shani et al., 2020; Vaswani et al., 2022; Grudzien et al., 2022; Chen and Theja Maguluri, 2022). In order to get a fast linear convergence rate for NPG, several recent works consider the regularized NPG methods, such as the entropy-regularized NPG (Cen et al., 2021a) and other convex regularized NPG methods (Lan, 2022; Zhan et al., 2021). By designing appropriate step sizes, Khodadadian et al. (2021b) and Xiao (2022) obtain linear convergence of NPG without regularization (See Appendix D.1 for a thorough review. In particular, Table D.1 provides a complete overview of our results.). However, all these linear convergence results are limited in the tabular setting (direct parametrization). It remains unclear whether this same linear convergence rate can be established in the function approximation regime.

In this chapter we provide an affirmative answer to this question for the log-linear policy class. Our approach is based on the framework of *compatible function approximation* (Sutton et al., 2000; Kakade, 2001), which was extensively developed by Agarwal et al. (2021). Using this framework, variants of NPG with log-linear policies can be written as policy mirror descent (PMD) methods with inexact evaluations of the advantage function or Q-function (giving rise to NPG or Q-NPG respectively). Then by extending a recent analysis of PMD (Xiao, 2022), we obtain a non-asymptotic linear convergence of both NPG and Q-NPG with log-linear policies. A distinctive feature of this approach is the use of a simple, non-adaptive geometrically increasing step size, without resorting to entropy or other (strongly) convex regularization.

### 5.1.1 Outline and contributions

In Section 5.2 we review the fundamentals of Markov decision processes (MDP), and describe the log-linear policy class and the general NPG method. In Section 5.3 we explain the compatible function approximation framework and show that both NPG and Q-NPG can be expressed as inexact versions of the PMD method.

Our main contributions start from Section 5.4, which contains our results on Q-NPG. We present convergence results of Q-NPG in two different settings: one assuming bounded *transfer error* and a relative condition number (Section 5.4.1) and the other assuming bounded approximation error (Section 5.4.2). In both cases, we obtain linear convergence up to an error floor towards the global optima. The extensions of the analysis of PMD (Xiao, 2022) are highly nontrivial and require quite different techniques (see Appendix D.1.1 for more details). Compared with the sublinear convergence results of Agarwal et al. (2021), we do not need a projection step nor the assumption of bounded feature maps. However, our results depends on some distribution mismatch coefficients and has larger error floors. In Section 5.4.3, by further assuming that the feature maps are bounded and have a non-singular covariance matrix, we obtain an $\tilde{\mathcal{O}}(1/\epsilon^2)$ sample complexity for Q-NPG with log-linear policies. In particular, our sample complexity analysis also fixes errors of previous work.

In Section 5.5, we analyze the NPG method under the assumption of bounded approximation error, and show that it also enjoys linear convergence up to an error floor as well as an $\tilde{\mathcal{O}}(1/\epsilon^2)$ sample complexity. As a by product of our analysis, we also obtain sublinear an $\mathcal{O}(1/k)$ convergence rate for both NPG and Q-NPG with unconstrained constant step sizes and no projection step.

## 5.2 Preliminaries on Markov decision processes

We consider an MDP denoted as $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{P}, c, \gamma\}$, where $\mathcal{S}$ is a finite state space, $\mathcal{A}$ is a finite action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is a Markovian transition model with $\mathcal{P}(s' \mid s, a)$ being the transition probability from state $s$ to $s'$ under action $a$, $c$ is a cost function with $c(s, a) \in [0, 1]$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$, and $\gamma \in [0, 1)$ is a discounted factor. Here we use cost instead of reward to better align with the minimization convention in the optimization literature.

Let $\Delta(\mathcal{X})$ denote the probability simplex for an arbitrary set $\mathcal{X}$. The agent's behavior is modeled as a stochastic policy $\pi \in \Delta(\mathcal{A})^{|\mathcal{S}|}$, where $\pi_s \in \Delta(\mathcal{A})$ is the probability distribution over actions $\mathcal{A}$ in state $s \in \mathcal{S}$. At each time $t$, the agent takes an action $a_t \in \mathcal{A}$ given the current state $s_t \in \mathcal{S}$, following the policy $\pi$, i.e., $a_t \sim \pi_{s_t}$. Then the MDP transitions into the next state $s_{t+1}$ with probability $\mathcal{P}(s_{t+1} \mid s_t, a_t)$ and the agent encounters the cost $c_t = c(s_t, a_t)$. Thus, a policy induces a distribution over trajectories $\{s_t, a_t, c_t\}_{t \geq 0}$. In the infinite-horizon discounted

setting, the cost function of $\pi$ with an initial state $s$ is defined as

$$V_s(\pi) \overset{\text{def}}{=} \mathop{\mathbb{E}}_{\substack{a_t \sim \pi_{s_t} \\ s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)}} \left[ \sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \mid s_0 = s \right]. \tag{5.1}$$

Given an initial state distribution $\rho \in \Delta(\mathcal{S})$, the goal of the agent is to find a policy $\pi$ that minimizes the expected cost function

$$V_\rho(\pi) \overset{\text{def}}{=} \mathbb{E}_{s \sim \rho} [V_s(\pi)] = \sum_{s \in \mathcal{S}} \rho_s V_s(\pi) = \langle V(\pi), \rho \rangle .$$

A more granular characterization of the performance of a policy is the state-action cost function (Q-function). For any pair $(s, a) \in \mathcal{S} \times \mathcal{A}$, it is defined as

$$Q_{s,a}(\pi) \overset{\text{def}}{=} \mathop{\mathbb{E}}_{\substack{a_t \sim \pi_{s_t} \\ s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)}} \left[ \sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \mid s_0 = s, a_0 = a \right]. \tag{5.2}$$

Let $Q_s \in \mathbb{R}^{|\mathcal{A}|}$ denote the vector $[Q_{s,a}]_{a \in \mathcal{A}}$. Then we have $V_s(\pi) = \mathbb{E}_{a \sim \pi_s} [Q_{s,a}(\pi)] = \langle \pi_s, Q_s(\pi) \rangle$. The advantage function[2] is a centered version of the Q-function:

$$A_{s,a}(\pi) \overset{\text{def}}{=} Q_{s,a}(\pi) - V_s(\pi), \tag{5.3}$$

which satisfies $\mathbb{E}_{a \sim \pi_s} [A_{s,a}(\pi)] = 0$ for all $s \in \mathcal{S}$.

**Visitation probabilities.** Given a starting state distribution $\rho \in \Delta(\mathcal{S})$, we define the *state visitation distribution* $d^\pi(\rho) \in \Delta(\mathcal{S})$, induced by a policy $\pi$, as

$$d_s^\pi(\rho) \overset{\text{def}}{=} (1 - \gamma) \, \mathbb{E}_{s_0 \sim \rho} \left[ \sum_{t=0}^{\infty} \gamma^t \Pr^\pi(s_t = s \mid s_0) \right],$$

where $\Pr^\pi(s_t = s \mid s_0)$ is the probability that the $t$-th state is equal to $s$ by following the trajectory generated by $\pi$ starting from $s_0$. Intuitively, the state visitation distribution measures the probability of being at state $s$ across the entire trajectory. We define the *state-action visitation distribution* $\bar{d}^\pi(\rho) \in \Delta(\mathcal{S} \times \mathcal{A})$ as

$$\bar{d}_{s,a}^\pi(\rho) \overset{\text{def}}{=} d_s^\pi(\rho) \pi_{s,a} = (1 - \gamma) \, \mathbb{E}_{s_0 \sim \rho} \left[ \sum_{t=0}^{\infty} \gamma^t \Pr^\pi(s_t = s, a_t = a \mid s_0) \right]. \tag{5.4}$$

[2] An advantage function should measure how much better is $a$ compared to $\pi$, while here $A$ is positive when $a$ is worse than $\pi$. We keep calling $A$ advantage function to better align with the convention in the RL literature.

In addition, we extend the definition of $\bar{d}^{\pi}(\rho)$ by specifying the initial state-action distribution $\nu \in \Delta(\mathcal{S} \times \mathcal{A})$, i.e.,

$$\tilde{d}^{\pi}_{s,a}(\nu) \stackrel{\text{def}}{=} (1-\gamma)\,\mathbb{E}_{(s_0,a_0)\sim\nu}\left[\sum_{t=0}^{\infty}\gamma^t \Pr^{\pi}(s_t = s, a_t = a \mid s_0, a_0)\right]. \tag{5.5}$$

The difference in the last two definitions is that for the former, the initial action $a_0$ is sampled directly from $\pi$, whereas for the latter, it is prescribed by the initial state-action distribution $\nu$. We use $\tilde{d}$ compared to $\bar{d}$ to better distinguish the cases with $\nu$ and $\rho$. Without specification, we even omit the argument $\nu$ or $\rho$ throughout the chapter to simplify the presentation as they are self-evident. From these definitions, we have for all $(s,a) \in \mathcal{S} \times \mathcal{A}$,

$$d^{\pi}_s \geq (1-\gamma)\rho_s, \qquad \bar{d}^{\pi}_{s,a} \geq (1-\gamma)\rho_s \pi_{s,a}, \qquad \tilde{d}^{\pi}_{s,a} \geq (1-\gamma)\nu_{s,a}. \tag{5.6}$$

**Policy parametrization.** In practice, both the state and action spaces $\mathcal{S}$ and $\mathcal{A}$ can be very large and some form of function approximation is needed to reduce the dimensions and make the computation feasible. In particular, the policy $\pi$ is often parametrized as $\pi(\theta)$ with $\theta \in \mathbb{R}^m$, where $m$ is much smaller than $|\mathcal{S}|$ and $|\mathcal{A}|$. In this chapter, we focus on the log-linear policy class. Specifically, we assume that for each state-action pair $(s,a)$, there is a feature mapping $\phi_{s,a} \in \mathbb{R}^m$ and the policy takes the form

$$\pi_{s,a}(\theta) \;=\; \frac{\exp(\phi_{s,a}^{\top}\theta)}{\sum_{a' \in \mathcal{A}} \exp(\phi_{s,a'}^{\top}\theta)}. \tag{5.7}$$

This setting is important since it is the simplest instantiation of the widely-used neural policy parametrization. To simplify notation in the rest of the chapter, we use the shorthand $V_{\rho}(\theta)$ for $V_{\rho}(\pi(\theta))$ and similarly $Q_{s,a}(\theta)$ for $Q_{s,a}(\pi(\theta))$, $A_{s,a}(\theta)$ for $A_{s,a}(\pi(\theta))$, $d^{\theta}_s$ for $d^{\pi(\theta)}_s$, $\bar{d}^{\theta}_{s,a}$ for $\bar{d}^{\pi(\theta)}_{s,a}$, and $\tilde{d}^{\theta}_{s,a}$ for $\tilde{d}^{\pi(\theta)}_{s,a}$.

**Natural Policy Gradient (NPG) Method.** Using the notations defined above, the parametrized policy optimization problem is to minimize the function $V_{\rho}(\theta)$ over $\theta \in \mathbb{R}^m$. The policy gradient is given by the policy gradient theorem (Sutton et al., 2000)

$$\nabla_{\theta} V_{\rho}(\theta) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\theta},\, a\sim\pi_s(\theta)}\left[Q_{s,a}(\theta)\,\nabla_{\theta}\log\pi_{s,a}(\theta)\right]. \tag{5.8}$$

For parametrizations that are differentiable and satisfy $\sum_{a\in\mathcal{A}}\pi_{s,a}(\theta) = 1$, including the log-linear class defined in (5.7), we can replace $Q_{s,a}(\theta)$ by $A_{s,a}(\theta)$ in the above expression (Agarwal et al., 2021). A derivation of (5.8) is provided in Appendix D.2 (Lemma D.1) for completeness.

The NPG method (Kakade, 2001) takes the form

$$\theta^{(k+1)} \;=\; \theta^{(k)} - \eta_k F_\rho(\theta^{(k)})^\dagger \nabla_\theta V_\rho(\theta^{(k)}), \tag{5.9}$$

where $\eta_k > 0$ is a scalar step size, $F_\rho(\theta)$ is the Fisher information matrix

$$F_\rho(\theta) \;\overset{\text{def}}{=}\; \mathbb{E}_{s\sim d^\theta,\, a\sim\pi_s(\theta)} \left[\nabla_\theta \log \pi_{s,a}(\theta) (\nabla_\theta \log \pi_{s,a}(\theta))^\top\right], \tag{5.10}$$

and $F_\rho(\theta)^\dagger$ denotes the Moore-Penrose pseudoinverse of $F_\rho(\theta)$.

## 5.3 NPG with compatible function approximation

The parametrized value function $V_\rho(\theta)$ is non-convex in general (see, e.g., Agarwal et al., 2021). Despite being a non-convex optimization problem, there is still additional structure we can leverage to ensure convergence. Following Agarwal et al. (2021), we adopt the framework of *compatible function approximation* (Sutton et al., 2000; Kakade, 2001), which exploits the MDP structure and leads to tight convergence rate analysis.

For any $w \in \mathbb{R}^m$, $\theta \in \mathbb{R}^m$ and state-action distribution $\zeta \in \Delta(\mathcal{S}\times\mathcal{A})$, we define the *compatible function approximation error* as

$$L_A(w, \theta, \zeta) \;\overset{\text{def}}{=}\; \mathbb{E}_{(s,a)\sim\zeta} \left[\left(w^\top \nabla_\theta \log \pi_{s,a}(\theta) - A_{s,a}(\theta)\right)^2\right]. \tag{5.11}$$

Kakade (2001) showed that the NPG update (5.9) is equivalent to (up to a constant scaling of $\eta_k$)

$$\theta^{(k+1)} \;=\; \theta^{(k)} - \eta_k w_\star^{(k)}, \qquad w_\star^{(k)} \in \operatorname{argmin}_{w\in\mathbb{R}^m} L_A(w, \theta^{(k)}, \bar{d}^{(k)}), \tag{5.12}$$

where $\bar{d}^{(k)}$ is a shorthand for the state-action visitation distribution $\bar{d}^{\pi(\theta^{(k)})}(\rho)$ defined in (5.4). A derivation of (5.12) is provided in Appendix D.2 (Lemma D.2) for completeness. In other words, $w_\star^{(k)}$ is the solution to a regression problem that tries to approximate $A_{s,a}(\theta^{(k)})$ using $\nabla_\theta \log \pi_{s,a}(\theta^{(k)})$ as features. This is where the term "compatible function approximation error" comes from. For the log-linear policy class defined in (5.7), we have

$$\nabla_\theta \log \pi_{s,a}(\theta) \;=\; \bar{\phi}_{s,a}(\theta) \;\overset{\text{def}}{=}\; \phi_{s,a} - \textstyle\sum_{a'\in\mathcal{A}} \pi_{s,a'}(\theta)\phi_{s,a'} \;=\; \phi_{s,a} - \mathbb{E}_{a'\sim\pi_s(\theta)}\left[\phi_{s,a'}\right], \tag{5.13}$$

where $\bar{\phi}_{s,a}(\theta)$ are called *centered features vectors*.

In practice, we cannot minimize $L_A$ exactly; instead, a sample-based regression problem is solved to obtain an inexact solution $w^{(k)}$. This leads to the following inexact NPG update rule:

$$\theta^{(k+1)} = \theta^{(k)} - \eta_k w^{(k)}, \qquad w^{(k)} \approx \operatorname{argmin}_w L_A(w, \theta^{(k)}, \bar{d}^{(k)}). \tag{5.14}$$

The inexact NPG updates require samples of unbiased estimates of $A_{s,a}(\theta)$, the corresponding sampling procedure is given in Algorithm 14, and a sample-based regression solver to minimize $L_A$ is given in Algorithm 15 in the Appendix.

Alternatively, as proposed by Agarwal et al. (2021), we can define the compatible function approximation error as

$$L_Q(w, \theta, \zeta) \overset{\text{def}}{=} \mathbb{E}_{(s,a)\sim\zeta}\left[\left(w^\top \phi_{s,a} - Q_{s,a}(\theta)\right)^2\right] \tag{5.15}$$

and use it to derive a variant of the inexact NPG update called *Q-NPG*:

$$\theta^{(k+1)} = \theta^{(k)} - \eta_k w^{(k)}, \qquad w^{(k)} \approx \operatorname{argmin}_w L_Q(w, \theta^{(k)}, \bar{d}^{(k)}). \tag{5.16}$$

For Q-NPG, the sampling procedure for estimating $Q_{s,a}(\theta)$ is given in Algorithm 13 and a sample-based regression solver for $w^{(k)}$ is proposed in Algorithm 16 in the Appendix.

The sampling procedure and the regression solver of NPG are less efficient than those of Q-NPG. Indeed, the sampling procedure for $A_{s,a}(\theta)$ in Algorithm 14 not only estimates $Q_{s,a}(\theta)$, but also requires an additional estimation of $V_s(\theta)$, and thus doubles the amount of samples as compared to Algorithm 13. Furthermore, the stochastic gradient estimator of $L_Q$ in Algorithm 16 only computes on a single action of the feature map $\phi_{s,a}$. Whereas the one of $L_A$ in Algorithm 15 computes on the centered feature map $\bar{\phi}_{s,a}(\theta)$ defined in (5.13), which needs to go through the entire action space, thus is $|\mathcal{A}|$ times more expensive to run. See Appendix D.3 for more details.

Following Agarwal et al. (2021), we consider slightly different variants of NPG and Q-NPG, where $\bar{d}^{(k)}$ in (5.14) and (5.16) is replaced by a more general state-action visitation distribution $\tilde{d}^{(k)} = \tilde{d}^{\pi(\theta^{(k)})}(\nu)$ defined in (5.5) with $\nu \in \Delta(\mathcal{S} \times \mathcal{A})$. The advantage of using $\tilde{d}^{(k)}$ is that it allows better exploration than $\bar{d}^{(k)}$ as $\nu$ can be chosen to be independent to the policy $\pi(\theta^{(k)})$. For example, it can be seen from (5.6) that the lower bound of $\tilde{d}^{\pi}$ is independent to $\pi$, which is not the case for $\bar{d}^{\pi}$. This property is crucial in the forthcoming convergence analysis.

### 5.3.1 Formulation as inexact policy mirror descent

Given an inexact solution $w^{(k)}$ for minimizing $L_Q(w, \theta^{(k)}, \tilde{d}^{(k)})$, the Q-NPG update rule $\theta^{(k+1)} = \theta^{(k)} - \eta_k w^{(k)}$, when plugged in the log-linear parametrization (5.7), results in a new policy

$$\pi_{s,a}^{(k+1)} = \frac{1}{Z_s^{(k)}} \pi_{s,a}^{(k)} \exp\left(-\eta_k \, \phi_{s,a}^T w^{(k)}\right), \qquad \forall\, (s,a) \in \mathcal{S} \times \mathcal{A},$$

where $\pi^{(k)}$ is a shorthand for $\pi_{s,a}(\theta^{(k)})$ and $Z_s^{(k)}$ is a normalization factor to ensure $\sum_{a \in \mathcal{A}} \pi_{s,a}^{(k+1)} = 1$, for each $s \in \mathcal{S}$. We note that the above $\pi^{(k+1)}$ can also be obtained by a mirror descent update:

$$\pi_s^{(k+1)} = \arg\min_{p \in \Delta(\mathcal{A})} \left\{ \eta_k \left\langle \Phi_s w^{(k)}, p \right\rangle + D(p, \pi_s^{(k)}) \right\}, \quad \forall s \in \mathcal{S}, \tag{5.17}$$

where $\Phi_s \in \mathbb{R}^{|\mathcal{A}| \times m}$ is a matrix with rows $(\phi_{s,a})^\top \in \mathbb{R}^m$ for $a \in \mathcal{A}$, and $D(p,q)$ denotes the Kullback-Leibler (KL) divergence between two distributions $p, q \in \Delta(\mathcal{A})$, i.e.,

$$D(p,q) \stackrel{\text{def}}{=} \sum_{a \in \mathcal{A}} p_a \log\left(\frac{p_a}{q_a}\right).$$

A derivation of (5.17) is provided in Appendix D.2 (Lemma D.3) for completeness.

If we replace $\Phi_s w^{(k)}$ in (5.17) by the vector $\left[Q_{s,a}(\pi^{(k)})\right]_{a \in \mathcal{A}} \in \mathbb{R}^{|\mathcal{A}|}$, then it becomes the *policy mirror descent* (PMD) method in the tabular setting studied by, for example, Shani et al. (2020), Lan (2022) and Xiao (2022). In fact, the update rule (5.17) can be viewed as an inexact PMD method where $Q_s(\pi^{(k)})$ is linearly approximated by $\Phi_s w^{(k)}$ through compatible function approximation (5.15). Besides, with the replacement of $\Phi_s w^{(k)}$ by $\left[Q_{s,a}(\pi^{(k)})\right]_{a \in \mathcal{A}}$, (5.17) can also be viewed as a special case of the mirror descent value iteration for the regularized MDP studied by Geist et al. (2019), Vieillard et al. (2020), and Kozuno et al. (2022). Similarly, we can write the inexact NPG update rule as

$$\pi_s^{(k+1)} = \arg\min_{p \in \Delta(\mathcal{A})} \left\{ \eta_k \left\langle \bar{\Phi}_s^{(k)} w^{(k)}, p \right\rangle + D(p, \pi_s^{(k)}) \right\}, \quad \forall s \in \mathcal{S}, \tag{5.18}$$

where $w^{(k)}$ is an inexact solution for minimizing $L_A(w, \theta^{(k)}, \tilde{d}^{(k)})$ defined in (5.11), and $\bar{\Phi}_s^{(k)} \in \mathbb{R}^{|\mathcal{A}| \times m}$ is a matrix whose rows consist of the centered feature maps $\left(\bar{\phi}_{s,a}(\theta^{(k)})\right)^\top$, as defined in (5.13).

Reformulating Q-NPG and NPG into the mirror descent forms (5.17) and (5.18), respectively, allows us to adapt the analysis of PMD method developed in Xiao (2022) to obtain sharp convergence rates. In particular, we show that with an increasing step size $\eta_k \propto \gamma^k$, both NPG and Q-NPG with log-linear policy parametrization converge linearly up to an error floor determined by the quality of the compatible function approximation.

## 5.4 Analysis of Q-NPG with log-linear policies

In this section, we provide the convergence analysis of the following inexact Q-NPG method

$$\theta^{(k+1)} = \theta^{(k)} - \eta_k w^{(k)}, \qquad w^{(k)} \approx \operatorname{argmin}_w L_Q(w, \theta^{(k)}, \tilde{d}^{(k)}), \tag{5.19}$$

where $\tilde{d}^{(k)}$ is shorthand for $\tilde{d}^{\pi(\theta^{(k)})}(\nu)$ and $\nu \in \Delta(\mathcal{S} \times \mathcal{A})$ is an arbitrary state-action distribution that does not depend on $\rho$. The exact minimizer is denoted as $w_\star^{(k)} \in \operatorname{argmin}_w L_Q(w, \theta^{(k)}, \tilde{d}^{(k)})$.

Following Agarwal et al. (2021), the compatible function approximation error can be decomposed as

$$L_Q(w^{(k)}, \theta^{(k)}, \tilde{d}^{(k)}) = \underbrace{L_Q(w^{(k)}, \theta^{(k)}, \tilde{d}^{(k)}) - L_Q(w_\star^{(k)}, \theta^{(k)}, \tilde{d}^{(k)})}_{\text{Statistical error (excess risk)}} + \underbrace{L_Q(w_\star^{(k)}, \theta^{(k)}, \tilde{d}^{(k)})}_{\text{Approximation error}}.$$

The statistical error measures how accurate is our solution to the regression problem, i.e., how good $w^{(k)}$ is compared with $w_\star^{(k)}$. The approximation error measures the best possible solution for approximating $Q_{s,a}(\theta^{(k)})$ using $\phi_{s,a}$ as features in the regression problem (modeling error). One way to proceed with the analysis is to assume that both the statistical error and the approximation error are bounded for all iterations, which is the approach we take in Section 5.4.2 and is also the approach we take later in Section 5.5 for the analysis of the NPG method.

However, in Section 5.4.1, we first take an alternative approach proposed by Agarwal et al. (2021), where the assumption of bounded approximation error is replaced by a bounded *transfer error*. The transfer error refers to $L_Q(w_\star^{(k)}, \theta^{(k)}, \tilde{d}^*)$, where the iteration-dependent visitation distribution $\tilde{d}^{(k)}$ is shifted to a fixed one $\tilde{d}^*$ (defined in Section 5.4.1).

These two approaches require different additional assumptions and result in slightly different convergence rates. Here we first state the common assumption on the bounded statistical error.

**Assumption 5.1** (Bounded statistical error, Assumption 6.1.1 in Agarwal et al. (2021))**.** *There exists $\epsilon_{\text{stat}} > 0$ such that for all iterations $k \geq 0$ of the Q-NPG method (5.19), we have*

$$\mathbb{E}\left[L_Q(w^{(k)}, \theta^{(k)}, \tilde{d}^{(k)}) - L_Q(w_\star^{(k)}, \theta^{(k)}, \tilde{d}^{(k)})\right] \leq \epsilon_{\text{stat}}. \tag{5.20}$$

By solving the regression problem with sampling based approaches, we can expect $\epsilon_{\text{stat}} = \mathcal{O}(1/\sqrt{T})$ (Agarwal et al., 2021) or $\epsilon_{\text{stat}} = \mathcal{O}(1/T)$ (see Corollary 5.11) where $T$ is the number of iterations used to find the inexact solution $w^{(k)}$.

### 5.4.1 Analysis with bounded transfer error

Here we introduce some additional notation. For any state distributions $p, q \in \Delta(\mathcal{S})$, we define the *distribution mismatch coefficient* of $p$ relative to $q$ as

$$\left\| \frac{p}{q} \right\|_\infty \stackrel{\text{def}}{=} \max_{s \in \mathcal{S}} \frac{p_s}{q_s}.$$

Let $\pi^*$ be an arbitrary *comparator policy*, which is not necessarily an optimal policy and does not need to belong to the log-linear policy class. Fix a state distribution $\rho \in \Delta(\mathcal{S})$. We denote $d^{\pi^*}(\rho)$ as $d^*$ and $d^{\pi(\theta^{(k)})}(\rho)$ as $d^{(k)}$, and define the following distribution mismatch coefficients:

$$\vartheta_k \stackrel{\text{def}}{=} \left\| \frac{d^*}{d^{(k)}} \right\|_\infty \stackrel{(5.6)}{\leq} \frac{1}{1-\gamma} \left\| \frac{d^*}{\rho} \right\|_\infty \qquad \text{and} \qquad \vartheta_\rho \stackrel{\text{def}}{=} \frac{1}{1-\gamma} \left\| \frac{d^*}{\rho} \right\|_\infty \geq \frac{1}{1-\gamma}. \tag{5.21}$$

Thus, for all $k \geq 0$, we have $\vartheta_k \leq \vartheta_\rho$. We assume that $\vartheta_\rho < \infty$, which is the case, for example, if $\rho_s > 0$ for all $s \in \mathcal{S}$. This is commonly used in the literature on policy gradient methods (e.g., Zhang et al., 2020a; Wang et al., 2020) and the NPG convergence analysis (e.g., Cayci et al., 2021; Xiao, 2022). We further relax this condition in Appendix D.6.1.

We also introduce a weighted KL divergence given by

$$D_k^* \stackrel{\text{def}}{=} \mathbb{E}_{s \sim d^*} \left[ D(\pi_s^*, \pi_s^{(k)}) \right].$$

If we choose the uniform initial policy, i.e., $\pi_{s,a}^{(0)} = 1/|\mathcal{A}|$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$ (or $\theta^{(0)} = 0$), then $D_0^* \leq \log |\mathcal{A}|$ for all $\rho \in \Delta(\mathcal{S})$ and for any $\pi^* \in \Delta(\mathcal{A})^{\mathcal{S}}$. The choice of the step size will directly depend on $D_0^*$ in our forthcoming linear convergence results.

Given a state distribution $\rho$ and a comparator policy $\pi^*$, we define a state-action measure $\tilde{d}^*$ as

$$\tilde{d}_{s,a}^* \stackrel{\text{def}}{=} d_s^* \cdot \text{Unif}_{\mathcal{A}}(a) \stackrel{\text{def}}{=} \frac{d_s^*}{|\mathcal{A}|}, \tag{5.22}$$

and use it to express the transfer error as $L_Q(w_\star^{(k)}, \theta^{(k)}, \tilde{d}^*)$.

**Assumption 5.2** (Bounded transfer error, Assumption 6.1.2 in Agarwal et al. (2021))**.** *There exists $\epsilon_{\text{bias}} > 0$ such that for all iterations $k \geq 0$ of the Q-NPG method (5.19), we have*

$$\mathbb{E}\left[ L_Q(w_\star^{(k)}, \theta^{(k)}, \tilde{d}^*) \right] \leq \epsilon_{\text{bias}}. \tag{5.23}$$

The $\epsilon_{\text{bias}}$ is often referred to as the transfer error, since it is the error due to replacing the relevant distribution $\tilde{d}^{(k)}$ by $\tilde{d}^*$. This transfer error bound characterizes how well the Q-values

can be linearly approximated by the feature maps $\phi_{s,a}$. It can be shown that $\epsilon_{\text{bias}} = 0$ when $\pi^{(k)}$ is the softmax tabular policy (Agarwal et al., 2021) or the MDP has a certain low-rank structure (Jiang et al., 2017; Yang and Wang, 2019; Yang and Wang, 2020; Jin et al., 2020). As mentioned in Agarwal et al. (2021, Remark 19), when $\epsilon_{\text{bias}} = 0$, one can easily verify that the NPG and Q-NPG are equivalent algorithms. For rich neural parametrizations, $\epsilon_{\text{bias}}$ can be made small (Wang et al., 2020).

The next assumption concerns the relative condition number between two covariance matrices of $\phi_{s,a}$ defined under different state-action distributions.

**Assumption 5.3** (Bounded relative condition number, Assumption 6.2 in Agarwal et al. (2021))**.** *Fix a state distribution $\rho$, a state-action distribution $\nu$ and a comparator policy $\pi^*$. Let*

$$\Sigma_{\tilde{d}^*} \overset{def}{=} \mathbb{E}_{(s,a)\sim\tilde{d}^*}\left[\phi_{s,a}\phi_{s,a}^\top\right], \qquad \text{and} \qquad \Sigma_\nu \overset{def}{=} \mathbb{E}_{(s,a)\sim\nu}\left[\phi_{s,a}\phi_{s,a}^\top\right], \tag{5.24}$$

*where $\tilde{d}^*$ is specified in (5.22). We define the relative condition number between $\Sigma_{\tilde{d}^*}$ and $\Sigma_\nu$ as*

$$\kappa_\nu \overset{def}{=} \max_{w\in\mathbb{R}^m} \frac{w^\top \Sigma_{\tilde{d}^*} w}{w^\top \Sigma_\nu w}, \tag{5.25}$$

*and assume that $\kappa_\nu$ is finite.*

The $\kappa_\nu$ is referred to as the relative condition number, since the ratio is between two different matrix induced norm. Notice that Assumption 5.3 benefits from the use of $\nu$. In fact, it is shown in Agarwal et al. (2021, Remark 22 and Lemma 23) that $\kappa_\nu$ can be reasonably small (e.g., $\kappa_\nu \leq m$ is always possible) and independent to the size of the state space by controlling $\nu$.

Our analysis also needs the following assumption, which does not appear in Agarwal et al. (2021).

**Assumption 5.4** (Concentrability coefficient for state visitation)**.** *There exists a finite $C_\rho > 0$ such that for all iterations $k \geq 0$ of the Q-NPG method (5.19), it holds that*

$$\mathbb{E}_{s\sim d^*}\left[\left(\frac{d_s^{(k)}}{d_s^*}\right)^2\right] \leq C_\rho. \tag{5.26}$$

The concentrability coefficient is studied in the analysis of approximate dynamic programming algorithms (Munos, 2003; Munos, 2005; Munos and Szepesvári, 2008). It measures how much $\rho$ can get amplified in $k$ steps as compared to the reference distribution $d_s^*$. Let

$\rho_{\min} = \min_{s\in\mathcal{S}} \rho_s$. A sufficient condition for Assumption 5.4 to hold is that $\rho_{\min} > 0$. Indeed,

$$\sqrt{\mathbb{E}_{s\sim d^*}\left[\left(\frac{d_s^{(k)}}{d_s^*}\right)^2\right]} \le \left\|\frac{d^{(k)}}{d^*}\right\|_\infty \overset{(5.6)}{\le} \frac{1}{1-\gamma}\left\|\frac{d^{(k)}}{\rho}\right\|_\infty \le \frac{1}{(1-\gamma)\rho_{\min}}. \tag{5.27}$$

In reality, $\sqrt{C_\rho}$ can be much smaller than the pessimistic bound shown above. This is especially the case if we choose $\pi^*$ to be the optimal policy and $d^{(k)} \to d^*$. We further replace $C_\rho$ by $C_\nu$ defined in Section 5.4.2 that is independent to $\rho$ and thus is more easily satisfied.

Now we present our first main result.

**Theorem 5.5.** *Fix a state distribution $\rho$, an state-action distribution $\nu$ and a comparator policy $\pi^*$. We consider the Q-NPG method (5.19) with the step sizes satisfying $\eta_0 \ge \frac{1-\gamma}{\gamma} D_0^*$ and $\eta_{k+1} \ge \frac{1}{\gamma}\eta_k$. Suppose that Assumptions 5.1, 5.2, 5.3 and 5.4 all hold. Then we have for all $k \ge 0$,*

$$\mathbb{E}\left[V_\rho(\pi^{(k)})\right] - V_\rho(\pi^*) \le \left(1 - \frac{1}{\vartheta_\rho}\right)^k \frac{2}{1-\gamma} + \frac{2\sqrt{|\mathcal{A}|}\left(\vartheta_\rho\sqrt{C_\rho}+1\right)}{1-\gamma}\left(\sqrt{\frac{\kappa_\nu}{1-\gamma}\epsilon_{\text{stat}}} + \sqrt{\epsilon_{\text{bias}}}\right).$$

The main differences between our Theorem 5.5 and Theorem 20 of Agarwal et al. (2021), which is their corresponding result on the inexact Q-NPG method, are summarized as follows.

- The convergence rate of Agarwal et al. (2021, Theorem 20) is $\mathcal{O}(1/\sqrt{k})$ up to an error floor determined by $\epsilon_{\text{stat}}$ and $\epsilon_{\text{bias}}$. We have linear convergence up to an error floor that also depends on $\epsilon_{\text{stat}}$ and $\epsilon_{\text{bias}}$. However, the magnitude of our error floor is worse (larger) by a factor of $\vartheta_\rho\sqrt{C_\rho}$, due to the concentrability and the distribution mismatch coefficients used in our proof. A very pessimistic bound on this factor is as large as $|\mathcal{S}|^2/(1-\gamma)^2$.
- In terms of required conditions, both results use Assumptions 5.1, 5.2 and 5.3. Agarwal et al. (2021, Theorem 20) further assume that the norms of the feature maps $\phi_{s,a}$ are uniformly bounded and $w^{(k)}$ has a bounded norm (e.g., obtained by a projected stochastic gradient descent). Due to different analysis techniques referred next, we avoid such boundedness assumptions but rely on the concentrability coefficient $C_\rho$ defined in Assumption 5.4.
- Agarwal et al. (2021, Theorem 20) uses a diminishing step size $\eta \propto 1/\sqrt{k}$ where $k$ is the total number of iterations, but we use a geometrically increasing step size $\eta_k \propto \gamma^k$ for all $k \ge 0$. This discrepancy reflects the different analysis techniques adopted. The key analysis tool in Agarwal et al. (2021) is a *NPG Regret Lemma* (their Lemma 34) which relies on the smoothness of the functions $\log \pi_{s,a}(\theta)$ (thus the boundedness of $\|\phi_{s,a}\|$) and the boundedness of $\|w^{(k)}\|$, and thus the classical $\mathcal{O}(1/\sqrt{k})$ diminishing step size in the optimization literature. Our analysis exploits the three-point descent lemma (Chen and

Teboulle, 1993) and the performance difference lemma (Kakade and Langford, 2002), without reliance on smoothness parameters. As a consequence, we can take advantage of exponentially growing step sizes and avoid assuming the boundedness of $\|\phi_{s,a}\|$ or $\|w^{(k)}\|$.

Using increasing step size induces fast linear convergence. The reason is that Q-NPG behaves more and more like policy iteration with large enough step size. Intuitively, when $\eta_k \to \infty$ and $Q_s(\theta^{(k)})$ is equal to the linear approximation $\Phi_s w^{(k)}$ which is the case of the linear MDP (Jin et al., 2020) with $\epsilon_{\text{bias}} = 0$, (5.17) becomes

$$\pi_s^{(k+1)} = \arg\min_{p \in \Delta(\mathcal{A})} \left\{ \left\langle Q_s(\theta^{(k)}), p \right\rangle \right\}, \quad \forall s \in \mathcal{S},$$

which is exactly the classical Policy Iteration method (e.g., Puterman, 1994; Bertsekas, 2012). Thus, Q-NPG can match the linear convergence rate of policy iteration in this case. We refer to Xiao (2022, Section 4.4) for more discussion on the connection with policy iteration.

As a by product, we also obtain a sublinear $\mathcal{O}(1/k)$ convergence result while using arbitrary constant step size.

**Theorem 5.6.** *Fix a state distribution $\rho$, an state-action distribution $\nu$ and an* optimal policy *$\pi^*$. We consider the Q-NPG method (5.19) with any constant step size $\eta_k = \eta > 0$. Suppose that Assumptions 5.1, 5.2, 5.3 and 5.4 all hold. Then we have for all $k \geq 0$,*

$$\frac{1}{k}\sum_{t=0}^{k-1} \mathbb{E}\left[V_\rho(\pi^{(t)})\right] - V_\rho(\pi^*) \leq \frac{1}{(1-\gamma)k}\left(\frac{D_0^*}{\eta} + 2\vartheta_\rho\right) + \frac{2\sqrt{|\mathcal{A}|}\left(\vartheta_\rho\sqrt{C_\rho}+1\right)}{1-\gamma}\left(\sqrt{\frac{\kappa_\nu}{1-\gamma}\epsilon_{\text{stat}}} + \sqrt{\epsilon_{\text{bias}}}\right).$$

A deviation from the setting of Theorem 5.5 is that here we require $\pi^*$ to be an optimal policy[3]. Compared to Theorem 20 in Agarwal et al. (2021), our convergence rate is also sublinear, but with an improved convergence rate of $\mathcal{O}(1/k)$, as opposed to $\mathcal{O}(1/\sqrt{k})$. Moreover, they use a diminishing step size of order $\mathcal{O}(1/\sqrt{k})$ while our constant step size is unconstrained.

### 5.4.2 Analysis with bounded approximation error

In this section, instead of assuming bounded transfer error, we provide a convergence analysis based on the usual notion of approximation error and a weaker concentrability coefficient.

[3] In our analysis, we need to drop the positive term $\mathbb{E}\left[V_\rho(\theta^{(k)}) - V_\rho(\pi^*)\right]$ to obtain a lower bound, thus require $\pi^*$ to be an optimal policy.

**Assumption 5.7** (Bounded approximation error). *There exists $\epsilon_{\text{approx}} > 0$ such that for all iterations $k \geq 0$ of the Q-NPG method (5.19), it holds that*

$$\mathbb{E}\left[L_Q(w_\star^{(k)}, \theta^{(k)}, \tilde{d}^{(k)})\right] \leq \epsilon_{\text{approx}}. \tag{5.28}$$

As mentioned in Agarwal et al. (2021), Assumption 5.7 is stronger than Assumption 5.2 (bounded transfer error). Indeed,

$$L_Q(w_\star^{(k)}, \theta^{(k)}, \tilde{d}^*) \leq \left\|\frac{\tilde{d}^*}{\tilde{d}^{(k)}}\right\|_\infty L_Q(w_\star^{(k)}, \theta^{(k)}, \tilde{d}^{(k)}) \overset{(5.6)}{\leq} \frac{1}{1-\gamma}\left\|\frac{\tilde{d}^*}{\nu}\right\|_\infty L_Q(w_\star^{(k)}, \theta^{(k)}, \tilde{d}^{(k)}).$$

**Assumption 5.8** (Concentrability coefficient for state-action visitation). *There exists $C_\nu < \infty$ such that for all iterations of the Q-NPG method (5.19), we have*

$$\mathbb{E}_{(s,a)\sim\tilde{d}^{(k)}}\left[\left(\frac{h_{s,a}^{(k)}}{\tilde{d}_{s,a}^{(k)}}\right)^2\right] \leq C_\nu, \tag{5.29}$$

*where $h_{s,a}^{(k)}$ represents all of the following quantities:*

$$d_s^{(k+1)}\pi_{s,a}^{(k+1)}, \qquad d_s^{(k+1)}\pi_{s,a}^{(k)}, \qquad d_s^*\pi_{s,a}^{(k)}, \qquad \text{and} \quad d_s^*\pi_{s,a}^*. \tag{5.30}$$

Since we are free to choose $\nu$ independently of $\rho$, we can choose $\nu_{s,a} > 0$ for all $(s,a) \in \mathcal{S}\times\mathcal{A}$ for Assumption 5.8 to hold. Indeed, with $\nu_{\min}$ denoting $\min_{(s,a)\in\mathcal{S}\times\mathcal{A}} \nu_{s,a}$, we have

$$\sqrt{\mathbb{E}_{(s,a)\sim\tilde{d}^{(k)}}\left[\left(\frac{h_{s,a}^{(k)}}{\tilde{d}_{s,a}^{(k)}}\right)^2\right]} \leq \max_{(s,a)\in\mathcal{S}\times\mathcal{A}} \frac{h_{s,a}^{(k)}}{\tilde{d}_{s,a}^{(k)}} \overset{(5.6)}{\leq} \frac{1}{(1-\gamma)\nu_{\min}}, \tag{5.31}$$

where the upper bound can be smaller than that in (5.27) if $\rho_{\min}$ is smaller than $\nu_{\min}$.

**Theorem 5.9.** *Fix a state distribution $\rho$, an state-action distribution $\nu$ and a comparator policy $\pi^*$. We consider the Q-NPG method (5.19) with the step sizes satisfying $\eta_0 \geq \frac{1-\gamma}{\gamma} D_0^*$ and $\eta_{k+1} \geq \frac{1}{\gamma}\eta_k$. Suppose that Assumptions 5.1, 5.7 and 5.8 hold. Then we have for all $k \geq 0$,*

$$\mathbb{E}\left[V_\rho(\pi^{(k)})\right] - V_\rho(\pi^*) \leq \left(1 - \frac{1}{\vartheta_\rho}\right)^k \frac{2}{1-\gamma} + \frac{2\sqrt{C_\nu}\,(\vartheta_\rho + 1)}{1-\gamma}\left(\sqrt{\epsilon_{\text{stat}}} + \sqrt{\epsilon_{\text{approx}}}\right).$$

Compared to Theorem 5.5, while the approximation error assumption is stronger than the transfer error assumption, we do not require the assumption on relative condition number $\kappa_\nu$ and the error floor does not depends on $\kappa_\nu$ nor explicitly on $|\mathcal{A}|$. Besides, we can always choose $\nu$ so that the concentrability coefficient $C_\nu$ is finite even if $C_\rho$ is unbounded. However, it is not clear if Theorem 5.9 is better than Theorem 5.5.

**Remark 5.10.** *Note that Theorems 5.5, 5.6 and 5.9 benefit from using the visitation distribution $\tilde{d}^{(k)}$ instead of $\bar{d}^{(k)}$ (i.e., benefit from using $\nu$ instead of $\rho$). In particular, from (5.6), $\tilde{d}^{(k)}$ has a lower bound that is independent to the policy $\pi^{(k)}$ or $\rho$. This property allows us to define a weak notion of relative condition number (Assumption 5.3) that is independent to the iterates, and also get a finite upper bound of $C_\nu$ (Assumption 5.8 and (5.31)) that is independent to $\rho$.*

### 5.4.3 Sample complexity of Q-NPG

The previous results focus on iteration complexity, i.e., number of iterations used for updating $\theta$. Here we establish the sample complexity results, i.e., total number of samples of single-step interaction with the environment, of a sample-based Q-NPG method (Algorithm 12 in Appendix D.3). Combined with a simple stochastic gradient descent (SGD) solver, `Q-NPG-SGD` in Algorithm 16, the following corollary shows that Algorithm 12 converges globally by further assuming that the feature map is bounded and has non-singular covariance matrix.

**Corollary 5.11.** *Consider the setting of Theorem 5.9. Suppose that the sample-based Q-NPG Algorithm 12 is run for $K$ iterations, with $T$ gradient steps of `Q-NPG-SGD` (Algorithm 16) per iteration. Furthermore, suppose that for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, we have $\|\phi_{s,a}\| \le B$ with $B > 0$, and we choose the step size $\alpha = \frac{1}{2B^2}$ and the initialization $w_0 = 0$ for `Q-NPG-SGD`. If for all $\theta \in \mathbb{R}^m$, the covariance matrix of the feature map followed by the initial state-action distribution $\nu$ satisfies*

$$\mathbb{E}_{(s,a)\sim\nu}\left[\phi_{s,a}\phi_{s,a}^\top\right] \overset{(5.24)}{=} \Sigma_\nu \;\ge\; \mu \mathbf{I}_m, \tag{5.32}$$

*where $\mathbf{I}_m \in \mathbb{R}^{m\times m}$ is the identity matrix and $\mu > 0$, then*

$$\mathbb{E}\left[V_\rho(\pi^{(K)})\right] - V_\rho(\pi^*) \le \left(1 - \frac{1}{\vartheta_\rho}\right)^K \frac{2}{1-\gamma} + \frac{2\left(\vartheta_\rho + 1\right)\sqrt{C_\nu \epsilon_{\text{approx}}}}{1-\gamma} + \frac{4\sqrt{C_\nu}\left(\vartheta_\rho + 1\right)}{(1-\gamma)^3\sqrt{T}}\left(\frac{B^2}{\mu}\left(\sqrt{2m}+1\right) + (1-\gamma)\sqrt{2m}\right).$$

In `Q-NPG-SGD`, each trajectory has the expected length $1/(1-\gamma)$ (see Lemma D.5). Consequently, with $K = \mathcal{O}(\log(1/\epsilon)\log(1/(1-\gamma)))$ and $T = \mathcal{O}(\frac{1}{(1-\gamma)^6\epsilon^2})$, Q-NPG requires $K * T/(1-$

$\gamma) = \tilde{\mathcal{O}}(\frac{1}{(1-\gamma)^7\epsilon^2})$ samples such that $\mathbb{E}\left[V_\rho(\pi^{(K)})\right] - V_\rho(\pi^*) \leq \mathcal{O}(\epsilon) + \mathcal{O}(\frac{\sqrt{\epsilon_{\text{approx}}}}{1-\gamma})$. The $\tilde{\mathcal{O}}(1/\epsilon^2)$ sample complexity matches with the one of value-based algorithms such as Q-learning (Li et al., 2020) and also matches with the one of model-based algorithms such as policy iteration (Puterman, 1994; Lazaric et al., 2016) .

Compared to Agarwal et al. (2021, Corollary 26) for the sampled based Q-NPG Algorithm 12, their sample complexity is $\mathcal{O}(\frac{1}{(1-\gamma)^{11}\epsilon^6})$ with $K = \frac{1}{(1-\gamma)^2\epsilon^2}$ and $T = \frac{1}{(1-\gamma)^8\epsilon^4}$. Despite the improvement on the convergence rate for $K$, they use the optimization results of Shalev-Shwartz and Ben-David (2014, Theorem 14.8) to obtain $\epsilon_{\text{stat}} = \mathcal{O}(1/\sqrt{T})$, while we use the one of Bach and Moulines (2013, Theorem 1) (see Theorem D.15 as well) to establish faster $\epsilon_{\text{stat}} = \mathcal{O}(1/T)$[4]. With further regularity (5.32), Agarwal et al. (2021) mentioned that $\epsilon_{\text{stat}} = \mathcal{O}(1/T)$ can also be achieved through Hsu et al. (2012, Theorem 16). In addition, Agarwal et al. (2021) use the projected SGD method and require that the stochastic gradient is bounded which is incorrectly verified in their proof [5]. In contrast, to apply Theorem D.15, we avoid proving the boundedness of the stochastic gradient. Alternatively, we require a different condition (5.32). A proof sketch of our corollary is provided in Appendix D.4.5 for more details.

As for the condition (5.32), it is shown in Cayci et al. (2021, Proposition 3) that with $\nu$ chosen as uniform distribution over $\mathcal{S} \times \mathcal{A}$ and $\phi_{s,a} \sim \mathcal{N}(0, \mathbf{I}_m)$ sampled as Gaussian random features, (5.32) is guaranteed with high probability. More generally, with $m \ll |\mathcal{S}||\mathcal{A}|$, it is easy to find $m$ linearly independent $\phi_{s,a}$ among all $|\mathcal{S}||\mathcal{A}|$ features such that the covariance matrix $\Sigma_\nu$ has full rank. This is a common requirement for linear function approximation settings (Tsitsiklis and Van Roy, 1996; Melo et al., 2008; Sutton et al., 2009).

## 5.5 Analysis of NPG with log-linear policies

We now return to the convergence analysis of the inexact NPG method, specifically,

$$\theta^{(k+1)} = \theta^{(k)} - \eta_k w^{(k)}, \qquad w^{(k)} \approx \operatorname{argmin}_w L_A(w, \theta^{(k)}, \tilde{d}^{(k)}), \tag{5.33}$$

where $\tilde{d}^{(k)}$ is a shorthand for $\tilde{d}^{\pi(\theta^{(k)})}(\nu)$ and $\nu \in \Delta(\mathcal{S}\times\mathcal{A})$ is an arbitrary state-action distribution that does not depend on $\rho$. Again, let $w_\star^{(k)} \in \operatorname{argmin}_w L_A(w, \theta^{(k)}, \tilde{d}^{(k)})$ denote the minimizer. Our analysis of NPG is analogous to that of Q-NPG shown in the previous section. That is, we again exploit the inexact PMD formulation (5.18) and use techniques developed in Xiao (2022).

[4] We are aware that Agarwal et al. (2021, Corollary 6.10) also use Bach and Moulines (2013, Theorem 1) in an early version https://arxiv.org/pdf/1908.00261v2.pdf to obtain $\epsilon_{\text{stat}} = \mathcal{O}(1/T)$.

[5] Indeed, the stochastic gradient of $L_Q$ is unbounded, since the estimate $\widehat{Q}_{s,a}(\theta)$ of $Q_{s,a}(\theta)$ is unbounded. This is because each single sampled trajectory has unbounded length. See Appendix D.4.5 for more explanations.

The set of assumptions we use for NPG is analogous to the assumptions used in Section 5.4.2. In particular, we assume a bounded approximation error instead of transfer error (c.f., Assumption 5.2) in minimizing $L_A$ and do not need the assumption on relative condition number.

**Assumption 5.12** (Bounded statistical error, Assumption 6.5.1 in Agarwal et al. (2021))**.** *There exists $\epsilon_{\text{stat}} > 0$ such that for all iterations $k \geq 0$ of the NPG method (5.33), we have*

$$\mathbb{E}\left[L_A(w^{(k)}, \theta^{(k)}, \tilde{d}^{(k)}) - L_A(w_\star^{(k)}, \theta^{(k)}, \tilde{d}^{(k)})\right] \leq \epsilon_{\text{stat}}. \tag{5.34}$$

**Assumption 5.13** (Bounded approximation error)**.** *There exists $\epsilon_{\text{approx}} > 0$ such that for all iterations $k \geq 0$ of the NPG method (5.33), we have*

$$\mathbb{E}\left[L_A(w_\star^{(k)}, \theta^{(k)}, \tilde{d}^{(k)})\right] \leq \epsilon_{\text{approx}}. \tag{5.35}$$

**Assumption 5.14** (Concentrability coefficient for state-action visitation)**.** *There exists $C_\nu < \infty$ such that for all iterations $k \geq 0$ of the NPG method (5.33), we have*

$$\mathbb{E}_{(s,a)\sim\tilde{d}^{(k)}}\left[\left(\frac{\bar{d}_{s,a}^{(k+1)}}{\tilde{d}_{s,a}^{(k)}}\right)^2\right] \leq C_\nu \quad \text{and} \quad \mathbb{E}_{(s,a)\sim\tilde{d}^{(k)}}\left[\left(\frac{\bar{d}_{s,a}^{\pi^*}}{\tilde{d}_{s,a}^{(k)}}\right)^2\right] \leq C_\nu. \tag{5.36}$$

Under the above assumptions, we have the following result.

**Theorem 5.15.** *Fix a state distribution $\rho$, a state-action distribution $\nu$, and a comparator policy $\pi^*$. We consider the NPG method (5.33) with the step sizes satisfying $\eta_0 \geq \frac{1-\gamma}{\gamma} D_0^*$ and $\eta_{k+1} \geq \frac{1}{\gamma}\eta_k$. Suppose that Assumptions 5.12, 5.13 and 5.14 hold. Then we have for all $k \geq 0$,*

$$\mathbb{E}\left[V_\rho(\pi^{(k)})\right] - V_\rho(\pi^*) \leq \left(1 - \frac{1}{\vartheta_\rho}\right)^k \frac{2}{1-\gamma} + \frac{\sqrt{C_\nu}\,(\vartheta_\rho + 1)}{1-\gamma}\left(\sqrt{\epsilon_{\text{stat}}} + \sqrt{\epsilon_{\text{approx}}}\right).$$

Compared to Theorem 5.9, our convergence guarantees for Q-NPG and NPG have the same convergence rate and error floor, and the same type of assumptions.

Now we compare Theorem 5.15 with Theorem 29 in Agarwal et al. (2021) for the NPG analysis. The main differences are similar to those for Q-NPG as summarized right after

Theorem 5.5: Their convergence rate is sublinear while ours is linear; they assume uniformly bounded $\phi_{s,a}$ and $w^{(k)}$ while we require bounded concentrability coefficient $C_\nu$ due to different proof techniques; they use diminishing step sizes and we use geometrically increasing ones. Moreover, Theorem 5.15 requires bounded approximation error, which is a stronger assumption than the bounded transfer error used by their Theorem 29, but we do not need the assumption on bounded relative condition number.

We note that the bounded relative condition number required by Agarwal et al. (2021, Theorem 29) must hold for the covariance matrix of $\bar{\phi}_{s,a}^{(k)}$ for all $k \geq 0$ because the centered feature maps $\bar{\phi}_{s,a}^{(k)}$ depends on the iterates $\theta^{(k)}$. This is in contrast to our Assumption 5.3, where we use a single fixed covariance matrix for Q-NPG that is independent to the iterates, as defined in (5.24).

In addition, the inequalities in (5.36) only involve half of the state-action visitation distributions listed in (5.30), i.e., the first and the fourth terms. From (5.31), the upper bound of $C_\nu$ is obtained only through (5.6), which is a property of $\tilde{d}^\pi$ itself for all policy $\pi \in \Delta(\mathcal{A})^{\mathcal{S}}$. Thus, $C_\nu$ in (5.36) can share the same upper bound in (5.31) independent to the use of the algorithm Q-NPG or NPG. Consequently, our concentrability coefficient assumption is weaker than Assumption 2 in Cayci et al. (2021) which studies the linear convergence of NPG with entropy regularization for the log-linear policy class. The reason is that the bound on $C_\nu$ in (5.31) does not depend on the policies throughout the iterations thanks to the use of $\tilde{d}^{(k)}$ instead of $\bar{d}^{(k)}$ (see Remark 5.10 as well). See Appendix D.6.2 for a thorough discussion on the concentrability coefficient $C_\nu$.

Similar to Theorem 5.6, we also obtain a sublinear rate for NPG while using an unconstrained constant step size.

**Theorem 5.16.** *Fix a state distribution $\rho$, an state-action distribution $\nu$ and an* optimal policy *$\pi^*$. We consider the NPG method (5.33) with any constant step size $\eta_k = \eta > 0$. Suppose that Assumptions 5.12, 5.13 and 5.14 hold. Then we have for all $k \geq 0$,*

$$\frac{1}{k}\sum_{t=0}^{k-1} \mathbb{E}\left[V_\rho(\pi^{(t)})\right] - V_\rho(\pi^*) \leq \frac{1}{(1-\gamma)k}\left(\frac{D_0^*}{\eta} + 2\vartheta_\rho\right) + \frac{\sqrt{C_\nu}\,(\vartheta_\rho + 1)}{1-\gamma}\left(\sqrt{\epsilon_{\text{stat}}} + \sqrt{\epsilon_{\text{approx}}}\right).$$

Compared to Theorem 5.6, again here we require $\pi^*$ to be an optimal policy for the same reason as indicated in Footnote 3. Furthermore our sublinear convergence guarantees for both Q-NPG and NPG are the same. Compared to Theorem 29 in Agarwal et al. (2021), the main differences are also similar to those for Q-NPG as summarized right after Theorem 5.6: our

convergence rate improves from $\mathcal{O}(1/\sqrt{k})$ to $\mathcal{O}(1/k)$; they use a diminishing step size of order $\mathcal{O}(1/\sqrt{k})$ while we can take any constant step size we want.

Despite the difference of using $\tilde{d}^{(k)}$ instead of $\bar{d}^{(k)}$ for the compatible function approximation $L_A(w^{(k)}, \theta^{(k)}, \tilde{d}^{(k)})$, notice that same sublinear convergence rate $\mathcal{O}(1/k)$ is established by Liu et al. (2020) for NPG with constant step size, while their step size is bounded by the inverse of a smoothness constant and they further require that the feature map is bounded and the Fisher information matrix (5.10) is strictly lower bounded for all parameters $\theta \in \mathbb{R}^m$ (see this condition later in (5.37)). With such additional conditions, we are able to provide a $\mathcal{O}(\frac{1}{(1-\gamma)^5\epsilon^2})$ sample complexity result of NPG next.

### 5.5.1 Sample complexity of NPG

Combined with a regression solver, `NPG-SGD` in Algorithm 15, which uses a slight modification of `Q-NPG-SGD` for the unbiased gradient estimates of $L_A$, we consider a sampled-based NPG Algorithm 11 proposed in Appendix D.3 and show its sample complexity result in the following corollary.

**Corollary 5.17.** *Consider the setting of Theorem 5.15. Suppose that the sample-based NPG Algorithm 11 is run for $K$ iterations, with $T$ gradient steps of `NPG-SGD` (Algorithm 15) per iteration. Furthermore, suppose that for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, we have $\|\phi_{s,a}\| \leq B$ with $B > 0$, and we choose the step size $\alpha = \frac{1}{8B^2}$ and the initialization $w_0 = 0$ for `NPG-SGD`. If for all $\theta \in \mathbb{R}^m$, the covariance matrix of the centered feature map induced by the policy $\pi(\theta)$ and the initial state-action distribution $\nu$ satisfies*

$$\mathbb{E}_{(s,a)\sim\tilde{d}^{\theta}}\left[\bar{\phi}_{s,a}(\theta)(\bar{\phi}_{s,a}(\theta))^{\top}\right] \;\geq\; \mu \mathbf{I}_m, \tag{5.37}$$

*where $\mathbf{I}_m \in \mathbb{R}^{m\times m}$ is the identity matrix and $\mu > 0$, then*

$$\mathbb{E}\left[V_\rho(\pi^{(K)})\right] - V_\rho(\pi^*) \leq \left(1 - \frac{1}{\vartheta_\rho}\right)^K \frac{2}{1-\gamma} + \frac{(\vartheta_\rho + 1)\sqrt{C_\nu \epsilon_{\text{approx}}}}{1-\gamma} + \frac{4\sqrt{C_\nu}\,(\vartheta_\rho+1)}{(1-\gamma)^2\sqrt{T}}\left(\frac{2B^2}{\mu}\left(\sqrt{2m}+1\right) + \sqrt{2m}\right).$$

Now we compare our Corollary 5.17 with Corollary 33 in Agarwal et al. (2021), which is their corresponding sample complexity results for NPG. The main differences between Corollary 5.17 and Corollary 33 in Agarwal et al. (2021) are similar to those for Q-NPG as summarized right after Corollary 5.11: Their sample complexity is $\mathcal{O}(\frac{1}{(1-\gamma)^{11}\epsilon^6})$ while ours is

$\tilde{\mathcal{O}}(\frac{1}{(1-\gamma)^5\epsilon^2})$; they consider a projection step for the iterates and incorrectly bound the stochastic gradient due to a similar error indicated in Footnote 5 (and see Appendix D.5.4 for more details), while we assume Fisher-non-degeneracy (5.37).

Compared to Corollary 5.11, the sample complexities for both Q-NPG and NPG are the same. The assumption (5.37) on the Fisher information matrix is much stronger than (5.32), as (5.32) is independent to the iterates. However, despite the difference of using $\nu$ instead of $\rho$, the Fisher-non-degeneracy (5.37) is commonly used in the optimization literature (Byrd et al., 2016; Gower et al., 2016; Wang et al., 2017) and in the RL literature (Liu et al., 2020; Ding et al., 2022; Yuan et al., 2022a). It characterizes that the Fisher information matrix behaves well as a preconditioner in the NPG update (5.9). Indeed, (5.37) is directly assumed to be positive definite in the pioneering NPG work (Kakade, 2001) and in the follow-up works on natural actor-critic algorithms (Peters and Schaal, 2008a; Bhatnagar et al., 2009). It is satisfied by a wide families of policies, including the Gaussian policy (Duan et al., 2016; Papini et al., 2018; Huang et al., 2020) and certain neural policy with log-linear policy as a special case. We refer to Liu et al. (2020, Section B.2) and Ding et al. (2022, Section 8) for more discussions on the Fisher-non-degenerate setting.

To prove Corollary 5.17, our approach is inspired from the proof of the sample complexity analysis of Liu et al. (2020, Theorem 4.9). That is, we require the Fisher-non-degeneracy (5.37) and apply Theorem D.15 to the minimization of function $L_A(w, \theta, \tilde{d}^\theta)$ without relying on the boundedness of the stochastic gradient. A proof sketch is provided in Appendix D.5.4. Compared to their result, they obtain worse $\mathcal{O}(\frac{1}{(1-\gamma)^7\epsilon^3})$ sample complexity for NPG due to a slower $\mathcal{O}(1/k)$ convergence rate.

## 5.6 Conclusion and discussion

In this chapter, for both NPG and Q-NPG methods applied for the log-linear policy, we establish the linear convergence results with non-adaptive geometrically increasing step sizes and the sublinear convergence results with arbitrary large constant step sizes. Our work is the first step of showing that the policy mirror descent proof techniques used in Xiao (2022) can be extended in function approximation regime.

The main focus of this chapter was the theoretical analysis of NPG method. The results we have obtained open up several experimental questions related to parameter settings for NPG and Q-NPG. We leave such questions as an important future work to further support our theoretical findings.

An interesting application from our work is to investigate the sample complexity of natural actor-critic with our PMD analysis. Indeed, our work obtains $w^{(k)}$ by a regression solver. One can also use temporal difference (TD) learning (e.g., Cayci et al. (2021), Chen and Theja

Maguluri (2022), and Telgarsky (2022)) with Markovian sampling to achieve similar $O(1/\epsilon^2)$ sample complexity result. The performance analysis of TD learning will be expressed for $\epsilon_{\text{stat}}$, which directly imply the total sample complexity results through our theorems.

One natural question is whether we can extend our analysis to the general policy classes. Here we provide one possible way. It can be extended by using a similar compatible function approximation framework. Concretely, consider the parameterized policy

$$\pi_{s,a}(\theta) = \frac{\exp(f_{s,a}(\theta))}{\sum_{a' \in \mathcal{A}} \exp(f_{s,a'}(\theta))},$$

where $f_{s,a}(\theta)$ is parameterized by $\theta \in \mathbb{R}^m$ and is differential. As Agarwal et al. (2021) mentioned, the gradient can be written as

$$\nabla_\theta \log \pi_{s,a}(\theta) = g_{s,a}(\theta) \quad \text{where} \quad g_{s,a}(\theta) = \nabla_\theta f_{s,a}(\theta) - \mathbb{E}_{a' \sim \pi_s(\theta)} \left[\nabla_\theta f_{s,a'}(\theta)\right].$$

The NPG update is equivalent to the following compatible function approximation framework

$$\theta^{(k+1)} = \theta^{(k)} - \eta_k w_\star^{(k)}, \qquad w_\star^{(k)} \in \arg\min_w \mathbb{E}_{(s,a)\sim \bar{d}^{(k)}} \left[\left(A_{s,a}(\theta^{(k)}) - w^\top g_{s,a}(\theta^{(k)})\right)^2\right].$$

As Alfano and Rebeschini (2022, Remark 4.8) mentioned, if we assume that for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, function $f(\theta)$ satisfies

$$f_{s,a}(\theta^{(k+1)}) = f_{s,a}(\theta^{(k)}) - \eta_k (w_\star^{(k)})^\top g_{s,a}(\theta^{(k)}),$$

which is the case for the log-linear policies, then one can easily verify that the NPG update resulted in a new policy is also equivalent to the policy mirror descent update

$$\pi_s^{(k+1)} = \arg\min_{p \in \Delta(\mathcal{A})} \left\{\eta_k \left\langle G_s^{(k)} w^{(k)}, p \right\rangle + D(p, \pi_s^{(k)})\right\}, \quad \forall s \in \mathcal{S},$$

where $G_s^{(k)} \in \mathbb{R}^{|\mathcal{A}| \times m}$ is a matrix with rows $(g_{s,a}(\theta^{(k)}))^\top \in \mathbb{R}^{1\times m}$ for $a \in \mathcal{A}$. Consequently, one can extend our work naturally in this general setting to derive linear convergence analysis for NPG. We refer to the recent work of Alfano et al. (2023) that follow this research direction and generalize our results to the general parametrization including the neural networks as special cases.

Perhaps one can consider the *exponential tilting*, a generalization of Softmax to more general probability distributions. Another interesting venue of investigation is to consider the *generalized linear model* instead of linear function approximation for the $Q$ function and the advantage function.

One interesting open question is that is there a way to increase stepsize when the discount factor is unknown. So far the PMD proof techniques used in Lan (2022) and Xiao (2022) and ours require that the discount factor is known. Perhaps the work of Li et al. (2022b) can help to find a way to increase stepsize when the discount factor is unknown. Indeed, Li et al. (2022b) consider the averaged MDP setting. So there is no discount factor. They achieve linear convergence for NPG by increasing the stepsize with some regularization parameters. It will be interesting to investigate if the way of increasing stepsize in Li et al. (2022b) can be applied in our setting.

# Chapter 6

# General Conclusion and Perspectives

## 6.1 About Stochastic Second Order Methods in Optimization

The first part of this thesis was devoted to design new globally convergent stochastic second order methods. We first developed a new Sketched Newton-Raphson (SNR) method (Algorithm 1) for solving large scale nonlinear equations by using the sketch-and-project techniques. We established the global convergence analysis of SNR by rewriting SNR as a variant of online stochastic gradient descent (SGD), and then leveraging proof techniques of SGD. SNR is so general that it accommodates the existing method (Stochastic Newton Method (SNM) (Rodomanov and Kropotov, 2016; Kovalev et al., 2019)), but also allows for the design of completely new methods, such as the Tossing-Coin-Sketch (TCS) method (Algorithm 5), the nonlinear Kaczmarz method (2.37) (Wang et al., 2022) and a variant of Stochastic Polyak method (Gower et al., 2021a). In particular, TCS is efficient, scales well with the number of samples for solving generalized linear models (GLMs), and is competitive as compared to classical variance reduced gradient methods.

Then, by adding adaptive norms for the projection, we extended SNR to a more general Sketched Newton-Raphson with Variable Metric (SNRVM) method (3.18), equipped with a global convergence theory. SNRVM is more general than SNR that it includes SNR and Randomized Subspace Newton method (Gower et al., 2019a) as special cases. Through the umbrella of SNRVM, we developed a new Stochastic Average Newton (SAN) method (Algorithm 2) and SANA (Algorithm 3) for solving finite sum optimization problems. In particular, SAN is incremental. That is, it samples only one single data point per iteration. SAN is efficient. It costs $\mathcal{O}(d)$ per iteration for GLMs with the dimension $d$ of the features. Last but not least, SAN requires no parameter tuning (e.g. step size), neither knowledge from the problem (no smoothness constant). As a result, SAN is also empirically highly competitive as compared to variance reduced gradient methods.

Overall, we presented a principled approach for designing stochastic Newton methods for solving both nonlinear equations and optimization problems. Our approach has two steps. First, we can re-write the nonlinear equations or the optimization problem as desired nonlinear equations. Second, we apply SNRVM to solve this system of nonlinear equations.

There are many ways to re-write the nonlinear equations into another nonlinear equations, such as introducing auxiliary variables and using function splitting or variable splitting formulation as we demonstrated for TCS, SNM, SAN and SAGA. Each re-write leads to a distinct method when using SNRVM. As such, we believe that SNRVM and its global convergence theory will open the way to designing and analyzing a host of new stochastic second order methods. This will be a very exciting direction for future work.

At the same time, there are also many ways to apply SNRVM by choosing different sketching matrices or by varying different norms for the projection step. For instance, one promising direction is to use new sophisticated sketching matrices, such as the Walsh-Hadamard matrix (Lu et al., 2013; Pilanci and Wainwright, 2016), the fast Johnson-Lindenstrauss sketch (Pilanci and Wainwright, 2017), sketches with determinantal sampling (Mutny et al., 2020), the count sketch (Clarkson and Woodruff, 2017) and the SubCount sketch (Gazagnadou et al., 2022), to design even faster variants of SNRVM or cover other stochastic second-order methods.

Perhaps the most exciting direction for future work is to extend SNRVM with regularization. Recently, there has been a surge of interest in developing variants of Newton's method and cubic Newton's method by adding regularization (Mishchenko, 2021; Doikov and Nesterov, 2021; Doikov et al., 2022) along with strong global convergence guarantees (e.g., $\mathcal{O}(1/k^2)$, $\mathcal{O}(1/k^3)$ which are much faster than $\mathcal{O}(1/k)$ convergence rate for gradient descent). However, these methods are deterministic, their stochastic forms for solving large-scale problems remain undiscovered. Thus there is still much to explore in designing new stochastic regularized Newton's method with the use of SNRVM. This will greatly extend SNRVM to much broader applications and keep versatile convergence guarantees.

## 6.2 About Finite Time Analysis of Policy Gradient Methods in Reinforcement Learning

In the reinforcement learning (RL) part of this thesis, we studied the finite time analysis of the vanilla policy gradient (PG) methods in Chapter 4 and natural policy gradient (NPG) methods in Chapter 5, respectively. We first adapted the modern proof techniques of SGD at the time of writing to establish a general but thorough finite time analysis of vanilla PG (Algorithm 4). The key assumption we used is the ABC assumption. This assumption allows us to unify much of the recent fragmented results in the RL literature and derive a better understanding for the hyperparameter choices of the PG algorithm. Combined with an additional (weak) gradient

domination assumption, we established the global optimum convergence results of vanilla PG and improved the current best known sample complexity results of the Fisher-non-degenerate policy from $\mathcal{O}(1/\epsilon^4)$ to $\mathcal{O}(1/\epsilon^3)$ for the stochastic vanilla PG.

One exciting future direction is that the generality of Theorem 4.4 opens the possibility to identify a broader set of configurations (i.e., MDP and policy space) for which PG is guaranteed to converge, notably thinking about settings such that the constant $A$ in the ABC assumption is *non-zero*, using additional assumptions such as the bijection assumptions based on the occupancy measure space (Zhang et al., 2020a; Zhang et al., 2021a) to not only get improved sample complexity for the global optimum convergence, but also allow a wider range of hyperparameter choices for the convergence. Another interesting future direction might be whether the ABC assumption analysis can be extended to the sample complexity analysis of the actor-critic (Yang et al., 2019; Kumar et al., 2021; Xu et al., 2020c). As mentioned in Section 4.5, Fatkhullin et al. (2022) naturally extend the SGD upper bound performance of Khaled and Richtárik (2023) and Yuan et al. (2022a). This might extend our PG analysis to even broader RL settings.

We then moved a step further to the finite time analysis of the (Q)-NPG methods (Algorithm 11 and 12), which are arguably one of the most important variants of the vanilla PG. Inspired from the proof techniques of Xiao (2022), we extend their linear convergence results of NPG from the softmax tabular policy to the log-linear policy with non-adaptive geometrically increasing step size. Our analysis relies on the reformulation of (Q)-NPG as the policy mirror descent (PMD) methods. The two main ingredients of our analysis are the three-point descent lemma (Lemma D.14) and the performance difference lemma (Lemma D.4). Thanks to the fast linear convergence results, we also established the fast $\tilde{\mathcal{O}}(1/\epsilon^2)$ sample complexity results for (Q)-NPG.

Our work of fast linear convergence analysis of NPG for the log-linear policies might inspire the current NPG convergence analysis in the function approximation regime from different perspectives. One important future direction is whether our proof techniques and the use of non-adaptive geometrically increasing step size can help improve the two-layer neural natural actor-critic (NAC) convergence analysis (Wang et al., 2020; Cayci et al., 2022a). Same interesting question can be asked for the linear MDP setting (Jin et al., 2020) to improve the analysis of Zanette et al. (2021) and Hu et al. (2022). As for the sample complexity analysis, it would be interesting to see if our work can improve the sample complexity analysis of NPG with Markovian sampling in Xu et al. (2020c) or the one with the off-policy sampling in Chen et al. (2022b).

Through our analysis of PG and NPG, the modern optimization results play an essential role. Our works are possible only because of the significant progress of these new improved optimization results. For instance, there has been extensive research on development of the

performance of SGD in recent years (Ghadimi and Lan, 2013; Zhou and Gu, 2019; Gower et al., 2019b; Khaled and Richtárik, 2023; Gower et al., 2021c; Arjevani et al., 2022; Zhou et al., 2022; Fatkhullin et al., 2022; Sa et al., 2022), including the lower bound and upper bound analysis of SGD performance. This provides a motivation that whether these SGD analysis can give some fresh understanding to RL algorithms. Some of these SGD analysis only deal with finite-sum structure, it might remains challenging to extend the analysis to the RL specific structure as we did for the vanilla PG.

It is noteworthy that all our analysis of PG and NPG are for the first-order stationary point (FOSP) convergence or for the global optimum convergence. It will be also interesting to investigate the performance of PG and NPG through the second-order stationary point convergence as Yang et al. (2021) did for the PG methods.

Recently, there has been a great success for variance reduced methods to improve the convergence rate of SGD, such as SVRG (Johnson and Zhang, 2013), SARAH (Nguyen et al., 2017), SPIDER (Fang et al., 2018), SpiderBoost (Wang et al., 2019), STORM (Cutkosky and Orabona, 2019), SNVRG (Zhou et al., 2020), PAGE (Li et al., 2021b), and more (Tran-Dinh et al., 2021). The RL community has then a great interest of applying these variance reduction techniques to improve the performance of PG methods, which results in SVRPG (Papini et al., 2018), SRVR-PG (Xu et al., 2020b), STORM-PG (Yuan et al., 2020) and ProxHSPGA (Pham et al., 2020). Leveraging these variance reduction techniques has led to an overall improved sample complexity of reaching FOSP. However, all these works require either the exact full gradient updates or large batch sizes per iteration. Liu et al. (2020) and Ding et al. (2022) establish the global optimum sample complexity analysis of SRVR-PG and STORM-PG, respectively. It will be interesting to understand whether the ABC assumption analysis can be applied to these algorithms and possibly allow for a wider range of hyperparameter choices, including the batch size. Furthermore, inspired from SARAH (Nguyen et al., 2017), Yang et al. (2022) and Huang et al. (2022) develop variance reduced versions of PMD. Similarly, Liu et al. (2020) also develop a variance reduced version of NPG. It will be interesting to consider other variance reduction techniques applied to PMD or NPG and see if our sample complexity analysis of PMD and NPG can help to improve their sample complexity analysis. When the gradient domination type assumptions are available, it will be interesting to rethink if we can achieve faster sample complexity for these variance reduced PG and NPG methods as we did for the vanilla PG. One of these cases was answered positively by the work of Fatkhullin et al. (2022) with their new variance reduced algorithm – PAGER.

On the other side, PG and NPG have been applied to a variety of domains beyond the common MDP setting. Understanding the limits of performance of these methods in different setting has become an important avenue of research in recent year. For instance, the performance of PG and NPG to achieve the Nash equilibrium is investigated in different game settings, such as the Markov potential games (Leonardos et al., 2022; Zhang et al., 2022b), zero-sum

matrix games (Cen et al., 2021b; Pattathil et al., 2022) and Markov games (Cen et al., 2021b; Pattathil et al., 2022; Zhang et al., 2022a; Cen et al., 2022). Other interesting settings for the analysis of PG and NPG include the multi-objective RL setup (Bai et al., 2021; Agarwal et al., 2022), the partially observed MDP (Cayci et al., 2022b) and the constrained MDP (Ding et al., 2020; Ying et al., 2022). The very interesting open question here is that whether the proof techniques we used for the analysis of PG and NPG in this thesis can immediately help improve the performance of these settings we just mentioned above.

## 6.3 Importing Stochastic Second-order Methods into RL

Finally, we would like to end this thesis by discussing the future direction where stochastic second-order methods can be applied into RL. This would bridge the seemingly independent Part I and Part II of the thesis together.

In spite of the success of PG and its variants, stochastic second-order methods applied into RL appears to be relatively lacking both in the literature and in practice. Intuitively, the use of the second-order information for the updates will help to accelerate the rate of convergence, hence improve the sample complexity of the algorithms. It might also require less parameter tuning, as compared to first-order algorithms. Recently, stochastic cubic regularized Newton has been applied to solve RL problems and achieved promising results (Masiha et al., 2022). Their global optimum convergence analysis benefits from the use of gradient domination property.

As presented in Part I of the thesis, we have a way to design many new stochastic second-order methods for solving optimization problems. It will be very interesting to see whether these second-order methods can be applied in RL. Furthermore, it will also be interesting to analysis these second-order methods using the proof techniques of PG, NPG or SNRVM.

# Chapter 7

# Introduction étendue en Français

## 7.1 Méthodes du Second Ordre Stochastiques en Optimisation

### 7.1.1 Contexte

**Optimisation en IA.** Au cours de la dernière décennie, nous avons témoigné le progrès de l'intelligence artificielle (IA), également appelée apprentissage automatique. Elle a été largement appliquée dans la société, du traitement automatique des langues, la vision par ordinateur, à la recommandation des publicités en ligne et même à la robotique, pour n'en citer que quelques-unes. Par exemple, dans le traitement automatique des langues, il existe des problèmes de traduction automatique, comme Google translate, et des problèmes de chatbot, comme ChatGPT. Dans le domaine de la vision par ordinateur, il existe des problèmes de segmentation, de classification d'images, et de détection d'objets, etc. En particulier, l'optimisation joue un rôle important dans l'IA. En fait, le problème de l'intérêt peut être formalisé sous la forme d'une fonction de perte $f$ de paramètre $w \in \mathbb{R}^d$ dans la dimension $d$. Par exemple, une fonction $f$ peut ressembler à celle de la Figure 7.1. Ici, la dimension de la fonction est de 3, pour simplifier. L'objectif est de concevoir un algorithme permettant de trouver automatiquement le meilleur paramètre $w^*$ pour minimiser la fonction de perte afin qu'elle puisse s'adapter au modèle d'IA.

**Méthodes du premier ordre.** Une méthode très classique pour résoudre ce problème est la méthode itérative – *Descente de Gradient*. C'est-à-dire qu'à la $k$-ième itération, le paramètre $w^k$ est mis à jour comme suit,

$$w^{k+1} = w^k - \eta^k \nabla f(w^k),$$

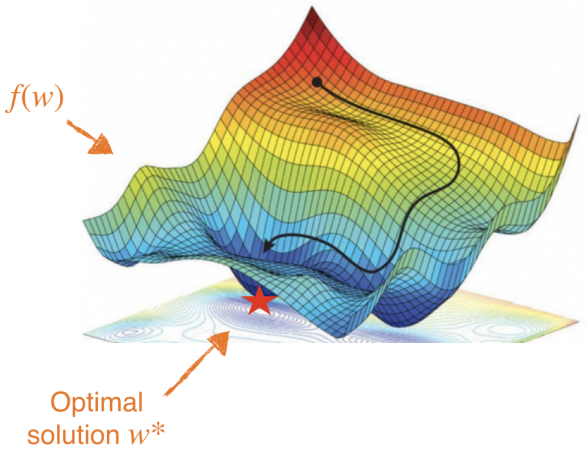


**Figure 7.1** – Paradigme d'optimisation.

où $\eta^k$ est la taille du pas, également connu sous le nom de taux d'apprentissage. La descente de gradient est également appelée méthode du premier ordre, car elle implique la dérivée première de la fonction.

Il s'agit d'une méthode d'optimisation très simple. Cependant, le problème des méthodes du premier ordre est qu'elles peuvent nécessiter un réglage important des paramètres et/ou une connaissance des paramètres du problème. Par exemple, la taille du taux d'apprentissage dépend de l'échelle de la fonction. En effet, étant donné une fonction $f$, le minimum de $f$ est identique au même problème multiplié par un nombre positif $C$, comme dans la figure 7.2. En revanche, les mises à jour de leur descente de gradient ne sont pas les mêmes. La deuxième mise à jour est proportionnelle à $C$. Donc, la descente de gradient est difficile à régler, car elle dépend fortement de l'échelle de la fonction. Par conséquent, le praticien doit injecter des connaissances de domaine à un niveau supérieur sur la manière dont les composants de modélisation interagissent pour qu'une méthode du premier ordre fonctionne bien. Cette dépendance à l'égard des méthodes du premier ordre limite en fin de compte le choix et le développement des modèles alternatifs.

**Méthodes du second ordre.** Examinons maintenant la *méthode de Newton*, qui est une autre méthode itérative classique pour minimiser $f$. En d'autres termes, à la $k$-ième itération, on fait

$$w^{k+1} = w^k - \eta \nabla^2 f(w^k)^{-1} \nabla f(w^k).$$

La méthode de Newton est également appelée méthode du second ordre, car elle fait intervenir la dérivée seconde de la fonction.

Grâce à l'accès aux informations du second ordre, la mise à jour de la méthode de Newton est capable de capturer la courbure locale de la fonction $f$, ce qui permet d'améliorer la direction de

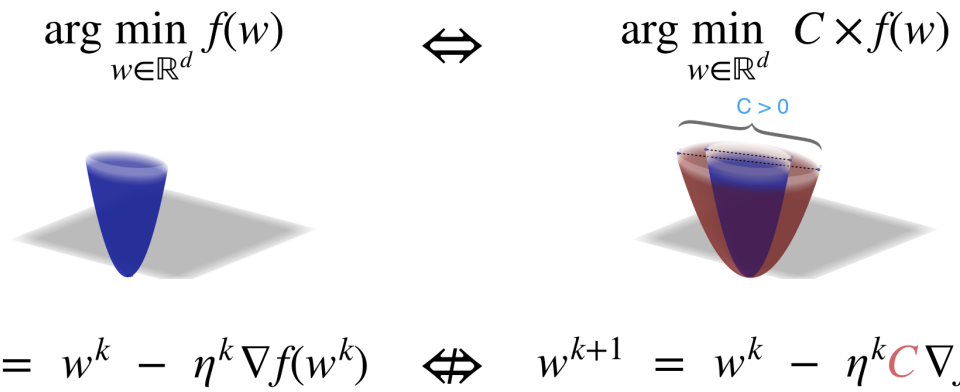


**Figure 7.2** – La descente de gradient dépend de l'échelle de la fonction.

la mise à jour par rapport à la méthode de descente de gradient. Cela conduit à une convergence plus rapide par rapport à la méthode de descente de gradient.

Plus important encore, la méthode de Newton est invariante à l'échelle. En effet, lorsque le problème est multiplié par un nombre positif $C$, la mise à jour de la méthode de Newton reste la même. C'est-à-dire,

$$w^{k+1} = w^k - \eta \nabla^2 f(w^k)^{-1} \nabla f(w^k) \quad \Longleftrightarrow \quad w^{k+1} = w^k - \eta \nabla^2 (C \cdot f(w^k))^{-1} \nabla (C \cdot f(w^k)).$$

Par conséquent, il est beaucoup plus facile d'ajuster la taille du taux d'apprentissage de la méthode de Newton que celle de la descente de gradient. Cependant, le calcul de l'opérateur inverse $\nabla^2 f(w^k)^{-1}$ est coûteux. Le coût par itération est de $d^3$, ce qui est prohibitif lorsque $d$ est grand. Une question naturelle se pose alors :

*Peut-on atteindre le meilleur des deux mondes ?*

C'est-à-dire avoir un algorithme qui ne souffre pas du réglage des paramètres, comme la taille du taux d'apprentissage, et qui maintient toujours un coût de calcul pas cher, même que les méthodes du premier ordre. Dans la première partie de la thèse, cette question sera abordée de manière positive.

### 7.1.2 Nouvelles méthodes du second ordre stochastiques

La croissance significative de la quantité des données dans les applications récentes d'apprentissage automatique (comme la publicité sur le web et la bio-informatique) empêche l'utilisation des méthodes de gradient déterministes ou des méthodes de Newton déterministes. Pour résoudre les problèmes d'apprentissage automatique à grande échelle, les méthodes stochastiques du premier ordre telles que *Descente de Gradient Stochastique* (Robbins and Monro, 1951, SGD), ADAGRAD (Duchi et al., 2011) et ADAM (Kingma and Ba, 2015) sont les méthodes de choix

dans la pratique en raison de leur faible coût par itération. Comme nous l'avons mentionné précédemment, le réglage de la taille des taux d'apprentissage peut prendre beaucoup de temps. Il y a maintenant un effort concerté pour développer des méthodes stochastiques efficaces du second ordre (Gupta et al., 2018) pour résoudre les problèmes d'apprentissage automatique à grande échelle. La motivation est qu'ils nécessitent moins de réglage des paramètres et convergent pour une plus grande variété de modèles et d'ensembles de données.

Dans cette première partie de la thèse, nous avons présenté une approche de principe afin de designer des méthodes stochastiques du second ordre pour résoudre à la fois des équations non linéaires et des problèmes d'optimisation de la somme finie d'une manière efficace. Notre approche comporte deux étapes. Premièrement, nous pouvons réécrire les équations non linéaires ou le problème de la somme finie comme des équations non linéaires souhaitées, en utilisant des astuces de séparation de variables ou de fonctions. Ensuite, nous appliquons de nouvelles méthodes du second ordre stochastiques pour résoudre ce système d'équations non linéaires. Pour les nouvelles méthodes du second ordre stochastiques, nous présentons la méthode de *Sketched Newton-Raphson* (SNR) au Chapitre 2 et la méthode de Sketched Newton-Raphson avec une métrique variable (SNRVM), qui est une extension de la méthode SNR dans la Section 3.4, Chapitre 3. SNR et SNRVM sont des variantes de la méthode de Newton-Raphson (NR) et peuvent avoir le même coût que SGD par itération lors de la résolution de problèmes de la somme finie. Cela permet de résoudre le souci de la méthode NR originale, dont le coût par itération est prohibitif lorsque la dimension des équations non linéaires est considérable. L'idée d'avoir un coût de calcul pas cher par itération est l'utilisation d'un outil stochastique : la technique de *sketch-and-project* (Gower and Richtárik, 2015b), qui nous permet de réduire la dimension du système de Newton et donc de rendre le coût de calcul par itération raisonable. À travers le SNR général et le SNRVM, nous présentons de nombreux nouveaux algorithmes spécifiques du second ordre qui peuvent résoudre efficacement les problèmes d'apprentissage automatique à grande échelle avec une structure de la somme finie sans besoin d'une connaissance du problème ni trop de réglage des paramètres. Vous trouverez plus de détails sur nos contributions dans la section suivante.

### 7.1.3 Plan et contributions de la Partie I

L'objectif général de la recherche qui mène la première partie de la thèse peut être formulé comme suivant

*concevoir un algorithme d'optimisation pour résoudre les problèmes d'apprentissage automatique à grande échelle, tel qu'il est incrémental et efficace, qu'il s'adapte bien à la dimension du problème et qu'il nécessite moins de réglage des paramètres.*

Pour atteindre cet objectif, notre première tentative est de proposer une nouvelle méthode du second ordre stochastique – Sketched Newton-Raphson (SNR) dans le Chapitre 2 qui combine la méthode de Newton-Raphson avec la technique de sketch-and-project (Gower and Richtárik, 2015b). Dans l'ensemble, notre principale contribution est une analyse approfondie du SNR sous différentes formes (par exemple, TCS dans la Section 2.8, méthode de Kaczmarz non linéaire (Wang et al., 2022) et méthode de Newton stochastique (Rodomanov and Kropotov, 2016; Kovalev et al., 2019, SNM)), avec la théorie de la convergence globale associée. À haut niveau, nous montrons comment SNR ouvre la porte à la construction et à l'analyse de nombreuses nouvelles méthodes du second ordre stochastiques (par exemple, TCS et méthode de Kaczmarz non linéaire (Wang et al., 2022)), ou à la récupération de méthodes du second ordre stochastiques existantes avec un bénéfice de la nouvelle théorie de convergence (par exemple, SNM (Rodomanov and Kropotov, 2016; Kovalev et al., 2019) et la méthode de Newton-Raphson originale). En ce qui concerne les théories de convergence de SNR, nous reformulons la méthode comme une variante de la méthode SGD. Cette reformulation est intéressante. Il s'avère que la reformulation est toujours une fonction smooth et interpolée. La condition d'interpolation signifie que la fonction a zero bruit pour le gradient stochastique à l'optimum. Ces propriétés sont fréquemment utilisées dans les preuves de convergence de SGD (Ma et al., 2018; Vaswani et al., 2019a). Grâce à cette reformulation, nous établissons la théorie de la convergence globale et les taux de convergence sous des hypothèses de type convexe en utilisant des techniques de preuve de SGD. Grâce à cette reformulation, notre théorie fournit également une nouvelle théorie de convergence globale pour la méthode de Newton-Raphson originale sous des hypothèses strictement plus faibles par rapport à la théorie de convergence monotone classique (Ortega and Rheinboldt, 2000; Deuflhard, 2011). Grâce au cadre général du SNR, nous préconisons le "Tossing-Coin-Sketch" – en bref, TCS – qui résout efficacement les problèmes d'apprentissage automatique à grande échelle. Quant à notre objectif de recherche, TCS est incrémental. Il est capable de ne prendre qu'un seul point de données par itération. Lors de l'échantillonnage d'un seul point de données par itération, TCS s'adapte bien à l'échelle du problème. Dans ce cas, le coût de calcul par itération est identique à celui de SGD. La méthode TCS nécessite également moins de réglage des paramètres de la taille du taux d'apprentissage par rapport à la méthode de premier ordre (par exemple SGD et ADAM (Kingma and Ba, 2015)), ce qui est normal puisqu'il s'agit d'une méthode de second ordre. Nous montrons à l'aide d'expériences numériques que TCS est compétitive par rapport aux méthodes classiques de gradient à variance réduite (par exemple SAG (Schmidt et al., 2017) et SVRG (Johnson and Zhang, 2013)). Cependant, TCS est efficace lorsqu'elle utilise un échantillonnage du mini-batch. Il converge lentement en expérience avec l'utilisation d'un seul point de données par itération. Pour que TCS fonctionne efficacement, il est nécessaire d'ajuster la taille de sketching. Par conséquent, l'objectif de la recherche est partiellement atteint, car nous devons encore ajuster la taille de sketching pour rendre l'algorithme efficace.

Motivés par notre objectif de recherche, plus précisément par la recherche d'un nouvel algorithme au-dessus du TCS qui nécessite moins de réglage des paramètres, y compris non seulement la taille du taux d'apprentissage mais aussi la taille de sketching, nous proposons *Stochastic Average Newton* (SAN) dans le Chapitre 3. En utilisant une approche similaire que SNR pour construire de nouvelles méthodes du second ordre stochastiques, nous développons SAN, qui est incrémental, dans le sens qu'il ne nécessite qu'un seul point de données par itération. Il est également peu coûteuse à mettre en œuvre lors de la résolution de modèles linéaires généralisés régularisés à grande échelle, avec le même coût par itération que SGD. Nous montrons par les expériences numériques que SAN est paramètre-free et compétitive par rapport aux méthodes de gradient à variance réduite (par exemple SAG (Schmidt et al., 2017) et SVRG (Johnson and Zhang, 2013)). Pour fournir une théorie de la convergence de nos méthodes, nous étendons SNR à SNRVM qui permet une métrique variable et qui inclut SAN comme un cas particulier.

Au total, la Partie I transmet un message conceptuel selon lequel il est possible de construire de nombreuses nouvelles méthodes du second ordre stochastiques capables de résoudre efficacement des problèmes d'apprentissage automatique à grande échelle sans connaissance du problème, ni réglage des paramètres.

## 7.2 Analysis de Temps Fini des Méthodes de Policy-Gradient en Apprentissage par Renforcement

Dans la deuxième partie de la thèse, nous nous concentrons ensuite sur les algorithmes d'optimisation appliqués à un domaine spécifique : l'apprentissage par renforcement (RL). Cette partie est indépendante de la première partie de la thèse.

### 7.2.1 Apprentissage par renforcement

Nous avons obtenu quelques résultats les plus impressionnants de l'IA dans le domaine de RL, comme les jeux vidéos et le jeu de Go (Mnih et al., 2015; Silver et al., 2017; OpenAI et al., 2019; Vinyals et al., 2019), le véhicule autonome (Shalev-Shwartz et al., 2016; Kiran et al., 2022), la robotique (Kober et al., 2013; Levine et al., 2016; Gu* et al., 2017; Levine et al., 2018) et bien plus encore. Alors, qu'est-ce que RL ? La réponse courte est que RL consiste à apprendre dans un environnement inconnu par des essais et des échecs pour prendre des décisions séquentielles.

**Processus de décision markovien (MDP).** Dans le paradigme traditionnel du RL, comme le montre la Figure 7.3, L'interaction entre un agent et un environnement peut être modélisé comme un processus de décision markovien (Puterman, 1994, MDP). Au temps $t$, l'agent est à

l'état $s_t$ quelque part dans l'environnement. L'environnement peut être modélisé comme un espace d'états $\mathcal{S}$. Ensuite, l'agent prend une action $a_t$ parmi toutes les actions possibles dans l'espace des actions $\mathcal{A}$. En fonction de l'état actuel $s_t$ et de l'action prise $a_t$, l'environnement conduira l'agent à l'état suivant $s_{t+1}$ avec une probabilité de transition $\mathcal{P}$, également connue sous le nom de dynamique de l'environnement. Grâce à cette interaction, l'agent gagnera une récompense $r(s_t, a_t)$[1]. En particulier, l'action $a_t$ est choisie selon la politique $\pi \in \Delta(\mathcal{A})^{\mathcal{S}}$, qui est une fonction de l'espace d'état $\mathcal{S}$ vers le simplexe de probabilité de l'espace d'action $\Delta(\mathcal{A})$. Nous notons $\pi_{s_t,a_t} \in \mathbb{R}$ la densité du choix de l'action $a_t$ sur l'espace des actions à l'état $s_t$ et $\pi_{s_t} \in \Delta(\mathcal{A})$ est la distribution des actions à l'état $s_t$. Ainsi, une politique induit une distribution sur les trajectoires $\{s_t, a_t, r(s_t, a_t)\}_{t\geq 0}$.

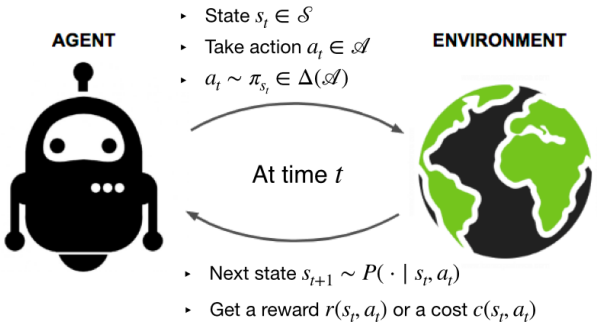


**Figure 7.3** – Un agent interagit avec l'environnement, en essayant de prendre des actions intelligentes pour maximiser les récompenses cumulées.

**L'optimisation de la politique.** L'objectif de l'agent est de résoudre le MDP. C'est-à-dire, il s'agit de trouver la politique optimale de sorte que la totale des récompenses cumulées sur la trajectoire en espérence $V_\rho(\pi)$, définie comme

$$V_\rho(\pi) \overset{\text{def}}{=} \mathbb{E}_{s_0\sim\rho,\ a_t\sim\pi_{s_t},\ s_{t+1}\sim\mathcal{P}(\cdot|s_t,a_t)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right],$$

soit maximale. Le problème est également appelé optimisation de la politique. Ici, l'espérance est liée à la distribution de l'état initial $\rho \in \Delta(\mathcal{S})$ pour $s_0$, suivie de la politique $\pi$ et de la dynamique $\mathcal{P}$. Le $\gamma \in [0, 1)$ est le facteur d'actualisation qui définit l'importance des récompenses futures. Un $\gamma$ proche de $0$ signifie que seuls les récompenses à court terme sont prises en compte, par conséquence, les récompenses anciennes auront un faible impact ; un $\gamma$ proche de $1$ signifie que nous nous concentrons sur les récompenses à long terme.

En pratique, l'espace de politique est très large. Pour réduire les dimensions et rendre le calcul faisable, la politique $\pi$ est souvent paramétrée comme $\pi(\theta)$ avec $\theta \in \Theta \subset \mathbb{R}^d$ appartenant

[1]Au Chapitre 5, nous utilisons le coût au lieu de la récompense pour mieux aligner sur la convention de minimisation dans la littérature de l'optimisation.

à une certaine famille $\Theta$. La fonction $V_\rho(\pi(\theta))$ dépend donc du paramètre $\theta$ et nous utilisons l'abréviation $V_\rho(\theta) \stackrel{\text{def}}{=} V_\rho(\pi(\theta))$. Notre objectif est maintenant de trouver le paramètre optimal $\theta$ pour maximiser $V_\rho(\theta)$, ce qui peut être formulé comme le problème suivant

$$\arg\max_{\theta\in\Theta} V_\rho(\theta) \stackrel{\text{def}}{=} \mathbb{E}_{s_0\sim\rho,\ a_t\sim\pi_{s_t}(\theta),\ s_{t+1}\sim\mathcal{P}(\cdot|s_t,a_t)}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right].$$

Tout au long de la thèse, nous considérons $\Theta = \mathbb{R}^d$ en général sans spécification.

### 7.2.2 Méthodes de policy-gradient

Naturellement, nous pouvons considérer la maximisation de $V_\rho(\theta)$ comme un problème d'optimisation. Nous pouvons donc le résoudre à l'aide de méthodes de type ascente de gradient, connues sous le nom de *Policy-Gradient* (PG) en RL. En d'autres termes, à la $k$-ième itération, on fait

$$\theta^{k+1} \;=\; \theta^k \;+\; \eta^k \nabla_\theta V_\rho(\theta).$$

PG est populaire en RL en raison de sa simplicité. Par exemple, il est plus facile à mettre en œuvre et à utiliser dans la pratique que les méthodes basées sur la valeur (value-based) ou sur le modèle, qui sont des méthodes spécifiques en RL. PG peut résoudre un grand ensemble de problèmes, y compris les environnements non markoviens et partiellement observables.

La popularité de PG est également due à sa polyvalence. Tout d'abord, PG dispose de plusieurs formes, telles que REINFORCE (Williams, 1992), PGT (Sutton et al., 2000), GPOMDP (Baxter and Bartlett, 2001) et actor-critic (Konda and Tsitsiklis, 2000). Il peut être efficacement associé à des techniques d'optimisation pour obtenir des algorithmes plus sophistiqués. Par exemple, le gradient naturel de la politique (Kakade, 2001, NPG) est une application directe de la méthode du gradient naturel (Amari, 1998) de l'optimisation au RL, et la descente en miroir de la politique (Lan, 2022; Xiao, 2022, PMD) est inspirée de la descente en miroir (Nemirovski and Yudin, 1983; Beck and Teboulle, 2003) en optimisation. Combinée aux techniques à variance réduite, telles que SVRG (Johnson and Zhang, 2013), SARAH (Nguyen et al., 2017), SPIDER (Fang et al., 2018), SpiderBoost (Wang et al., 2019), STORM (Cutkosky and Orabona, 2019), SNVRG (Zhou et al., 2020), PAGE (Li et al., 2021b), et plus encore (Tran-Dinh et al., 2021), de nombreuses méthodes de PG à variance réduite en RL (Papini et al., 2018; Shen et al., 2019; Xu et al., 2020b; Yuan et al., 2020; Huang et al., 2020; Pham et al., 2020; Yang et al., 2022; Huang et al., 2022) ont été développées récemment. En fait, les algorithmes de l'état de l'art actuel en optimisation de la politique, comme TRPO (Schulman et al., 2015) et PPO (Schulman et al., 2017), sont développés en utilisant des structures spécifiques de RL et des techniques d'optimisation (par exemple, la méthode de la région de confiance et la méthode proximale). Dans l'ensemble, les variantes des méthodes de PG avec des techniques

d'optimisation se sont révélées avoir des succès empiriques impressionnants (Schulman et al., 2015; Lillicrap et al., 2016; Mnih et al., 2016; Schulman et al., 2017; Haarnoja et al., 2018), en particulier dans le RL profond.

Malgré le succès des méthodes de PG dans la pratique, une solide compréhension théorique même de PG original a longtemps été difficile à obtenir jusqu'à récemment. Cependant, la littérature reste fragmentaire. Une partie de la littérature se concentre sur l'analyse de PG déterministe, y compris le travail de pionnier de Agarwal et al. (2021) et d'autres travaux (Zhang et al., 2020a; Mei et al., 2020) ; certains se concentrent sur le PG stochastique (Papini, 2020; Liu et al., 2020; Zhang et al., 2020b; Xiong et al., 2021). En termes de résultats, ils s'appuient sur différents critères de convergence, tels que la convergence du point stationnaire de premier ordre (Papini, 2020; Zhang et al., 2020b), la convergence vers l'optimum global (Agarwal et al., 2021; Zhang et al., 2020a; Mei et al., 2020) et le regret moyen par rapport à l'optimum global (Zhang et al., 2021b; Liu et al., 2020). Différents résultats sont appliqués dans différents environnements de RL, tels que la politique tabulaire de softmax avec ou sans différentes régularisations (Agarwal et al., 2021; Zhang et al., 2020a; Zhang et al., 2021b; Mei et al., 2020), ou avec des hypothèses différentes, telles que la politique Lipschitz et smooth (Liu et al., 2020; Zhang et al., 2020b; Xiong et al., 2021) et l'hypothèse de la bijection entre l'espace primal et l'espace dual (Zhang et al., 2020a). En particulier, une grande partie de la littérature nécessite un grand mini-batch de trajectoires échantillonnées, telles que $\mathcal{O}(\epsilon^{-1})$ ou $\mathcal{O}(\epsilon^{-2})$ trajectoires par itération pour les mises à jour stochastiques des paramètres (Papini, 2020; Liu et al., 2020; Zhang et al., 2020b; Xiong et al., 2021). Ici, $\epsilon$ est la précision de la performance. Cela est étrange, car dans la littérature sur la théorie de la convergence SGD en optimisation, une seule donnée par itération n'est généralement pas un problème.

Le deuxième défi de PG est que, contrairement aux méthodes basées sur la valeur ou sur le modèle, les méthodes de PG existantes ne sont pas efficaces pour l'échantillonnage en théorie. Récemment, Xiao (2022) a prouvé que la méthode NPG était efficace pour le cas tabulaire avec une convergence linéaire, ce qui est aussi le taux de convergence des méthodes basées sur la valeur, tel que l'algorithme d'itération sur la politique (Puterman, 1994; Bertsekas, 2012). Comme mentionné, NPG (Kakade, 2001) s'inspire de la méthode du gradient naturel (Amari, 1998), utilise un préconditionneur pour améliorer la direction de PG, similaire aux méthodes de quasi-Newton en optimisation classique (Martens, 2020). Pouvons-nous donc étendre la convergence linéaire de NPG du régime tabulaire au régime d'approximation des fonctions, qui est un cas plus réaliste dans la pratique ? En outre, NPG est important. C'est la pierre angulaire de TRPO (Schulman et al., 2015) et de PPO (Schulman et al., 2017). Il est donc très intéressant de bien comprendre NPG et de repousser ses limites. Nous abordons ces deux défis de PG dans la deuxième partie de la thèse, séparément.

### 7.2.3 Plan et contributions de la Partie II

A la lumière de ce qui précède, cette deuxième partie de la thèse est consacrée à une meilleure compréhension théorique des méthodes de PG. Nous posons la question suivante : pourquoi les méthodes de PG sont-elles efficaces et comment justifier le choix de leurs hyper paramètres ? En utilisant la structure de RL du problème et des techniques modernes de preuve d'optimisation (Khaled and Richtárik, 2023; Lan, 2022; Xiao, 2022), nous dérivons de nouvelles analyses en temps fini de PG original et de NPG dans les Chapitres 4 et 5, respectivement.

Tout d'abord, dans le Chapitre 4, nous adaptons des outils récents développés pour l'analyse de SGD dans l'optimisation non convexe de Khaled and Richtárik (2023) afin d'obtenir des garanties de convergence et de complexité d'échantillon pour le PG original, y compris REINFORCE (Williams, 1992), PGT (Sutton et al., 2000) et GPOMDP (Baxter and Bartlett, 2001). Tout au long de la thèse, nous appellerons les mises à jour de REINFORCE, PGT et GPOMDP PG vanille. Notre principale contribution est de fournir une analyse générale de PG vanille avec des hypothèses plus faibles par rapport à la littérature. Cette analyse générale permet non seulement d'unifier la plupart des résultats fragmentés de la littérature sous une même forme, mais aussi de retrouver les meilleurs résultats pour chaque contexte, avec un intervalle de choix d'hyper paramètres plus large, ce qui peut avoir un grand intérêt dans la pratique, et parfois même d'améliorer les résultats existants avec une hypothèse supplémentaire de domination du gradient. Plus précisément, nous fournissons un seul théorème de convergence qui récupère la complexité d'échantillonnage $\widetilde{\mathcal{O}}(\epsilon^{-4})$ du PG vanille vers un point stationnaire. En d'autres termes, considérons un échantillon comme un triple $(s_t, a_t, r(s_t, a_t))$ qui est une interaction en une étape avec l'environnement au temps $t$ parmi une seule trajectoire échantillonnée $\{s_{t'}, a_{t'}, r(s_{t'}, a_{t'})\}_{t' \geq 0}$ par itération. Avec des échantillons $\widetilde{\mathcal{O}}(\epsilon^{-4})$, le PG vanille est garantie de converger vers un $\epsilon$-point stationnaire . Nos résultats offrent également une plus grande flexibilité dans le choix des hyper paramètres tels que la taille du taux d'apprentissage et la taille du mini-batch $m$ des trajectoires, y compris l'utilisation d'une seule trajectoire (i.e. $m = 1$) par itération. Lorsqu'une hypothèse supplémentaire *relaxed weak gradient domination* est disponible, nous établissons une nouvelle théorie de convergence de l'optimum global de PG avec une complexité d'échantillonnage $\widetilde{\mathcal{O}}(\epsilon^{-3})$. Nous instancions ensuite nos théorèmes dans différents contextes, où nous retrouvons les résultats existants et obtenons une complexité d'échantillon améliorée, par exemple $\widetilde{\mathcal{O}}(\epsilon^{-3})$ complexité d'échantillon pour la convergence vers l'optimum global pour les politiques paramétrées non dégénérées de Fisher. L'ingrédient clé de l'analyse consiste à considérer l'hypothèse ABC (Khaled and Richtárik, 2023), qui limite le gradient empirique en termes d'écart de sous-optimalité (A), de gradient en espérance mais tronqué (B) et d'une constante additive (C). Cette hypothèse ABC peut sembler un peu obscure à première vue, mais il s'agit en fait d'un moyen astucieux d'unifier un grand nombre des hypothèses actuelles utilisées dans la littérature RL. Notamment, les politiques Lipschitz et smooth espérance, les politiques tabulaires de softmax avec ou sans régularisation, et les

politiques non dégénérées de Fisher sont considérées comme des cas particuliers de l'hypothèse ABC. Grâce à notre analyse générale, nous parvenons à une meilleure compréhension théorique du PG vanille et nous avons la liberté de choisir les hyper paramètres de PG dans la pratique en fonction des ressources informatiques disponibles.

Comme indiqué dans la section précédente, le PG vanille n'est pas efficace en termes d'échantillonnage. Au Chapitre 5, nous développons la convergence linéaire d'un autre algorithme RL populaire connu sous le nom de NPG et de sa variante, Q-NPG, pour la classe des politiques log-linéaires. Les théorèmes qui en résultent étendent les travaux de Xiao (2022) des politiques softmax tabulaires au régime d'approximation des fonctions. Nous montrons qu'en utilisant une taille de taux d'apprentisage géométriquement croissante, ces algorithmes peuvent atteindre un taux de convergence linéaire, similaire au cas tabulaire, jusqu'à l'erreur d'approximation de la fonction. Le cœur de l'analyse est que, en s'appuyant sur le framework d'approximation de fonction compatible développé par Agarwal et al. (2021), NPG peut être interprétée comme une méthode de gradient d'ascente en miroir. C'est le point de vue adopté pour l'analyse, où une mise à jour inexacte de l'ascenste en miroir est considérée. Le chapitre fournit en outre des résultats de complexité d'échantillon $\widetilde{\mathcal{O}}(\epsilon^{-2})$ sous certaines hypothèses techniques supplémentaires, qui améliorent les meilleurs résultats connus dans la littérature.

Tout au long de la Partie II, nous obtenons une meilleure compréhension et une meilleure efficacité des échantillons dans les méthodes de PG en RL.

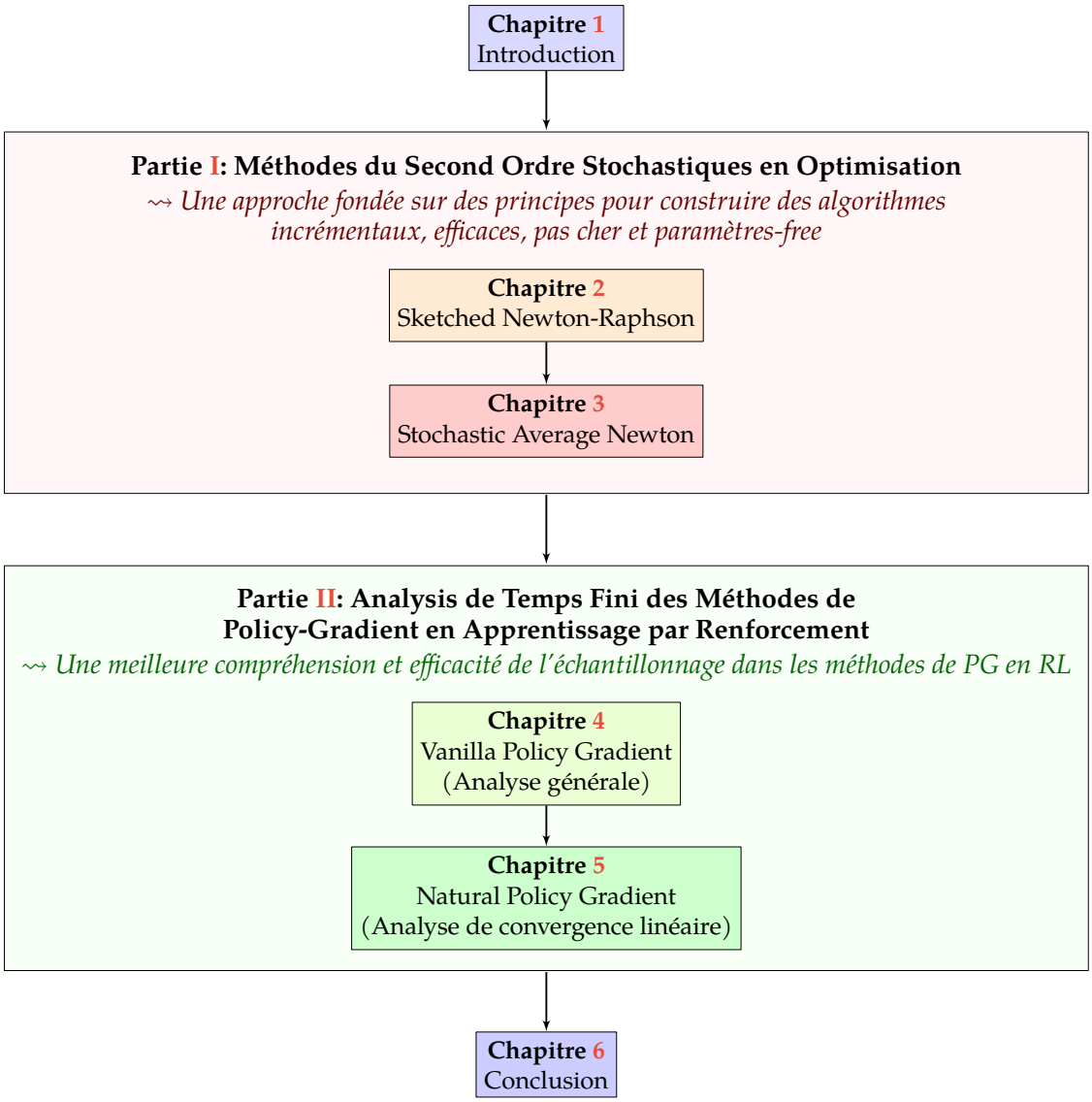


**Figure 7.4** – Cette thèse est divisée en deux parties. Nous commençons par l'optimisation dans la Partie I où nous concevons de nouvelles méthodes du second ordre stochastiques et efficaces avec des garanties de convergence. En nous appuyant sur les techniques de preuve d'optimisation, nous passons ensuite à l'apprentissage par renforcement (RL) dans la Partie II qui se concentre sur les fondements théoriques des méthodes de policy-gradient (PG), y compris le PG vanille et le PG naturel. Ces deux sujets sont présentés comme étant orthogonaux, mais le fil conducteur est l'optimisation.

## Liste de publications

### Publications dans des conférences internationales avec proceedings

- Rui Yuan, Alessandro Lazaric, Robert M. Gower. **Sketched Newton-Raphson**. In *Society for Industrial and Applied Mathematics* (*SIAM*) *Journal on Optimization* (SIOPT), 2022 (presenté dans le Chapitre 2)
- Jiabin Chen *, Rui Yuan *, Guillaume Garrigos, Robert M. Gower. **SAN: Stochastic Average Newton Algorithm for Minimizing Finite Sums**. In *International Conference on Artificial Intelligence and Statistics* (AISTATS), 2022 (presenté dans le Chapitre 3)
- Rui Yuan, Robert M. Gower, Alessandro Lazaric. **A general sample complexity analysis of vanilla policy gradient**. In *International Conference on Artificial Intelligence and Statistics* (AISTATS), 2022 (presenté dans le Chapitre 4)
- Rui Yuan, Simon S. Du, Robert M. Gower, Alessandro Lazaric, Lin Xiao. **Linear Convergence of Natural Policy Gradient Methods with Log-Linear Policies**. In *International Conference on Learning Representations* (ICLR), 2023 (presenté dans le Chapitre 5)

### Publication discutée dans cette thèse

- Carlo Alfano, Rui Yuan, Patrick Rebeschini. **A Novel Framework for Policy Mirror Descent with General Parametrization and Linear Convergence**. Preprint, 2023 (discuté dans le Chapitre 5).

*indique une contribution égale.

# Appendix A

# Complements on Chapter 2

## Contents



Here we provide some additional noteworthy observations made in Chapter 2. In particular, we show that SNR can also be seen as a type of stochastic Gauss-Newton method in Appendix A.1.1 or as a type of stochastic fixed point method in Appendix A.1.2, respectively. In Appendix A.6, we provide the sufficient conditions for the reformulation assumption (2.10), where our examples SNM and TCS satisfy those conditions. In Appendix A.8, we carefully derive the closed form updates of TCS presented in Section 2.8. In Appendix A.9, we provide more detailed and efficient pseudo-codes for TCS. In Appendix B.3.1, we give further details on the numerical experiments. And we add additional experiments of TCS methods combined with stochastic line-search in Appendix A.11.

## A.1 Other viewpoints of SNR

Beside the connection between SNR and SGD, in the next section we reformulate SNR as a stochastic Gauss-Newton (GN) method and a stochastic fixed point method in the subsequent Appendix A.1.2.

### A.1.1 Stochastic Gauss-Newton method

The GN method is a method for solving nonlinear least-squares problems such as

$$\min_{x\in\mathbb{R}^p} \|F(x)\|^2_{\mathbf{G}}\,, \tag{A.1}$$

where $\mathbf{G}$ is a symmetric positive-definite matrix. Like the Newton-Raphson method, at each step of the GN method, the function $F(x)$ is replaced by its linearization in (A.1) and then solved to give the next iterate. That is

$$x^{k+1} \in \operatorname{argmin}_{x\in\mathbb{R}^p} \left\|DF(x^k)^\top(x-x^k)+\gamma F(x^k)\right\|^2_{\mathbf{G}}\,, \tag{A.2}$$

where $x^{k+1}$ is the least-norm solution to the above.

Now consider the GN method where the matrix that defines the norm in (A.2) changes at each iteration as is given by $\mathbf{G}\equiv\mathbf{G}^k \overset{\text{def}}{=} \mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x^k)\right]$ and let $d \overset{\text{def}}{=} x-x^k$. Since $\mathbf{G}^k$ is an expected matrix, we can write

$$\left\|DF(x^k)^\top d+\gamma F(x^k)\right\|^2_{\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x^k)\right]} \;=\; \mathbb{E}\left[\left\|DF(x^k)^\top d+\gamma F(x^k)\right\|^2_{\mathbf{H}_{\mathbf{S}}(x^k)}\right].$$

This suggests a stochastic variant of the GN where we use the unbiased estimate $\mathbf{H}_{\mathbf{S}}(x^k)$ instead of $\mathbf{G}^k$. This stochastic GN method is in fact equivalent to SNR, as we show next.

**Lemma A.1.** *Let $x^0\in\mathbb{R}^p$ and consider the following* Stochastic Gauss-Newton *method*

$$\begin{aligned} d^k &\in \operatorname*{argmin}_{d\in\mathbb{R}^p} \left\|DF(x^k)^\top d+\gamma F(x^k)\right\|^2_{\mathbf{H}_{\mathbf{S}_k}(x^k)} \\ x^{k+1} &= x^k + d^k \end{aligned} \tag{A.3}$$

*where $\mathbf{S}_k$ is sampled from $\mathcal{D}_{x^k}$ at kth iteration and $d^k$ is the least-norm solution. If Assumption 3.5 holds, then the iterates (A.3) are equal to the iterates of SNR (line 4 in Algorithm 1).*

*Proof.* Differentiating (A.3) in $d$, we find that $d^k$ is a solution to

$$DF(x^k)\mathbf{H}_{\mathbf{S}_k}(x^k)DF(x^k)^\top d^k = -\gamma DF(x^k)\mathbf{H}_{\mathbf{S}_k}(x^k)F(x^k).$$

Let $\mathbf{A} \overset{\text{def}}{=} DF(x^k)\mathbf{H}_{\mathbf{S}_k}(x^k)DF(x^k)^\top$. Taking the least-norm solution to the above gives

$$\begin{aligned} d^k &= -\gamma \mathbf{A}^\dagger DF(x^k)\mathbf{H}_{\mathbf{S}_k}(x^k)F(x^k) = -\gamma \mathbf{A}^\dagger \mathbf{A} v \\ &= -\gamma \mathbf{A}^\dagger \mathbf{A}\mathbf{A} v = -\gamma \mathbf{A} v \\ &= -\gamma DF(x^k)\mathbf{H}_{\mathbf{S}_k}(x^k)F(x^k), \end{aligned}$$

where on the first line, we used that Assumption 3.5 shows there exists $v \in \mathbb{R}^p$ such that $F(x^k) = DF(x^k)^\top v$. On the second line, we used that $\mathbf{A} = \mathbf{A}\mathbf{A}$ which is shown in the proof of Corollary 2.8 . Then we used $\mathbf{A}^\dagger \mathbf{A}\mathbf{A} = \mathbf{A}$ which is a property of the pseudoinverse operator that holds for all symmetric matrices. Consequently $x^{k+1} = x^k + d^k$ which is exactly the update given in line 4 in Algorithm 1. □

Thus our sketched Newton-Raphson method can also be seen as a stochastic Gauss-Newton method. Furthermore, if $\mathbf{S} = \mathbf{I}$, then (A.3) is no longer stochastic and is given by

$$\begin{aligned} d^k &\in \operatorname*{argmin}_{d \in \mathbb{R}^p} \left\| DF(x^k)^\top d + \gamma F(x^k) \right\|^2_{(DF(x^k)^\top DF(x^k))^\dagger} \\ x^{k+1} &= x^k + d^k. \end{aligned} \tag{A.4}$$

Thus as a consequence of Lemma A.1, we have that this variant (A.4) of GN is in fact the Newton-Raphson method.

### A.1.2 Stochastic fixed point method

In this section, we reformulate SNR as a stochastic fixed point method. Such interpretation is inspired from Richtárik and Takáč (2020)'s stochastic fixed point viewpoint. We extend their results from the linear case to the nonlinear case.

Assume that Assumption 3.5 holds and re-consider the sketch-and-project viewpoint (2.7) in Section 2.2. Note the zeros of the function $F$

$$\mathcal{L} \overset{\text{def}}{=} \{x \mid F(x) = 0\}$$

and the sketched Newton system based on $y$

$$\mathcal{L}_{\mathbf{S},y} \overset{\text{def}}{=} \left\{ x \in \mathbb{R}^p \mid \mathbf{S}^\top DF(y)^\top (x - y) = -\mathbf{S}^\top F(y) \right\}$$

with $y \in \mathbb{R}^p$ and $\mathbf{S} \sim \mathcal{D}_y$. For a closed convex set $\mathcal{Y} \subseteq \mathbb{R}^d$, let $\Pi_{\mathcal{Y}}$ denote the projection operator onto $\mathcal{Y}$. That is

$$\Pi_{\mathcal{Y}}(x) \overset{\text{def}}{=} \operatorname{argmin}_{y \in \mathbb{R}^p} \{\|y - x\| : y \in \mathcal{Y}\}. \tag{A.5}$$

Then, from (2.7) by plugging $\mathcal{Y} = \mathcal{L}_{\mathbf{S},y}$ and $y = x$ into (A.5), we have

$$\Pi_{\mathcal{L}_{\mathbf{S},x}}(x) \;=\; x - DF(x)\mathbf{H}_{\mathbf{S}}(x)F(x). \tag{A.6}$$

Now we can introduce the fixed point equation as follows

$$\chi \overset{\text{def}}{=} \left\{x \mid x = \mathbb{E}_{\mathbf{S}\sim\mathcal{D}_x}\left[\Pi_{\mathcal{L}_{\mathbf{S},x}}(x)\right]\right\}. \tag{A.7}$$

Assumption 3.5 guarantees that finding fixed points of (A.7) is equivalent to the reformulated optimization problem (2.8) with $y = x$, as we show next.

**Lemma A.2.** *If Assumption 3.5 holds, then*

$$\chi \;=\; \operatorname{argmin}_{x\in\mathbb{R}^p} \frac{1}{2}\|F(x)\|^2_{\mathbb{E}_{\mathbf{S}\sim\mathcal{D}_x}[\mathbf{H}_{\mathbf{S}}(x)]}. \tag{A.8}$$

*Proof.* Let $\chi_{\mathbf{S}} \overset{\text{def}}{=} \left\{x \mid x = \Pi_{\mathcal{L}_{\mathbf{S},x}}(x)\right\}$ with $\mathbf{S} \sim \mathcal{D}_x$. First, we show that

$$\chi_{\mathbf{S}} = \operatorname{argmin}_{x\in\mathbb{R}^d} \frac{1}{2}\|F(x)\|^2_{\mathbf{H}_{\mathbf{S}}(x)}. \tag{A.9}$$

In fact,

$$\begin{aligned}
x \in \chi_{\mathbf{S}} \quad &\overset{(A.6)}{\iff} \quad DF(x)\mathbf{H}_{\mathbf{S}}(x)F(x) = 0 \\
&\overset{\text{Assumption 3.5}}{\iff} \quad \exists v \in \mathbb{R}^d \text{ s.t. } F(x) = DF(x)^\top v \text{ and } DF(x)\mathbf{H}_{\mathbf{S}}(x)DF(x)^\top v = 0 \\
&\iff \quad \mathbf{H}_{\mathbf{S}}(x)F(x) = 0 \qquad (\text{as } DF(x)\mathbf{H}_{\mathbf{S}}(x)DF(x)^\top \succeq 0) \\
&\iff \quad \frac{1}{2}\|F(x)\|^2_{\mathbf{H}_{\mathbf{S}}(x)} = 0.
\end{aligned}$$

So we induce (A.9). Finally (A.8) follows by taking expectations with respect to $\mathbf{S}$ in (A.9). □

To solve the fixed point equation (A.7), the natural choice of method is the stochastic fixed point method with relaxation. That is, we pick a relaxation parameter $\gamma > 0$, and consider the following equivalent fixed point problem

$$x \;=\; \mathbb{E}_{\mathbf{S}\sim\mathcal{D}_x}\left[\gamma\Pi_{\mathcal{L}_{\mathbf{S},x}}(x) + (1-\gamma)x\right].$$

Using relaxation is to improve the contraction properties of the map. Then at $k$th iteration,

$$x^{k+1} \;=\; \gamma \Pi_{\mathcal{L}_{\mathbf{S},x^k}}(x^k) + (1-\gamma)x^k, \tag{A.10}$$

where $\mathbf{S} \sim \mathcal{D}_{x^k}$. Consequently, it is straight forward to verify that (A.10) is exactly the update given in line 4 in Algorithm 1.

## A.2 Proof of Section 2.4

### A.2.1 Proof of Lemma 2.5

*Proof.* Turning to the definition of $f_{\mathbf{S},x}$ in (2.9), we have that

$$\begin{aligned} \|\nabla f_{\mathbf{S},x}(x)\|^2 &\overset{(2.11)}{=} \|DF(x)\mathbf{H}_{\mathbf{S}}(x)F(x)\|^2 = F(x)^\top \mathbf{H}_{\mathbf{S}}(x)^\top DF(x)^\top DF(x)\mathbf{H}_{\mathbf{S}}(x)F(x) \\ &= F(x)^\top \mathbf{H}_{\mathbf{S}}(x)F(x) = 2f_{\mathbf{S},x}(x), \end{aligned}$$

where we used the property $\mathbf{M}^\dagger\mathbf{M}\mathbf{M}^\dagger = \mathbf{M}^\dagger$ with $\mathbf{M} = \mathbf{S}^\top DF(x)^\top DF(x)\mathbf{S}$ to establish that $\mathbf{H}_{\mathbf{S}}(x)^\top DF(x)^\top DF(x)\mathbf{H}_{\mathbf{S}}(x) \overset{(2.4)}{=} \mathbf{H}_{\mathbf{S}}(x)$. □

### A.2.2 Proof of Theorem 2.7

*Proof.* Let $t \in \{0,\dots,k-1\}$ and $\delta_t \overset{\text{def}}{=} x^t - x^*$. We have that

$$\begin{aligned} \mathbb{E}_t\left[\|\delta_{t+1}\|^2\right] &\overset{(2.12)}{=} \mathbb{E}_t\left[\left\|x^t - \gamma\nabla f_{\mathbf{S}_t,t}(x^t) - x^*\right\|^2\right] \\ &= \|\delta_t\|^2 - 2\gamma\left\langle \delta_t, \nabla f_t(x^t)\right\rangle + \gamma^2\mathbb{E}_t\left[\left\|\nabla f_{\mathbf{S}_t,t}(x^t)\right\|^2\right] \\ &\overset{(2.14)}{\leq} \|\delta_t\|^2 - 2\gamma(f_t(x^t) - f_t(x^*)) + \gamma^2\mathbb{E}_t\left[\left\|\nabla f_{\mathbf{S}_t,t}(x^t)\right\|^2\right] \\ &\overset{(2.13)}{=} \|\delta_t\|^2 - 2\gamma\left(1-\gamma\right)\left(f_t(x^t) - f_t(x^*)\right) \\ &\overset{f_t(x^*)=0}{=} \|\delta_t\|^2 - 2\gamma\left(1-\gamma\right)f_t(x^t). \end{aligned} \tag{A.11}$$

Taking total expectation for all $t \in \{0,\dots,k-1\}$, we have that

$$\mathbb{E}\left[\|\delta_{t+1}\|^2\right] \leq \mathbb{E}\left[\|\delta_t\|^2\right] - 2\gamma\left(1-\gamma\right)\mathbb{E}\left[f_t(x^t)\right]. \tag{A.12}$$

Summing both sides of (A.12) from 0 to $k-1$ gives

$$\mathbb{E}\left[\left\|x^k - x^*\right\|^2\right] + 2\gamma\left(1-\gamma\right)\sum_{t=0}^{k-1}\mathbb{E}\left[f_t(x^t)\right] \leq \left\|x^0 - x^*\right\|^2.$$

Dividing through by $2\gamma(1-\gamma) > 0$ and by $k$, we have that

$$\mathbb{E}\left[\min_{t=0,\ldots,k-1} f_t(x^t)\right] \leq \min_{t=0,\ldots,k-1} \mathbb{E}\left[f_t(x^t)\right] \leq \frac{1}{k}\sum_{t=0}^{k-1}\mathbb{E}\left[f_t(x^t)\right] \leq \frac{1}{k}\frac{\|x^0 - x^*\|^2}{2\gamma(1-\gamma)},$$

where in the most left inequality we used Jensen's inequality.

Finally, if (2.16) holds, then we can repeat the steps leading up to (A.11) without the conditional expectation, so that

$$\|\delta_{t+1}\|^2 \overset{(2.12)+(2.16)+(2.13)}{\leq} \|\delta_t\|^2 - 2\gamma(1-\gamma) f_{\mathbf{S}_t,t}(x^t).$$

Since $f_{S_t,t}(x^t) \geq 0$, we have $\|\delta_{t+1}\|^2 \leq \|\delta_t\|^2$, i.e. (2.17) holds. □

### A.2.3 Proof of Corollary 2.8

To prove Corollary 2.8, we need the following lemma first.

**Lemma A.3** (Lemma 10 in Gower et al. (2019a)). *For any matrix* $\mathbf{W}$ *and symmetric positive semi-definite matrix* $\mathbf{G}$ *s.t.* $\mathbf{Ker}(\mathbf{G}) \subset \mathbf{Ker}(\mathbf{W})$, *we have* $\mathbf{Ker}(\mathbf{W}^\top) = \mathbf{Ker}(\mathbf{W}\mathbf{G}\mathbf{W}^\top)$.

Now we show the proof of Corollary 2.8.

*Proof.* First recall that

$$DF(x)\mathbf{H}_{\mathbf{S}}(x)DF(x)^\top DF(x)\mathbf{H}_{\mathbf{S}}(x)DF(x)^\top = DF(x)\mathbf{H}_{\mathbf{S}}(x)DF(x)^\top \quad \text{for all } x \in \mathbb{R}^p,$$

which is shown in the proof of Lemma 2.5. Thus $DF(x)\mathbf{H}_{\mathbf{S}}(x)DF(x)^\top$ is a projection. By Jensen's inequality, the eigenvalues of an expected projection are between 0 and 1. Thus by the definition of $\rho(x)$, we have $0 \leq \rho(x) \leq 1$. Next, by (2.22), we have $\mathbf{Ker}\left(\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x)\right]\right) \subset \mathbf{Ker}\left(DF(x)\right)$. Thus, by Lemma A.3 we have that

$$\mathbf{Im}\left(DF(x)\right) = \left(\mathbf{Ker}\left(DF(x)^\top\right)\right)^\perp = \left(\mathbf{Ker}\left(DF(x)\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x)\right]DF(x)^\top\right)\right)^\perp, \tag{A.13}$$

where the second equality is obtained by Lemma A.3. Now from the definition of $\rho(x)$ in (2.20), we have

$$\begin{aligned}\rho(x) &\overset{(A.13)}{=} \min_{v \in \left(\mathbf{Ker}\left(DF(x)\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x)\right]DF(x)^\top\right)\right)^\perp/\{0\}} \frac{v^\top DF(x)\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x)\right]DF(x)^\top v}{\|v\|^2} \\ &= \lambda_{\min}^+\left(DF(x)\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x)\right]DF(x)^\top\right) > 0.\end{aligned}$$

It now follows that $\rho > 0$, since the definition of $\rho$ in (2.21) is given by minimizing $\rho(x)$ over the closed bounded set $\{x \mid \|x - x^*\| \leq \|x^0 - x^*\|\}$. Next, given $x \in \{x \mid \|x - x^*\| \leq \|x^0 - x^*\|\}$, since $F(x) \in \mathbf{Im}(DF(x)^\top)$ by (2.22) and notice that $\mathbf{Im}(DF(x)^\top) = \mathbf{Im}(DF(x)^\top DF(x))$, there exists $v \in \mathbb{R}^m$ s.t. $F(x) = DF(x)^\top DF(x)v$.

If $F(x) \neq 0$, then $DF(x)v \in \mathbf{Im}\left(DF(x)\right)/\{0\}$, we have

$$\begin{aligned} \|F(x)\|^2_{\mathbb{E}[\mathbf{H_S}(x)]} &= v^\top DF(x)^\top DF(x)\mathbb{E}\left[\mathbf{H_S}(x)\right] DF(x)^\top DF(x)v \\ &\overset{(2.20)}{\geq} \rho(x) v^\top DF(x)^\top DF(x)v. \end{aligned} \tag{A.14}$$

Since $F(x) = DF(x)^\top DF(x)v$ and $\mathbf{Im}(DF(x)^\top) \oplus \mathbf{Ker}(DF(x)) = \mathbb{R}^m$,[1] we have that

$$\exists!\ y \in \mathbf{Ker}(DF(x)) \subset \mathbb{R}^m \text{ s.t. } v = (DF(x)^\top DF(x))^\dagger F(x) + y.$$

Thus

$$DF(x)v = DF(x)(DF(x)^\top DF(x))^\dagger F(x) = (DF(x)^\top)^\dagger F(x).$$

Substituting this in (A.14), we have that

$$\|F(x)\|^2_{\mathbb{E}[\mathbf{H_S}(x)]} \geq \rho(x) \|F(x)\|^2_{\left(DF(x)^\top DF(x)\right)^\dagger} \geq \frac{\rho}{L^2} \|F(x)\|^2, \tag{A.15}$$

where on the last inequality, we use that $\sup_{x \in \{x \mid \|x - x^*\| \leq \|x^0 - x^*\|\}} \|DF(x)\| \leq L$ and $\rho(x) \geq \rho$ by the definition of $\rho$ in (2.21) .

If $F(x) = 0$, (A.15) still holds. Thus, for all $x \in \{x \mid \|x - x^*\| \leq \|x^0 - x^*\|\}$, (A.15) holds. Consequently by Theorem 2.7 and (2.17) under the star-convexity condition (2.16) with $\|x^t - x^*\| \leq \|x^0 - x^*\|$ for all $t \in \{0, \cdots, k-1\}$, we have that

$$\frac{\rho}{L^2}\mathbb{E}\left[\min_{t=0,\ldots,k-1} \left\|F(x^t)\right\|^2\right] \overset{(A.15)}{\leq} \mathbb{E}\left[\min_{t=0,\ldots,k-1} \left\|F(x^t)\right\|^2_{\mathbb{E}[\mathbf{H_S}(x^t)]}\right] \overset{(2.15)}{\leq} \frac{1}{k}\frac{\|x^0 - x^*\|^2}{\gamma\left(1 - \gamma\right)},$$

which after multiplying through by $L^2/\rho > 0$ concludes the proof. □

### A.2.4 Proof of Lemma 2.9

*Proof.* Since $\mathbf{Ker}(DF(x)^\top DF(x)) = \mathbf{Ker}(DF(x)) \overset{(2.24)}{\subset} \mathbf{Ker}(\mathbf{S}^\top)$, we have

$$\mathbf{Ker}\left(\left(\mathbf{S}^\top DF(x)^\top DF(x)\mathbf{S}\right)^\dagger\right) = \mathbf{Ker}\left(\mathbf{S}^\top DF(x)^\top DF(x)\mathbf{S}\right) = \mathbf{Ker}(\mathbf{S}), \tag{A.16}$$

[1] The operator $\oplus$ denotes the direct sum of two vector spaces.

where the last equality is obtained by Lemma A.3 with $\mathbf{Ker}(DF(x)^\top DF(x)) \subset \mathbf{Ker}(\mathbf{S}^\top)$. Thus, using Lemma A.3 again with $\mathbf{G} = \left(\mathbf{S}^\top DF(x)^\top DF(x)\mathbf{S}\right)^\dagger$, $\mathbf{W} = \mathbf{S}$ and $\mathbf{Ker}(\mathbf{G}) \subset \mathbf{Ker}(\mathbf{W})$ given by (A.16), we have that

$$\mathbf{Ker}(\mathbf{H}_\mathbf{S}(x)) \overset{(2.4)}{=} \mathbf{Ker}(\mathbf{S}\left(\mathbf{S}^\top DF(x)^\top DF(x)\mathbf{S}\right)^\dagger \mathbf{S}^\top) = \mathbf{Ker}(\mathbf{S}^\top) = \mathbf{Ker}(\mathbf{S}\mathbf{S}^\top). \tag{A.17}$$

As $\mathbf{H}_\mathbf{S}(x)$ is symmetric positive semi-definite $\forall\ \mathbf{S} \sim \mathcal{D}_x$, we have that

$$\begin{aligned}
&v \in \mathbf{Ker}\left(\mathbb{E}\left[\mathbf{H}_\mathbf{S}(x)\right]\right) \iff \mathbb{E}\left[\mathbf{H}_\mathbf{S}(x)\right] v = 0 \iff \|v\|^2_{\mathbb{E}[\mathbf{H}_\mathbf{S}(x)]} = 0 \quad (\text{as } \mathbb{E}\left[\mathbf{H}_\mathbf{S}(x)\right] \succeq 0) \\
&\iff \mathbb{E}\left[\|v\|^2_{\mathbf{H}_\mathbf{S}(x)}\right] = 0 \iff \int_\mathbf{S} \|v\|^2_{\mathbf{H}_\mathbf{S}(x)}\, d\mathbb{P}_{\mathcal{D}_x}(\mathbf{S}) = 0 \\
&\iff \|v\|^2_{\mathbf{H}_\mathbf{S}(x)} = 0\ \forall\ \mathbf{S} \sim \mathcal{D}_x \quad \left(\text{as } \|v\|^2_{\mathbf{H}_\mathbf{S}(x)} \geq 0\ \forall\ \mathbf{S}\right) \\
&\overset{\mathbf{H}_\mathbf{S}(x)\succeq 0}{\iff} v \in \mathbf{Ker}\left(\mathbf{H}_\mathbf{S}(x)\right)\ \forall\ \mathbf{S} \sim \mathcal{D}_x \iff v \in \bigcap_{\mathbf{S}\sim\mathcal{D}_x} \mathbf{Ker}\left(\mathbf{H}_\mathbf{S}(x)\right),
\end{aligned}$$

where we use $\bigcap_{\mathbf{S}\sim\mathcal{D}_x} \mathbf{Ker}\left(\mathbf{H}_\mathbf{S}(x)\right)$ to note the intersection of the random subsets $\mathbf{Ker}\left(\mathbf{H}_\mathbf{S}(x)\right)$ for all $\mathbf{S} \sim \mathcal{D}_x$. Similarly, we have $\mathbf{Ker}\left(\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right]\right) = \bigcap_{\mathbf{S}\sim\mathcal{D}_x} \mathbf{Ker}\left(\mathbf{S}\mathbf{S}^\top\right)$ because $\mathbf{S}\mathbf{S}^\top$ is also symmetric, positive semi-definite for all $\mathbf{S} \sim \mathcal{D}_x$. Thus we have

$$\begin{aligned}
\mathbf{Ker}(\mathbb{E}\left[\mathbf{H}_\mathbf{S}(x)\right]) \quad &= \quad \bigcap_{\mathbf{S}\sim\mathcal{D}_x} \mathbf{Ker}(\mathbf{H}_\mathbf{S}(x)) \\
&\overset{(A.17)}{=} \quad \bigcap_{\mathbf{S}\sim\mathcal{D}_x} \mathbf{Ker}(\mathbf{S}\mathbf{S}^\top) \;=\; \mathbf{Ker}(\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right]) \overset{(2.24)}{\subset} \mathbf{Ker}(DF(x)).
\end{aligned}$$

Consequently, by considering the complement of the above, we arrive at (2.22). □

### A.2.5 Proof of Lemma 2.10

*Proof.* First, $\mathbf{Ker}(DF(x)) \subset \mathbf{Ker}(\hat{\mathbf{S}}^\top DF(x)) = \mathbf{Ker}(\mathbf{S}^\top)$. Furthermore, from Lemma A.3 with $\mathbf{Ker}(\mathbb{E}[\hat{\mathbf{S}}\hat{\mathbf{S}}^\top]) \subset \mathbf{Ker}(DF(x)^\top)$, we conclude the proof with

$$\mathbf{Ker}\left(\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right]\right) \;=\; \mathbf{Ker}(DF(x)^\top \mathbb{E}[\hat{\mathbf{S}}\hat{\mathbf{S}}^\top] DF(x)) \;\subset \mathbf{Ker}(DF(x)).$$

□

### A.2.6 Proof of Lemma 2.11

*Proof.* For Gaussian sketches with $\mathbf{S}_{ij} \sim \mathcal{N}(0, \frac{1}{r})$, we have that $c = 1$. Indeed, since the mean is zero, off-diagonal elements of $\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right]$ are all zero. We note $\mathbf{S}_{i:}$ the $i$th row of $\mathbf{S}$, then the $i$th

diagonal element of the matrix $\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right]$ is given by

$$\mathbb{E}\left[\mathbf{S}_{i:}\mathbf{S}_{i:}^\top\right] = \sum_{j=1}^{\tau} \mathbb{E}\left[\mathbf{S}_{ij}^2\right] = \sum_{j=1}^{\tau} \frac{1}{\tau} = 1.$$

For the uniform subsampling sketch (2.5), we have again that off-diagonal elements are zero since the rows of $\mathbf{S}$ are orthogonal. The diagonal elements are constant with

$$\mathbb{E}\left[\mathbf{S}_{i:}\mathbf{S}_{i:}^\top\right] = \frac{1}{\binom{m}{\tau}} \sum_{C\subset\{1,\dots,m\},|C|=\tau, i\in C} 1 = \frac{\binom{m-1}{\tau-1}}{\binom{m}{\tau}} = \frac{\tau}{m}, \quad \text{for all } i = 1, \cdots, m.$$

□

### A.2.7 Proof of Lemma 2.13

*Proof.* Let $y \in \mathbb{R}^p$ and let $u \in F(\mathbb{R}^p) \cap \mathbf{Ker}\left(\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(y)\right]\right)$. $u \in F(\mathbb{R}^p)$ implies that $\exists x \in \mathbb{R}^p$ s.t. $F(x) = u$. Besides, $u \in \mathbf{Ker}\left(\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(y)\right]\right)$ implies that $\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(y)\right] F(x) = 0$. Now we apply (2.25) at point $x$ knowing that $f_y(x^*) = 0$:

$$\begin{aligned} & 0 \geq f_y(x) + \langle \nabla f_y(x), x^* - x\rangle + \frac{\mu}{2}\left\|x^* - x\right\|^2 \\ \Longrightarrow\ & 0 \geq 0 + \langle 0, x^* - x\rangle + \frac{\mu}{2}\left\|x^* - x\right\|^2 \quad (\text{as } \mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(y)\right] F(x) = 0) \Longleftrightarrow x = x^*. \end{aligned}$$

Thus $F(x) = u = 0$. We conclude $F(\mathbb{R}^p) \cap \mathbf{Ker}\left(\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(y)\right]\right) = \{0\}$, i.e. (2.10) holds.

Besides, let $x'$ be a global minimizer of $f_y(\cdot)$. Then $f_y(x') = f_y(x^*) = 0$ and $\nabla f_y(x') = 0$. Similarly, by applying (2.25) at point $x'$, we obtain $x' = x^*$. Consequently, $x^*$ is the unique minimizer of $f_y(\cdot)$ for all $y$, thus the unique solution to (2.1), according to (2.10) and Lemma 2.4.

□

### A.2.8 Proof of Theorem 2.14

*Proof.* Let $\delta_k \overset{\text{def}}{=} x^k - x^*$. By expanding the squares, similarly we have that

$$\begin{aligned} \mathbb{E}_k\left[\left\|\delta_{k+1}\right\|^2\right] &= \left\|\delta_k\right\|^2 - 2\gamma\left\langle \delta_k, \nabla f_k(x^k)\right\rangle + \gamma^2 \mathbb{E}_k\left[\left\|\nabla f_{\mathbf{S}_k,k}(x^k)\right\|^2\right] \\ &\overset{(2.25)}{\leq} (1-\gamma\mu)\left\|\delta_k\right\|^2 - 2\gamma(f_k(x^k) - f_k(x^*)) + \gamma^2 \mathbb{E}_k\left[\left\|\nabla f_{\mathbf{S}_k,k}(x^k)\right\|^2\right] \\ &\overset{(2.13)}{\leq} (1-\gamma\mu)\left\|\delta_k\right\|^2 - 2\gamma(1-\gamma)(f_k(x^k) - f_k(x^*)) \\ &\leq (1-\gamma\mu)\left\|\delta_k\right\|^2. \qquad \left(\text{since } \gamma(1-\gamma)(f_k(x^k) - f_k(x^*)) \geq 0\right) \end{aligned}$$

Now by taking total expectation, we have that

$$\mathbb{E}\left[\left\|x^{k+1}-x^*\right\|^2\right] \leq (1-\gamma\mu)\mathbb{E}\left[\left\|x^k-x^*\right\|^2\right] \leq (1-\gamma\mu)^{k+1}\left\|x^0-x^*\right\|^2.$$

Next, we show that $\mu \leq 1$. In fact, when we imply (2.25) at the point $x^k$, it shows

$$\begin{aligned}
(2.25) \quad &\overset{(2.13)}{\Longrightarrow} \quad f_k(x^*) \geq \frac{1}{2}\mathbb{E}_k\left[\left\|\nabla f_{\mathbf{S}_k,k}(x^k)\right\|^2\right] + \left\langle x^*-x^k, \nabla f_k(x^k)\right\rangle + \frac{\mu}{2}\|x^*-x^k\|^2 \\
&\Longleftrightarrow \quad f_k(x^*) \geq \frac{1}{2}\mathbb{E}_k\left[\left\|x^*-\left(x^k-\nabla f_{\mathbf{S}_k,k}(x^k)\right)\right\|^2\right] - \frac{1-\mu}{2}\|x^*-x^k\|^2 \\
&\overset{f_k(x^*)=0}{\Longrightarrow} \quad (1-\mu)\|x^*-x^k\|^2 \geq \mathbb{E}_k\left[\left\|x^*-\left(x^k-\nabla f_{\mathbf{S}_k,k}(x^k)\right)\right\|^2\right] \geq 0.
\end{aligned}$$

Thus $\mu \leq 1$. □

## A.3 Proof of Section 2.5

### A.3.1 Proof of Corollary 2.15

*Proof.* From (2.11), we have that

$$\nabla f_x(x) = DF(x)(DF(x)^\top DF(x))^\dagger F(x) = (DF(x)^\top)^\dagger F(x) = -n(x). \tag{A.18}$$

Substituting (2.32) and (A.18) in (2.16) yields (2.34). Next, for $\mathbf{S} = \mathbf{I}_m$, we have that

$$\mathbf{Im}(\mathbb{E}\left[\mathbf{H}_\mathbf{S}(x)\right])) = \mathbf{Im}((DF(x)^\top DF(x))^\dagger) = \mathbf{Im}(DF(x)^\top DF(x)) = \mathbf{Im}(DF(x)^\top).$$

Thus, we have that $F(x) \in \mathbf{Im}(DF(x)^\top) \subset \mathbf{Im}(\mathbb{E}\left[\mathbf{H}_\mathbf{S}(x)\right]))$, i.e. (2.22) holds. So all the conditions in Corollary 2.8 are verified. Since $\mathbf{S} = \mathbf{I}_m$, we have that $\rho(x) = 1$ for all $x$, so $\rho = 1$. Furthermore, because we assume that $DF(\cdot)$ is continuous and the iterates $x^k$ are in a closed bounded convex set (2.17) which is implied by (2.16) from Theorem 2.7, there exists $L > 0$ s.t. $\|DF(x^k)\| \leq L$ for all the iterates. Finally, by Corollary 2.8, the iterates converge sublinearly according to (2.23) which in this case is given by (2.35). □

### A.3.2 Proof of Theorem 2.16

*Proof.* First, we prove (I) $\implies$ (II). Assume that (I) holds. Since $DF(x)$ is invertible, (2.33) holds trivially. By 13.3.4 in Ortega and Rheinboldt (2000), we know that there exists a unique $x^* \in \mathbb{R}^p$ s.t. $F(x^*) = 0$. It remains to verify if (2.34) holds for $k \geq 1$. First, note that the

invertibility of $DF(x^k)$ gives

$$f_k(x^k) \;=\; \frac{1}{2}\left\|F(x^k)\right\|^2_{(DF_k^\top DF_k)^\dagger} \;=\; \frac{1}{2}\left\|(DF_k^\top)^{-1}F_k\right\|^2 \overset{(2.36)}{=} \frac{1}{2}\left\|x^{k+1}-x^k\right\|^2, \tag{A.19}$$

with abbreviations $f_k(x^k) \equiv f_{x^k}(x^k)$, $F_k \equiv F(x^k)$ and $DF_k \equiv DF(x^k)$. Furthermore,

$$\nabla f_k(x^k) \;=\; DF_k(DF_k^\top DF_k)^{-1}F(x^k) \;=\; (DF_k^\top)^{-1}F(x^k) \overset{(2.36)}{=} x^k - x^{k+1}. \tag{A.20}$$

Thus we can re-write the right hand side of the star-convexity assumption (2.14) as

$$\begin{aligned} f_k(x^k) + \left\langle \nabla f_k(x^k), x^* - x^k\right\rangle &\overset{(A.19)+(A.20)}{=} \frac{1}{2}\left\|x^{k+1}-x^k\right\|^2 + \left\langle x^k - x^{k+1}, x^* - x^k\right\rangle \\ &= \frac{1}{2}\left\|x^{k+1}-x^k\right\|^2 + \left\langle x^k - x^{k+1}, x^{k+1} - x^k + x^* - x^{k+1}\right\rangle \\ &= -\frac{1}{2}\left\|x^{k+1}-x^k\right\|^2 + \left\langle x^k - x^{k+1}, x^* - x^{k+1}\right\rangle. \end{aligned}$$

From (I), we induce by Lemma 3.1 in Deuflhard (2011) that NR is component wise monotone with $x^* \le x^{k+1} \le x^k$ for $k \ge 1$. Thus $x^k - x^{k+1} \ge 0$ and $x^* - x^{k+1} \le 0$ component wise and consequently, $\left\langle x^k - x^{k+1}, x^* - x^{k+1}\right\rangle \le 0$. Thus it follows that

$$f_k(x^k) + \left\langle \nabla f_k(x^k), x^* - x^k\right\rangle \;\le\; 0 \;=\; f_k(x^*).$$

Thus (2.34) holds for $k \ge 1$ and this concludes that (I) $\implies$ (II).

We now prove that (II) does **not** imply (I). Consider the example $F(x) = Ax - b$, where $A \in \mathbb{R}^{p\times p}$ is invertible and $b \in \mathbb{R}^p$. Thus, $DF(x) = A^\top$ is invertible and (2.33) holds. As for (2.34), let $x^*$ be the solution, i.e. $Ax^* = b$, we have that

$$f_k(x) = \frac{1}{2}\left\|F(x)\right\|^2_{(DF(x_k)^\top DF(x_k))^{-1}} = \frac{1}{2}\left\|A(x-x^*)\right\|^2_{(AA^\top)^{-1}} = \frac{1}{2}\left\|x-x^*\right\|^2,$$

which is a convex function and so (2.34) holds and thus (II) holds. However, (I) does not necessarily hold. Indeed, if $A = -\mathbf{I}_p$, then $DF(x)$ is not element-wise positive. □

### A.3.3 The monotone convergence theory of NR with stepsize $\gamma < 1$

The monotone convergence theory of the NR in both Ortega and Rheinboldt (2000) and Deuflhard (2011) need to have the stepsize $\gamma = 1$. If $\gamma < 1$ which is the case in our convergence Theorem 2.7 and Corollary 2.8, the iterates $\{x^k\}_{k\ge1}$ under the set of assumptions (I) proposed in Theorem 2.16 are no longer guaranteed to be component wise monotonically decreasing. Here we investigate alternatives. In particular, we consider the case in 1-dimension for function $F = \phi : \mathbb{R} \to \mathbb{R}$.

**Lemma A.4.** *Let $x^k$ be the iterate of the NR method with stepsize $\gamma < 1$ for solving $\phi(x) = 0$, that is*

$$x^{k+1} = x^k - \gamma \frac{\phi(x^k)}{\phi'(x^k)}. \tag{A.21}$$

*If $\phi$ satisfies the set of assumptions $(I)$ proposed in Theorem 2.16, then*

- *$(a)$ The iterates of the ordinary NR method (2.36) are necessarily monotonically decreasing.*
- *$(b)$ The iterates of the NR method (A.21) with $\gamma < 1$ are not necessarily monotonically decreasing.*
- *$(c)$ Assumption (2.33) holds; for $\frac{1}{2} \leq \gamma < 1$, there exists $k' \geq 0$ such that for all $k \neq k'$, there exists a unique $x^*$ that satisfies Assumption 2.1 and the iterates $x^k$ and the optimum $x^*$ satisfy (2.34).*
- *$(d)$ The iterates $x^k$ following the NR method (A.21) with $\frac{1}{2} \leq \gamma < 1$ converge sublinearly to a zero of $\phi$.*

**Remark of (a)** Even though this result is known and generalized in $d$-dimension in Ortega and Rheinboldt (2000) and Deuflhard (2011), we stress it here to highlight the impact of the stepsize $\gamma$ in the NR method and leverage the analysis of (a) in the special 1-dimensional case to prove (b).

*Proof.* If $\phi$ satisfies (I), then $\phi$ is convex and $\phi'^{-1} > 0$, which implies $\phi'' \geq 0$ and $\phi' > 0$. From $\phi' > 0$, we obtain that $\phi$ is strictly increasing. Besides, from (I), $\exists\, x, y \in \mathbb{R}$ such that $\phi(x) \leq 0 \leq \phi(y)$. This with the strictly increase of $\phi$ induces that $\exists!\ x^*$ such that $\phi(x^*) = 0$, i.e. $x^*$ satisfies Assumption 2.1. So $\forall x < x^*$, $\phi(x) < 0$ and $\forall x > x^*$, $\phi(x) > 0$. Now consider the following two functions

$$u(x) \overset{\text{def}}{=} x - \frac{\phi(x)}{\phi'(x)} \qquad \text{and} \qquad U(x) \overset{\text{def}}{=} x - \gamma \frac{\phi(x)}{\phi'(x)}$$

which are exactly the updates of the ordinary NR (2.36) and the NR (A.21) with a stepsize $\gamma \in (0, 1)$, respectively. We first analyze the behaviour of the function $u$ and show (a), which can be formulated as $x^* \leq x^{k+1} \leq x^k$ for all $k \geq 1$. The derivative of $u$ is

$$u'(x) = \frac{\phi(x)\phi''(x)}{\phi'(x)^2}.$$

By the sign of functions $\phi$, $\phi'$ and $\phi''$, we know that if $x > x^*$, then $u'(x) \geq 0$ and if $x < x^*$, then $u'(x) \leq 0$. This implies that the function $u$ is increasing in $[x^*, +\infty[$ and decreasing in $]-\infty, x^*]$.

Overall, we have

$$\begin{cases} \min_{x\in\mathbb{R}} u(x) = u(x^*) \overset{\phi(x^*)=0}{=} x^*, & \text{(A.22a)} \\ u(x) < x \text{ and } u \text{ increasing}, \quad \text{when } x > x^*, & \text{(A.22b)} \\ u(x) > x \text{ and } u \text{ decreasing}, \quad \text{when } x < x^*. & \text{(A.22c)} \end{cases}$$

Consequently, $x^* \leq x^{k+1}$ is obtained by $x^* \overset{\text{(A.22a)}}{=} \min u(x) \leq u(x^k) = x^{k+1}$. As for the inequality $x^{k+1} \leq x^k$, $x^* \overset{\text{(A.22a)}}{=} \min u(x) \leq u(x^{k-1}) = x^k$ for $k \geq 1$ and $x^{k+1} = u(x^k) \overset{\text{(A.22b)}}{\leq} x^k$ as $x^k \geq x^*$.

To show (b), we analyze the behavior of the function $U$. Consider its derivative

$$U'(x) = (1-\gamma) + \gamma \frac{\phi(x)\phi''(x)}{\phi'(x)^2}.$$

By the sign of functions $\phi$, $\phi'$ and $\phi''$, if $x > x^*$, $U'(x) > 0$. However, $U'(x^*) \overset{\phi(x^*)=0}{=} 1-\gamma > 0$, which implies $\min U(x) < x^*$. Here we include the case where $\min U(x) = -\infty$. Also by the sign of functions $\phi$ and $\phi'$ and $\gamma < 1$, when $x > x^*$, we have $u(x) < U(x)$ and $u(x) > U(x)$ for $x < x^*$. In summary, we have

$$\begin{cases} \min_{x\in\mathbb{R}} U(x) < U(x^*) \overset{\phi(x^*)=0}{=} u(x^*) \overset{\text{(A.22a)}}{=} x^*, & \text{(A.23a)} \\ u(x) < U(x) < x \text{ and } U \text{ increasing}, \quad \text{when } x > x^*, & \text{(A.23b)} \\ u(x) > U(x) > x \text{ when } x < x^*. & \text{(A.23c)} \end{cases}$$

In NR with stepsize $\gamma < 1$, consider $x^0 \in \mathbb{R}$. We discuss different cases based on the comparison between $x^0$ and $x^*$.

If $x^0 \geq x^*$ named as case **(i)**, by induction, we get $x^* \overset{\text{(A.23a)}}{=} U(x^*) \overset{\text{(A.23b)}}{\leq} = U(x^k) = x^{k+1} \overset{\text{(A.23b)}}{\leq} x^k$ for all $k \geq 0$. In this case, the iterates decrease monotonically.

If $x^0 < x^*$, there are two cases, named as case **(ii)**, for all $k \in \mathbb{N}$, $U(x^k) \leq x^*$, and case **(iii)**, $\exists k' \in \mathbb{N}$, $U(x^{k'}) > x^*$.

If **(ii)** holds, we have that the iterates increase monotonically. Indeed, by **(ii)** and by induction, we get $x^k \overset{\text{(A.23c)}}{\leq} U(x^k) = x^{k+1} \overset{\textbf{(ii)}}{\leq} x^*$ for all $k \in \mathbb{N}$.

Otherwise, we are in case **(iii)**. Let $k'$ be the smallest index that $U(x^{k'}) > x^*$. Then we conclude that the iterates $\{x^k\}_{k\geq 0}$ increase monotonically when $k \leq k'$ and $\{x^k\}_{k\geq k'+1}$ decrease monotonically. In fact, by the definition of $k'$, we know that for $k \in [\![0, k'-1]\!]$, $U(k) \leq x^*$. Then by induction as in case **(ii)** but for $k \leq k'$, we get $\{x^k\}_{k\geq 0}$ increase monotonically when $k \leq k'$.

When $k \geq k'+1$, by induction as in case **(i)** but for $U(x^{k'}) = x^{k'+1} > x^*$, we get $\{x^k\}_{k\geq k'+1}$ decrease monotonically. We thus observe (b).

Statement (c) follows from the proof of Theorem 2.16 in 1-dimension in taking account the stepsize $\gamma < 1$. Then (2.33) holds and (2.34) becomes

$$f_k(x^k) + \left\langle \nabla f_k(x^k), x^* - x^k \right\rangle = \underbrace{\frac{1-2\gamma}{2\gamma^2}(x^{k+1}-x^k)^2}_{\leq 0 \text{ as } \frac{1}{2}\leq\gamma \text{ in (c)}} + \frac{1}{\gamma}\underbrace{(x^k - x^{k+1})(x^* - x^{k+1})}_{\stackrel{\text{def}}{=}(*)} \tag{A.24}$$

in considering

$$f_k(x^k) = \frac{1}{2}\left(\frac{\phi(x^k)}{\phi'(x^k)}\right)^2 \stackrel{(A.21)}{=} \frac{1}{2\gamma^2}(x^{k+1}-x^k)^2,$$
$$\nabla f_k(x^k) = \frac{\phi(x^k)}{\phi'(x^k)} \stackrel{(A.21)}{=} \frac{1}{\gamma}\left(x^k - x^{k+1}\right),$$

with $\phi = F$ and $\phi' = DF$. To get (2.34) hold, from (A.24), it suffices to prove $(*) \leq 0$.

By the analysis of (b), we know: in case **(i)**, $(*) \leq 0$ for all $k \geq 0$ as $x^* \leq x^{k+1} \leq x^k$; in case **(ii)**, $(*) \leq 0$ for all $k \geq 0$ as $x^* \geq x^{k+1} \geq x^k$; finally in case **(iii)**, for $k \neq k'$, $(*) \leq 0$ as $x^* \geq x^{k+1} \geq x^k$ for $k \leq k'-1$ and $x^* \leq x^{k+1} \leq x^k$ for $k \geq k'+1$. So in all cases, $(*) \leq 0$ for all $k$ or for $k \neq k'$. We thus obtain (c).

It remains to show (d), which is simply obtained by (c) and Corollary 2.8, as (2.34) holds for all iterates $x^k$ except for just one iterate $x^{k'}$ potentially. □

The monotone convergence theory is based on assumptions (I) with stepsize $\gamma = 1$. Under the same assumptions with $\gamma < 1$, such theory may not hold. Indeed, following the analysis in Lemma A.4 in 1-dimension case, by (A.23c) we do not have the monotone property for the function $U$ when $x < x^*$. That is the reason why (b) happens but not (a) in Lemma A.4. In $d$-dimension case, without such monotone property for the function $U$, $\{x^k\}$ is not guaranteed to be monotone, which is the main clue in their theory's proof. However, with stepsize $\gamma < 1$, assumptions (I) can still imply our Assumptions 2.1, (2.33) and (2.34) under constraint $\frac{1}{2} \leq \gamma < 1$ in 1-dimension case. In addition, though our theory does not either require any constraint for stepsize $\gamma < 1$ or guarantee that the NR method is monotonic in terms of the iterates component wisely, we still guarantee the sublinear global convergence. We thus conclude that Assumptions 2.1, (2.33) and (2.34) are strictly weaker than the assumptions used in the monotone convergence theory in Ortega and Rheinboldt (2000) and Deuflhard (2011), albeit for different step sizes.

## A.4 Proof of Section 2.7

### A.4.1 Proof of Lemma 2.17

In our upcoming proof of Lemma 2.17, we still need the following lemma.

**Lemma A.5.** *Let $\phi_i$ be twice differentiable and strictly convex for $i = 1, \dots, n$. The Jacobian $DF(x)^\top$ of $F(x)$ defined in (2.42) is invertible for all $x \in \mathbb{R}^{(n+1)d}$.*

*Proof.* Let $x \in \mathbb{R}^{(n+1)d}$. Let $y \overset{\text{def}}{=} (u; v_1; \cdots; v_n) \in \mathbb{R}^{(n+1)d}$ with $u, v_1, \cdots, v_n \in \mathbb{R}^d$ such that $DF(x)y = 0$. The transpose of the Jacobian of $F(x)$ is given by

$$DF(x) = \begin{bmatrix} 0 & \mathbf{I}_d & \cdots & \mathbf{I}_d \\ \frac{1}{n}\nabla^2\phi_1(\alpha_1) & & & \\ \vdots & & -\mathbf{I}_{nd} & \\ \frac{1}{n}\nabla^2\phi_n(\alpha_n) & & & \end{bmatrix}. \tag{A.25}$$

From $DF(x)y = 0$ and (A.25), we obtain

$$\sum_{i=1}^n v_i = 0, \quad \text{and} \quad \frac{1}{n}\nabla^2\phi_i(\alpha_i)u = v_i \quad \text{for all } i = 1, \cdots, n.$$

Plugging the second equation in the first one gives $\left(\frac{1}{n}\sum_{i=1}^n \nabla^2\phi_i(\alpha_i)\right) u = 0$. Since every $\phi_i$ is twice differentiable and strictly convex, we have $\nabla^2\phi_i(\alpha_i) > 0$. This implies $\frac{1}{n}\sum_{i=1}^n \nabla^2\phi_i(\alpha_i) > 0$, and is thus invertible. Consequently $u = 0$ and $v_i = 0$ from which we conclude that the Jacobian $DF(x)^\top$ is invertible. □

Now we can give the proof of Lemma 2.17.

*Proof.* Consider an update of SNR (line 4 in Algorithm 1) with $F$ defined in (2.42), the sketching matrix $\mathbf{S}_k$ defined in (2.43) , and stepsize $\gamma = 1$ at the $k$th iteration. By Lemma A.5, we have that $DF(x)$ is invertible and thus Assumption 2.3 holds. By (2.7), the SNR update (line 4 in Algorithm 1) can be re-written as

$$x^{k+1} = \operatorname{argmin} \left\|w - w^k\right\|^2 + \sum_{i=1}^n \left\|\alpha_i - \alpha_i^k\right\|^2 \text{ s.t. } \mathbf{S}_k^\top DF(x^k)^\top (x - x^k) = -\mathbf{S}_k^\top F(x^k). \tag{A.26}$$

Plugging (2.42), (2.43) and (A.25) into the constraint in (A.26) gives

$$\begin{aligned}
&\begin{bmatrix} \mathbf{I}_d & 0 \\ \frac{1}{n}\nabla^2\phi_1(\alpha_1^k) & \\ \vdots & \mathbf{I}_{B_n} \\ \frac{1}{n}\nabla^2\phi_n(\alpha_n^k) & \end{bmatrix}^\top \begin{bmatrix} 0 & \mathbf{I}_d & \cdots & \mathbf{I}_d \\ \frac{1}{n}\nabla^2\phi_1(\alpha_1) & & & \\ \vdots & & -\mathbf{I}_{nd} & \\ \frac{1}{n}\nabla^2\phi_n(\alpha_n) & & & \end{bmatrix}^\top \begin{bmatrix} w - w^k \\ \alpha_1 - \alpha_1^k \\ \vdots \\ \alpha_n - \alpha_n^k \end{bmatrix} \\
&= - \begin{bmatrix} \mathbf{I}_d & 0 \\ \frac{1}{n}\nabla^2\phi_1(\alpha_1^k) & \\ \vdots & \mathbf{I}_{B_n} \\ \frac{1}{n}\nabla^2\phi_n(\alpha_n^k) & \end{bmatrix}^\top \begin{bmatrix} \frac{1}{n}\sum_{i=1}^n \nabla\phi_i(\alpha_i^k) \\ w^k - \alpha_1^k \\ \vdots \\ w^k - \alpha_n^k \end{bmatrix}.
\end{aligned}$$

After simplifying the above matrix multiplications, we have that (A.26) is given by

$$\begin{aligned}
x^{k+1} = [w^{k+1}; \alpha_1^{k+1}; \cdots; \alpha_n^{k+1}] = \operatorname{argmin} \left\|w - w^k\right\|^2 + \sum_{i=1}^n \left\|\alpha_i - \alpha_i^k\right\|^2 \\
\text{s. t.} \quad \frac{1}{n}\sum_{i=1}^n \nabla^2\phi_i(\alpha_i^k)(w - \alpha_i^k) = -\frac{1}{n}\sum_{i=1}^n \nabla\phi_i(\alpha_i^k), \\
w = \alpha_j, \quad \text{for } j \in B_n. \qquad \text{(A.27)}
\end{aligned}$$

To solve (A.27), first note that $\alpha_i^{k+1} = \alpha_i^k$ for $i \notin B_n$, since there is no constraint on the variable $\alpha_i$ in this case. Furthermore, by the invertibility of $\frac{1}{n}\sum_{i=1}^n \nabla^2\phi_i(\alpha_i^k)$, we have that (A.27) has a unique solution s.t. $\alpha_j = w$ for all $j \in B_n$ and

$$w = \left(\frac{1}{n}\sum_{i=1}^n \nabla^2\phi_i(\alpha_i^k)\right)^{-1}\left(\frac{1}{n}\sum_{i=1}^n \nabla^2\phi_i(\alpha_i^k)\alpha_i^k - \frac{1}{n}\sum_{i=1}^n \nabla\phi_i(\alpha_i^k)\right).$$

Concluding, we have that the SNR update (A.27) is given by

$$\begin{aligned}
w^{k+1} &= \left(\frac{1}{n}\sum_{i=1}^n \nabla^2\phi_i(\alpha_i^k)\right)^{-1}\left(\frac{1}{n}\sum_{i=1}^n \nabla^2\phi_i(\alpha_i^k)\alpha_i^k - \frac{1}{n}\sum_{i=1}^n \nabla\phi_i(\alpha_i^k)\right), \\
\alpha_i^{k+1} &= \begin{cases} w^{k+1} & \text{if } i \in B_n \\ \alpha_i^k & \text{if } i \notin B_n \end{cases},
\end{aligned}$$

which is exactly the Stochastic Newton method's updates (2.40) and (2.41) in (Kovalev et al., 2019). □

### A.4.2 Proof of Lemma 2.19

*Proof.* First, we show that $\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right]$ is invertible $\forall x \in \mathbb{R}^{(n+1)d}$. By the definition of $\mathbf{S}$ in (2.43),

$$\mathbf{S}\mathbf{S}^\top = \begin{bmatrix} \mathbf{I}_d & \frac{1}{n}\nabla^2\phi_1(\alpha_1) & \cdots & \frac{1}{n}\nabla^2\phi_n(\alpha_n) \\ \frac{1}{n}\nabla^2\phi_1(\alpha_1) & & & \\ \vdots & & \mathbf{I}_{B_n}\mathbf{I}_{B_n}^\top + \mathbf{M} & \\ \frac{1}{n}\nabla^2\phi_n(\alpha_n) & & & \end{bmatrix},$$

where $\mathbf{M} = \{\mathbf{M}_{ij}\}_{1\le i\le n, 1\le j\le n}$ is divided into $n \times n$ contiguous blocks of size $d \times d$ with each block $\mathbf{M}_{ij}$ defined as the following

$$\mathbf{M}_{ij} \overset{\text{def}}{=} \frac{1}{n}\nabla^2\phi_i(\alpha_i) \cdot \frac{1}{n}\nabla^2\phi_j(\alpha_j) \in \mathbb{R}^{d\times d} \qquad \text{and} \qquad \mathbf{M} \in \mathbb{R}^{nd\times nd}.$$

Taking the expectation over $\mathbf{S}$ w.r.t. the distribution $\mathcal{D}_{(w,\alpha_1,\cdots,\alpha_n)}$ gives

$$\begin{aligned} \mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right] &= \begin{bmatrix} \mathbf{I}_d & \frac{1}{n}\nabla^2\phi_1(\alpha_1) & \cdots & \frac{1}{n}\nabla^2\phi_n(\alpha_n) \\ \frac{1}{n}\nabla^2\phi_1(\alpha_1) & & & \\ \vdots & & \frac{\tau}{n}\mathbf{I}_{nd} + \mathbf{M} & \\ \frac{1}{n}\nabla^2\phi_n(\alpha_n) & & & \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I}_d \\ \frac{1}{n}\nabla^2\phi_1(\alpha_1) \\ \vdots \\ \frac{1}{n}\nabla^2\phi_n(\alpha_n) \end{bmatrix} \begin{bmatrix} \mathbf{I}_d \\ \frac{1}{n}\nabla^2\phi_1(\alpha_1) \\ \vdots \\ \frac{1}{n}\nabla^2\phi_n(\alpha_n) \end{bmatrix}^\top + \frac{\tau}{n}\begin{bmatrix} 0 & 0 \\ 0 & \mathbf{I}_{nd} \end{bmatrix}. \end{aligned} \tag{A.28}$$

$\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right]$ is symmetric, positive semi-definite. Let $(u; v_1; \cdots; v_n) \in \mathbb{R}^{(n+1)d}$ s.t.

$$(u; v_1; \cdots; v_n)^\top \mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right](u; v_1; \cdots; v_n) = 0.$$

From (A.28), we obtain

$$\left\| \left[\mathbf{I}_d; \frac{1}{n}\nabla^2\phi_1(\alpha_1); \cdots; \frac{1}{n}\nabla^2\phi_n(\alpha_n)\right]^\top [u; v_1; \cdots; v_n] \right\|^2 + \frac{\tau}{n}\sum_{i=1}^n \|v_i\|^2 = 0.$$

Since both terms are non negative, we obtain $\sum_{i=1}^n \|v_i\|^2 = 0 \implies \forall\, i, v_i = 0$, and then $u = 0$. This confirms that $\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right]$ is positive-definite, thus invertible and $\mathbf{Ker}(\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right]) = \{0\}$. Besides, from Lemma A.5, we get $DF(x)$ invertible. Thus $F(x) \in \mathbf{Im}(DF(x)^\top)$ and $\mathbf{Ker}(DF(x)) = \{0\}$. We have (2.24) hold. By Lemma 2.9, we have that (2.22) holds for all $x \in \mathbb{R}^{(n+1)d}$. □

### A.4.3 Stochastic Newton method with relaxation

Consider the function $P(\cdot)$ defined in (2.39). By the analysis of Lemma 2.17, we can even develop a variant of SNM in the case stepsize $\gamma < 1$ and we call the method *Stochastic Newton method with relaxation*. The updates are the following

$$w^{k+1} = \gamma\left(\frac{1}{n}\sum_{i=1}^{n}\nabla^2\phi_i(\alpha_i^k)\right)^{-1}\left(\frac{1}{n}\sum_{i=1}^{n}\nabla^2\phi_i(\alpha_i^k)\alpha_i^k - \frac{1}{n}\sum_{i=1}^{n}\nabla\phi_i(\alpha_i^k)\right) + (1-\gamma)w^k, \quad \text{(A.29)}$$

$$\alpha_i^{k+1} = \begin{cases} w^{k+1} - (1-\gamma)(w^k - \alpha_i^k) & \text{if } i \in B_n \\ \alpha_i^k & \text{if } i \notin B_n \end{cases}, \quad \text{(A.30)}$$

In the rest of Appendix A.4.3, we use the shorthand $x \overset{\text{def}}{=} (w;\alpha_1;\cdots;\alpha_n) \in \mathbb{R}^{(n+1)d}$ and $x^k \overset{\text{def}}{=} (w^k;\alpha_1^k;\cdots;\alpha_n^k)$ the iterates of SNR in Lemma 2.17 with stepsize $\gamma < 1$ at the $k$th iteration.

**Lemma A.6.** *At each iteration $k$, the updates of SNR $x^k$ are equal to the updates (A.29), (A.30) of SNM with relaxation.*

*Proof.* Following the proof of Lemma 2.17 and taking account the stepsize $\gamma$, by (A.26) and (2.7), the updates of SNR $x^{k+1}$ at $(k+1)$th iteration are given by

$$x^{k+1} = \operatorname{argmin}\left\|w - w^k\right\|^2 + \sum_{i=1}^{n}\left\|\alpha_i - \alpha_i^k\right\|^2 \text{ s.t. } \mathbf{S}^\top DF(x^k)^\top(x - x^k) = -\gamma\mathbf{S}^\top F(x^k), \quad \text{(A.31)}$$

where the sketching matrix $\mathbf{S} \sim \mathcal{D}_{x^k}$ is defined in (2.43). Similar to (A.27), (A.31) can be re-written as

$$\begin{aligned} x^{k+1} &= \operatorname{argmin}\left\|w - w^k\right\|^2 + \sum_{i=1}^{n}\left\|\alpha_i - \alpha_i^k\right\|^2 \quad \text{(A.32)} \\ &\text{s.t. } \frac{1}{n}\sum_{i=1}^{n}\nabla^2\phi_i(\alpha_i^k)(w - w^k) = -\gamma\left(\frac{1}{n}\sum_{i=1}^{n}\nabla\phi_i(\alpha_i^k) + \frac{1}{n}\sum_{i=1}^{n}\nabla^2\phi_i(\alpha_i^k)(w^k - \alpha_i^k)\right), \\ &\quad w - \alpha_i = (1-\gamma)(w^k - \alpha_i^k), \quad \text{for } i \in B_n. \end{aligned}$$

Similarly, note that if $i \notin B_n$, then $\alpha_i^{k+1} = \alpha_i^k$, since there is no constraint on the variable $\alpha_i$ in this case. Then by the invertibility of $\frac{1}{n}\sum_{i=1}^{n}\nabla^2\phi_i(\alpha_i^k)$, we have the unique solution of (A.32), which is

$$\begin{aligned} w^{k+1} &= \gamma\left(\frac{1}{n}\sum_{i=1}^{n}\nabla^2\phi_i(\alpha_i^k)\right)^{-1}\left(\frac{1}{n}\sum_{i=1}^{n}\nabla^2\phi_i(\alpha_i^k)\alpha_i^k - \frac{1}{n}\sum_{i=1}^{n}\nabla\phi_i(\alpha_i^k)\right) + (1-\gamma)w^k, \\ \alpha_i^{k+1} &= w^{k+1} - (1-\gamma)(w^k - \alpha_i^k), \quad \text{for } i \in B_n. \end{aligned}$$

Overall, we have

$$w^{k+1} = \gamma\left(\frac{1}{n}\sum_{i=1}^{n}\nabla^2\phi_i(\alpha_i^k)\right)^{-1}\left(\frac{1}{n}\sum_{i=1}^{n}\nabla^2\phi_i(\alpha_i^k)\alpha_i^k - \frac{1}{n}\sum_{i=1}^{n}\nabla\phi_i(\alpha_i^k)\right) + (1-\gamma)w^k,$$
$$\alpha_i^{k+1} = \begin{cases} w^{k+1} - (1-\gamma)(w^k - \alpha_i^k) & \text{if } i \in B_n \\ \alpha_i^k & \text{if } i \notin B_n \end{cases},$$

which is exactly the updates (A.29) and (A.30) in SNM with relaxation. □

Notice that both the original SNM and SNM with relaxation have the same complexity. Consequently, Theorem 2.7 allows us to develop the following global convergence theory of SNM with relaxation $\gamma$.

**Corollary A.7.** *Consider the iterate* $x^k = \left(w^k; \alpha_1^k; \cdots; \alpha_n^k\right)$ *given by* (A.29) *and* (A.30). *Note* $x^* \overset{def}{=} (w^*; w^*; \cdots; w^*) \in \mathbb{R}^{(n+1)d}$ *where* $w^*$ *is the stationary point of* $\nabla P(w)$ *that satisfies*

$$f_{x^k}(x^*) \geq f_{x^k}(x^k) + \left\langle \nabla f_{x^k}(x^k), x^* - x^k \right\rangle, \quad \text{for all } k \in \mathbb{N}, \tag{A.33}$$

*then*

$$\min_{t=0,\ldots,k-1} \mathbb{E}\left[f_{x^t}(x^t)\right] \leq \frac{1}{k}\sum_{t=0}^{k-1}\mathbb{E}\left[f_{x^t}(x^t)\right] \leq \frac{1}{k}\frac{\left\|x^0 - x^*\right\|^2}{2\gamma\left(1-\gamma\right)}.$$

*Proof.* Equation (A.33) with the assumption that $w^*$ is the stationary point of $\nabla P(w)$ implies that Assumption 2.1 and 2.6 hold. Then we conclude the proof by Theorem 2.7. □

## A.5 Proof of Lemma 2.22

*Proof.* First, we show that $\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right]$ is invertible. By Definition 2.21, it is straightforward to verify that

$$\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right] = \begin{bmatrix} \frac{(1-b)\tau_d}{n}\mathbf{I}_d & 0 \\ 0 & \frac{b\tau_n}{n}\mathbf{I}_n \end{bmatrix}$$

is invertible and $\mathbf{Ker}(\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right]) = \{0\}$. Now we show the Jacobian $DF^\top(x)$ invertible. Let $x = [\alpha; w] \in \mathbb{R}^{n+d}$ with $\alpha \in \mathbb{R}^n$ and $w \in \mathbb{R}^d$. Then $DF(x)$ is written as

$$DF(x)^\top = \begin{bmatrix} \frac{1}{\lambda n}\mathbf{A} & -\mathbf{I}_d \\ \mathbf{I}_n & \nabla\Phi(w)^\top \end{bmatrix}, \tag{A.34}$$

where $\nabla\Phi(w)^\top = \mathbf{Diag}\left(\phi_1''(a_1^\top w), \dots, \phi_n''(a_n^\top w)\right)\mathbf{A}^\top \in \mathbb{R}^{n\times d}$. Denote the diagonal matrix $D(w) \overset{\text{def}}{=} \mathbf{Diag}\left(\phi_1''(a_1^\top w), \dots, \phi_n''(a_n^\top w)\right)$. Since $\phi_i$ is continuously twice differentiable and convex, $\phi_i''(a_i^\top w) \geq 0$ for all $i$. Thus, $D(w) \geq 0$.

Let $(u; v) \in \mathbb{R}^{n+d}$ with $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^d$ such that $DF(x)^\top [u; v] = 0$. We have

$$DF(x)^\top \begin{bmatrix} u \\ v \end{bmatrix} = 0 \overset{(A.34)}{\iff} \begin{bmatrix} \frac{1}{\lambda n}\mathbf{A} & -\mathbf{I}_d \\ \mathbf{I}_n & \nabla\Phi(w)^\top \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = 0 \Longrightarrow \left(\mathbf{I}_n + \frac{1}{\lambda n} D(w)\mathbf{A}^\top\mathbf{A}\right) u = 0. \quad \text{(A.35)}$$

If $D(w)$ is invertible, (A.35) becomes

$$D(w)\left(D(w)^{-1} + \frac{1}{\lambda n}\mathbf{A}^\top\mathbf{A}\right) u = 0 \iff \left(D(w)^{-1} + \frac{1}{\lambda n}\mathbf{A}^\top\mathbf{A}\right) u = 0. \quad \text{(A.36)}$$

Since $D(w)$ is invertible, i.e. $D(w) > 0$, we obtain $D(w)^{-1} > 0$. As $\frac{1}{\lambda n}\mathbf{A}^\top\mathbf{A} \geq 0$, we get $D(w)^{-1} + \frac{1}{\lambda n}\mathbf{A}^\top\mathbf{A} > 0$, thus invertible. From (A.36), we get $u = 0$.

Otherwise, $D(w)$ is not invertible. Without losing generality, we assume that $\phi_1''(a_1^\top w) \geq \phi_2''(a_2^\top w) \geq \cdots \geq \phi_n''(a_n^\top w) = 0$. Let $j$ be the largest index for which $\phi_j''(a_j^\top w) > 0$. If $j$ does not exist, then $D(w) = 0$. From (A.35), we get $u = 0$ directly. If $j$ exists, we have $1 \leq j < n$ and

$$\begin{aligned} D(w)\mathbf{A}^\top\mathbf{A} &= \mathbf{Diag}\left(\phi_1''(a_1^\top w), \dots, \phi_j''(a_j^\top w), 0, \dots, 0\right)\mathbf{A}^\top\mathbf{A} \\ &= \begin{bmatrix} \mathbf{Diag}\left(\phi_1''(a_1^\top w), \dots, \phi_j''(a_j^\top w)\right)\mathbf{A}_{1:j}^\top\mathbf{A}_{1:j} & 0 \\ 0 & 0 \end{bmatrix}, \end{aligned} \quad \text{(A.37)}$$

where $\mathbf{A}_{1:j} \overset{\text{def}}{=} [a_1 \ \cdots \ a_j] \in \mathbb{R}^{d\times j}$. Note $u = [u_1; \cdots; u_n] \in \mathbb{R}^n$. Plugging (A.37) into (A.35), we get

$$\begin{aligned} &\left(\mathbf{I}_n + \frac{1}{\lambda n}\begin{bmatrix} \mathbf{Diag}\left(\phi_1''(a_1^\top w), \dots, \phi_j''(a_j^\top w)\right)\mathbf{A}_{1:j}^\top\mathbf{A}_{1:j} & 0 \\ 0 & 0 \end{bmatrix}\right) u = 0 \\ \iff &\begin{cases} \left(\mathbf{I}_j + \frac{1}{\lambda n}\mathbf{Diag}\left(\phi_1''(a_1^\top w), \dots, \phi_j''(a_j^\top w)\right)\mathbf{A}_{1:j}^\top\mathbf{A}_{1:j}\right) u_{1:j} = 0 \\ u_{(j+1):n} = 0 \end{cases}, \end{aligned} \quad \text{(A.38)}$$

where $u_{1:j} \overset{\text{def}}{=} [u_1; \cdots; u_j] \in \mathbb{R}^j$ and $u_{(j+1):n} \overset{\text{def}}{=} [u_{j+1}; \cdots; u_n] \in \mathbb{R}^{n-j}$. From (A.38), $u_{(j+1):n} = 0$. Now $\mathbf{Diag}\left(\phi_1''(a_1^\top w), \dots, \phi_j''(a_j^\top w)\right)$ is invertible in the subspace $\mathbb{R}^j$ as every coordinate in the diagonal $\phi_i''(a_i^\top w)$ is strictly positive for all $1 \leq i \leq j$. Similarly, we obtain $u_{1:j} = 0$ from the first equation of (A.38). Overall we get $u = 0$.

Thus, in all cases, $u = 0$, then $v = \frac{1}{\lambda n}Au = 0$. We can thus induce that $DF(\alpha; w)^\top$ is invertible for all $\alpha$ and $w$. Similar to Lemma 2.19, we have (2.24) hold, and by Lemma 2.9, we have that (2.22) holds. □

## A.6 Sufficient conditions for reformulation assumption (2.10)

To give sufficient conditions for (2.10) to hold, we need to study the spectra of $\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x)\right]$ . The expected matrix $\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x)\right]$ has made an appearance in several references (Gower et al., 2019a; Mutny et al., 2020; Derezinski et al., 2020) in different contexts and with different sketches. We build upon some of these past results and adapt them to our setting.

First note that (2.10) holds if $\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x)\right]$ is invertible. The invertibility of $\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x)\right]$ was already studied in detail in the linear setting in Theorem 3 in Gower and Richtárik (2015a) when $\mathbf{S}$ is sampled from a discrete distribution. Here we can state a sufficient condition of (2.10) for sketching matrices that have a continuous distribution.

**Lemma A.8.** *For every $x \in \mathbb{R}^p$, if $\mathbb{E}_{\mathbf{S}\sim\mathcal{D}_x}\left[\mathbf{S}\mathbf{S}^\top\right]$ and $DF(x)^\top DF(x)$ are invertible, then $\mathbb{E}_{\mathbf{S}\sim\mathcal{D}_x}\left[\mathbf{H}_{\mathbf{S}}(x)\right]$ is invertible.*

*Proof.* Let $x \in \mathbb{R}^p$ and $\mathbf{S} \sim \mathcal{D}_x$. Let $\mathbf{G} = DF(x)^\top DF(x)$ which is thus symmetric positive definite and $\mathbf{W} = \mathbf{S}^\top$. In this case, since $\mathbf{G}$ is invertible we have that $\mathbf{Ker}(\mathbf{G}) = \{0\} \subset \mathbf{Ker}(\mathbf{W})$ verified, by Lemma A.3, we have that

$$\mathbf{Ker}\left(\left(\mathbf{S}^\top DF(x)^\top DF(x)\mathbf{S}\right)^\dagger\right) = \mathbf{Ker}\left(\mathbf{S}^\top DF(x)^\top DF(x)\mathbf{S}\right) = \mathbf{Ker}(\mathbf{S}), \tag{A.39}$$

Consequently, using Lemma A.3 again with $\mathbf{G} = \left(\mathbf{S}^\top DF(x)^\top DF(x)\mathbf{S}\right)^\dagger$, $\mathbf{W} = \mathbf{S}$ and $\mathbf{Ker}(\mathbf{G}) \subset \mathbf{Ker}(\mathbf{W})$ given by (A.39), we have that

$$\mathbf{Ker}(\mathbf{H}_{\mathbf{S}}(x)) = \mathbf{Ker}\left(\mathbf{S}\left(\mathbf{S}^\top DF(x)^\top DF(x)\mathbf{S}\right)^\dagger \mathbf{S}^\top\right) = \mathbf{Ker}(\mathbf{S}^\top) = \mathbf{Ker}(\mathbf{S}\mathbf{S}^\top). \tag{A.40}$$

Following the same steps in the proof of Lemma 2.9 right after (A.17), we obtain

$$\mathbf{Ker}(\mathbb{E}\left[\mathbf{H}_{\mathbf{S}}(x)\right]) = \bigcap_{\mathbf{S}\sim\mathcal{D}_x} \mathbf{Ker}(\mathbf{H}_{\mathbf{S}}(x)) \overset{(A.40)}{=} \bigcap_{\mathbf{S}\sim\mathcal{D}_x} \mathbf{Ker}(\mathbf{S}\mathbf{S}^\top) = \mathbf{Ker}(\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right]) = \{0\},$$

where the last equality follows as $\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right]$ is invertible, which concludes the proof. □

The invertibility of $\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right]$ states that the sketching matrices need to "span every dimension of the space" in expectation. This is the case for Gaussian and subsampling sketches which are shown in Lemma 2.11. This is also the case for our applications SNM and TCS which are shown in the proofs of Lemma 2.19 and Lemma 2.22, respectively.

As for the invertibility of $DF(x)^\top DF(x) \in \mathbb{R}^{m\times m}$, this imposes that $DF(x)$ has full-column rank for all $x \in \mathbb{R}^p$, thus $m \leq p$. This excludes the regime of solving $F(x) = 0$ with $m > p$.

However, our applications SNM and TCS also satisfy this condition which are again shown in the proofs of Lemma A.5 and Lemma 2.22, respectively.

Consequently, by Lemma A.8, we have that SNM and TCS satisfy (2.10).

## A.7 Extension of SNR and Randomized Subspace Newton

In the SNR method in line 4 in Algorithm 1, we only consider a projection under the standard Euclidean norm. If we allow SNR for a changing norm that depends on the iterates, we find that the Randomized Subspace Newton (Gower et al., 2019a) (RSN) method is in fact a special case of SNR under this extension.

The changing norm projection of SNR is that, at $k$th iteration of SNR, instead of applying line 4 in Algorithm 1, we can apply the following update

$$x^{k+1} \;=\; x^k - \gamma \mathbf{W}_k^{-1} DF(x^k)\mathbf{S}_k \left(\mathbf{S}_k^\top DF(x^k)^\top \mathbf{W}_k^{-1} DF(x^k)\mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top F(x^k) \tag{A.41}$$

where $\mathbf{W}_k \equiv \mathbf{W}(x^k)$ with $\mathbf{W}(x^k)$ a certain symmetric positive-definite matrix associated with the $k$th iterate $x^k \in \mathbb{R}^p$.

The interpretation of using the matrix $\mathbf{W}(x^k)$ is that, assuming Assumption 3.5 holds, then instead of considering (2.7), we apply the following updates

$$\begin{aligned} x^{k+1} = \operatorname{argmin}_{x\in\mathbb{R}^p} & \left\|x - x^k\right\|_{\mathbf{W}_k}^2 \\ \text{s. t.} \quad & \mathbf{S}_k^\top DF(x^k)^\top (x - x^k) \;=\; -\gamma \mathbf{S}_k^\top F(x^k), \end{aligned} \tag{A.42}$$

using the projection $\|\cdot\|_{\mathbf{W}_k}$ which changes at each iteration. One can verify easily that (A.42) is equivalent to (A.41) under Assumption 3.5, even though this assumption is not necessary and the update (A.41) is still available. Besides, the update (A.41) is known as sketched Newton-Raphson with variable metric (SNRVM), studied in our later work Chen et al. (2022a). See Section 3.4 in Chapter 3 for more details.

Now we can show that RSN is a special case of SNR with a changing norm projection. The RSN method (Gower et al., 2019a) is a stochastic second order method that takes a Newton-type step at each iteration to solve the minimization problem

$$\min_{x\in\mathbb{R}^p} P(x)$$

where $P : \mathbb{R}^p \to \mathbb{R}$ is a twice differentiable and convex function. In brevity, the updates in RSN at the $k$th iteration are given by

$$x^{k+1} \;=\; x^k - \frac{1}{\hat{L}}\mathbf{S}_k \left(\mathbf{S}_k^\top \nabla^2 P(x^k)\mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top \nabla P(x^k) \tag{A.43}$$

where $\mathbf{S}_k$ is sampled i.i.d from a fixed distribution $\mathcal{D}$ and $\hat{L} > 0$ is the *relative smoothness* constant (Gower et al., 2019a).

Since $P(x)$ is convex, it suffices to find a stationary point $x$ such that $\nabla P(x) = 0$. We can recover the exact same update (A.43) by applying SNR to solve $\nabla P(x) = 0$ with an adaptive changing norm. That is, let $F(x) = \nabla P(x)$ and $DF(x) = \nabla^2 P(x)$. At the $k$th iteration, let $\mathbf{W}_k = \nabla^2 P(x^k)$. Then (A.41) is exactly the RSN update (A.43) with $\gamma = \frac{1}{\hat{L}}$.

## A.8 Explicit formulation of the TCS method

Here we provide details about how TCS method presented in Section 2.8 is obtained from the general SNR method 1.

Consider the SNR method (line 4 in Algorithm 1) applied for the nonlinear equations $F(\alpha; w) = 0$ with $F$ defined in (2.49) and the Jacobian of $F(\alpha; w)$ in (A.34).

At $k$th iteration $(\alpha^k, w^k) \in \mathbb{R}^n \times \mathbb{R}^d$, let

$$\begin{bmatrix} \alpha^{k+1} \\ w^{k+1} \end{bmatrix} \overset{\text{def}}{=} \begin{bmatrix} \alpha^k \\ w^k \end{bmatrix} + \gamma \cdot \begin{bmatrix} \Delta\alpha^k \\ \Delta w^k \end{bmatrix}$$

and $\mathbf{S}_k \in \mathbb{R}^{(d+n)\times(\tau_d+\tau_n)}$ the random sketching matrix. By line 4 in Algorithm 1, we obtain the closed form update

$$\begin{bmatrix} \Delta\alpha^k \\ \Delta w^k \end{bmatrix} \;=\; -DF(\alpha^k; w^k)\mathbf{S}_k \left(\mathbf{S}_k^\top DF(\alpha^k; w^k)^\top DF(\alpha^k; w^k)\mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top \begin{bmatrix} \frac{1}{\lambda n}\mathbf{A}\alpha^k - w^k \\ \alpha^k + \Phi(w^k) \end{bmatrix}. \tag{A.44}$$

As for the tossing-coin-sketch, consider a Bernoulli parameter $b$ with $b \in (0, 1)$. There is a probability $1 - b$ that the random sketching matrix has the type $\mathbf{S}_k = \begin{bmatrix} \mathbf{S}_d & 0 \\ 0 & 0 \end{bmatrix}$ with $\mathbf{S}_d \in \mathbb{R}^{d\times\tau_d}$, a $(d, \tau_d)$–block sketch, and a probability $b$ that the random sketching matrix has the type $\mathbf{S}_k = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{S}_n \end{bmatrix}$ with $\mathbf{S}_n \in \mathbb{R}^{n\times\tau_n}$, a $(n, \tau_n)$–block sketch. So $\mathbf{S}_d = \mathbf{I}_{B_d}$ and $\mathbf{S}_n = \mathbf{I}_{B_n}$.

Let $\mathbf{A}_{B_d,:} \equiv \mathbf{I}_{B_d}^\top \mathbf{A} \in \mathbb{R}^{\tau_d\times n}$ denote a row subsampling of $\mathbf{A}$ and $\mathbf{A}_{:,B_n} \equiv \mathbf{A}\mathbf{I}_{B_n} \in \mathbb{R}^{d\times\tau_n}$ denote a column subsampling of $\mathbf{A}$. Let $\nabla\Phi^k_{B_n} \equiv \nabla\Phi^k \mathbf{I}_{B_n} \in \mathbb{R}^{d\times\tau_n}$ denote a column subsampling

of $\nabla\Phi^k$ with $\nabla\Phi^k \equiv \nabla\Phi(w^k)$ and $\Phi^k \equiv \Phi(w^k)$. We also use the shorthands $v_{B_n} \equiv \mathbf{I}_{B_n}^\top v \in \mathbb{R}^{\tau_n}$ with $v \in \mathbb{R}^n$ and $v_{B_d} \equiv \mathbf{I}_{B_d}^\top v \in \mathbb{R}^{\tau_d}$ with $v \in \mathbb{R}^d$.

If $\mathbf{S}_k = \begin{bmatrix} \mathbf{S}_d & 0 \\ 0 & 0 \end{bmatrix}$, the update (A.44) applied for the function (2.49) and its Jacobian (A.34) becomes

$$\begin{aligned} \begin{bmatrix} \Delta\alpha^k \\ \Delta w^k \end{bmatrix} &= -\begin{bmatrix} \frac{1}{\lambda n}\mathbf{A}^\top \mathbf{S}_d \\ -\mathbf{S}_d \end{bmatrix} \left(\mathbf{S}_d^\top \left(\frac{1}{\lambda^2 n^2}\mathbf{A}\mathbf{A}^\top + \mathbf{I}_d\right)\mathbf{S}_d\right)^\dagger \mathbf{S}_d^\top \left(\frac{1}{\lambda n}\mathbf{A}\alpha^k - w^k\right) \\ &= -\begin{bmatrix} \frac{1}{\lambda n}\mathbf{A}_{B_d,:}^\top \\ -\mathbf{I}_{B_d} \end{bmatrix} \left(\frac{\mathbf{A}_{B_d,:}\mathbf{A}_{B_d,:}^\top}{\lambda^2 n^2} + \mathbf{I}_{\tau_d}\right)^\dagger \left(\frac{\mathbf{A}_{B_d,:}\alpha^k}{\lambda n} - w_{B_d}^k\right). \end{aligned} \tag{A.45}$$

Similarly, if $\mathbf{S}_k = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{S}_n \end{bmatrix}$, the update (A.44) becomes

$$\begin{aligned} \begin{bmatrix} \Delta\alpha^k \\ \Delta w^k \end{bmatrix} &= -\begin{bmatrix} \mathbf{S}_n \\ \nabla\Phi^k \mathbf{S}_n \end{bmatrix} \left(\mathbf{S}_n^\top \left([\nabla\Phi^k]^\top \nabla\Phi^k + \mathbf{I}_n\right)\mathbf{S}_n\right)^\dagger \mathbf{S}_n^\top \left(\alpha^k + \Phi^k\right) \\ &= -\begin{bmatrix} \mathbf{I}_{B_n} \\ \nabla\Phi_{B_n}^k \end{bmatrix} \left([\nabla\Phi_{B_n}^k]^\top \nabla\Phi_{B_n}^k + \mathbf{I}_{\tau_n}\right)^\dagger \left(\alpha_{B_n}^k + \Phi_{B_n}^k\right). \end{aligned} \tag{A.46}$$

Then we update $\begin{bmatrix} \alpha^{k+1} \\ w^{k+1} \end{bmatrix} = \begin{bmatrix} \alpha^k \\ w^k \end{bmatrix} + \gamma \cdot \begin{bmatrix} \Delta\alpha^k \\ \Delta w^k \end{bmatrix}$.

See Algorithm 5 the pseudocode for the updates (A.45) and (A.46).

## A.9 Pseudo code and implementation details for GLMs

We also provide a more efficient and detailed implementation of Algorithm 5 in Algorithm 7 in this section.

It is beneficial to first understand Algorithm 5 in the simple setting where $\tau_d = \tau_n = 1$. We refer to this setting as the Kaczmarz–TCS method.

**Algorithm 5:** $\tau$–TCS

**Input:** Stepsize $\gamma > 0$, sketch sizes $\tau_d, \tau_n \in \mathbb{N}$, probability $b \in (0,1)$, $v \sim B(b)$ be a Bernoulli random variable (the coin toss)

Initialize $(\alpha^0; w^0) \in \mathbb{R}^{n+d}$

**for** $k = 0, 1, \cdots$ **do**

  Sample $v \sim B(b)$ with $v \in \{0, 1\}$

  **if** $v = 0$ **then**

    Sample $B_d \subset \{1, \ldots, d\}$ with $|B_d| = \tau_d$ uniformly

    Compute $y_d \in \mathbb{R}^{\tau_d}$ the least norm solution to

    $$\left(\frac{\mathbf{A}_{B_d,:}\mathbf{A}_{B_d,:}^\top}{\lambda^2 n^2} + \mathbf{I}_{\tau_d}\right) y_d = \frac{\mathbf{A}_{B_d,:}\alpha^k}{\lambda n} - w^k_{B_d}$$

    Compute the updates

    $$\begin{bmatrix}\Delta\alpha^k \\ \Delta w^k\end{bmatrix} = -\begin{bmatrix}\frac{1}{\lambda n}\mathbf{A}_{B_d,:}^\top \\ -\mathbf{I}_{B_d}\end{bmatrix} y_d$$

  **else**

    Sample $B_n \subset \{1, \ldots, n\}$ with $|B_n| = \tau_n$ uniformly

    Compute $y_n \in \mathbb{R}^{\tau_n}$ the least norm solution to

    $$\left([\nabla\Phi^k_{B_n}]^\top \nabla\Phi^k_{B_n} + \mathbf{I}_{\tau_n}\right) y_n = \alpha^k_{B_n} + \Phi^k_{B_n}$$

    Compute the updates

    $$\begin{bmatrix}\Delta\alpha^k \\ \Delta w^k\end{bmatrix} = -\begin{bmatrix}\mathbf{I}_{B_n} \\ \nabla\Phi^k_{B_n}\end{bmatrix} y_n$$

  $w^{k+1} = w^k + \gamma\Delta w^k$

  $\alpha^{k+1} = \alpha^k + \gamma\Delta\alpha^k$

**Output:** Last iterate $\alpha^k$, $w^k$

### A.9.1 Kaczmarz–TCS

Let $f_j \in \mathbb{R}^d$ ($e_i \in \mathbb{R}^n$) be the $j$th (the $i$th) unit coordinate vector in $\mathbb{R}^d$ (in $\mathbb{R}^n$, respectively). For $\mathbf{S}_k = \begin{bmatrix} \mathbf{S}_d & 0 \\ 0 & 0 \end{bmatrix}$ with $\mathbf{S}_d = f_j$, from (A.45) we get

$$\begin{bmatrix} \Delta\alpha^k \\ \Delta w^k \end{bmatrix} = -\frac{\frac{1}{\lambda n}\sum_{l=1}^n a_{lj}\alpha_l^k - w_j^k}{\frac{1}{\lambda^2 n^2}\sum_{l=1}^n a_{lj}^2 + 1} \begin{bmatrix} \frac{1}{\lambda n}\begin{pmatrix} a_{1j} \\ \vdots \\ a_{nj} \end{pmatrix} \\ -f_j \end{bmatrix}. \tag{A.47}$$

For $\mathbf{S}_k = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{S}_n \end{bmatrix}$ with $\mathbf{S}_n = e_i$, from (A.46) we get

$$\begin{bmatrix} \Delta\alpha^k \\ \Delta w^k \end{bmatrix} = -\frac{\alpha_i^k + \phi_i'(a_i^\top w^k)}{\|a_i\|_2^2 \phi_i''(a_i^\top w^k)^2 + 1} \begin{bmatrix} e_i \\ \phi_i''(a_i^\top w^k) a_i \end{bmatrix}. \tag{A.48}$$

See Algorithm 6 an efficient implementation of Algorithm 5 in a single row sampling case. Notice that we introduce an auxiliary variable $\overline{\alpha}^k$ to update the term $\frac{1}{\lambda n}\sum_{l=1}^n a_{lj}\alpha_l^k$ for $j = 1, \cdots, d$ in (A.47) and we store a $d \times d$ matrix `cov` which can be seen as the covariance matrix of the dataset $\mathbf{A}$ to update the term $\frac{1}{\lambda^2 n^2}\sum_{l=1}^n a_{lj}^2$ for $j = 1, \cdots, d$ in (A.47) (see Algorithm 6 Line 10). We also store a vector `sample` $\in \mathbb{R}^n$ to update the term $\|a_i\|_2^2$ for $i = 1, \cdots, n$ in (A.48) (see Algorithm 6 Line 19).

**Cost per iteration analysis of Algorithm 6** From Algorithm 6, the cost of computing (A.47) is $\mathcal{O}(n)$ with $n$ coordinates' updates of the auxiliary variable $\alpha$ (see Algorithm 6 Line 13). This is affordable as the cost of each coordinate's update is 1. Besides $\mathbf{A}$ is often sparse. The update in this case can be much cheaper than $n$. Besides, the cost of computing (A.48) is $\mathcal{O}(d)$. If we choose the Bernoulli parameter $b = n/(n+d)$ which selects one row of $F$ uniformly, the total cost of the updates TCS in expectation with respect to the Bernoulli distribution will be

$$\text{Cost(update TCS)} = \mathcal{O}(n) * (1-b) + \mathcal{O}(d) * b = \mathcal{O}(nd/(n+d)) = \mathcal{O}(\min(n, d)).$$

So the TCS method can have the same cost per iteration as the stochastic first-order methods in the case $d < n$, such as SVRG (Johnson and Zhang, 2013), SAG (Schmidt et al., 2017), dfSDCA (Shalev-Shwartz, 2016) and Quartz (Qu et al., 2015).

**Algorithm 6:** Kaczmarz-TCS

**Input:** $\mathcal{D}$ = distribution over random matrices

**store in memory:**

  $\texttt{sample}$: $(\|a_i\|_2^2)_{1\leq i\leq n} \in \mathbb{R}^n$

  $\texttt{cov}$: $\frac{1}{\lambda^2 n^2}\mathbf{A}\mathbf{A}^\top \in \mathbb{R}^{d\times d}$

**initialization:**

  Choose $(\alpha^0, w^0) \in \mathbb{R}^n \times \mathbb{R}^d$ and a step size $\gamma \in \mathbb{R}^{++}$

  Set $\overline{\alpha}^0 = \frac{1}{\lambda n}\mathbf{A}\alpha^0$

**for** $k = 0, 1, \cdots$ **do**

  Sample a fresh tossing-coin sketching matrix: $\mathbf{S}_k \sim \mathcal{D}$

  **if** $\mathbf{S}_k = \begin{bmatrix} \mathbf{S}_d & 0 \\ 0 & 0 \end{bmatrix}$ *with* $\mathbf{S}_d = f_j$ **then**

    **Update** (A.47): \\ *Sketch a linear system based on the first $d$ rows of the Jacobian*

      $\Delta w_j^k = \frac{\overline{\alpha}_j^k - w_j^k}{\texttt{cov}[j,\, j]+1}$

      $\Delta\alpha^k = -\Delta w_j^k \cdot \frac{1}{\lambda n}\begin{bmatrix} a_{1j} \\ \vdots \\ a_{nj} \end{bmatrix}$

      $w_j^{k+1} = w_j^k + \gamma \cdot \Delta w_j^k$ \\ *$j$th coordinate's update of the variable $w^k$*

      $\alpha^{k+1} = \alpha^k + \gamma \cdot \Delta\alpha^k$ \\ *full vector's update of the auxiliary variable $\alpha^k$*

      $\overline{\alpha}^{k+1} = \overline{\alpha}^k - \gamma \cdot \Delta w_j^k \cdot \texttt{cov}[:,\, j]$

  **else**

    $\mathbf{S}_k = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{S}_n \end{bmatrix}$ with $\mathbf{S}_n = e_i$

    **Update** (A.48): \\ *Sketch a system based on the last $n$ rows of the Jacobian*

      $\texttt{temp} = a_i^\top w^k$ \\ *temporal scalar*

      $\Delta\alpha_i^k = -\frac{\alpha_i^k + \phi_i'(\texttt{temp})}{\texttt{sample}[i]\cdot\phi_i''(\texttt{temp})^2+1}$

      $\Delta w^k = \Delta\alpha_i^k \cdot \phi_i''(\texttt{temp}) \cdot a_i$

      $w^{k+1} = w^k + \gamma \cdot \Delta w^k$ \\ *full vector's update of the variable $w^k$*

      $\alpha_i^{k+1} = \alpha_i^k + \gamma \cdot \Delta\alpha_i^k$ \\ *$i$th coordinate's update of the auxiliary variable $\alpha^k$*

      $\overline{\alpha}^{k+1} = \overline{\alpha}^k + \gamma \cdot \Delta\alpha_i^k \cdot \frac{1}{\lambda n} a_i$

**Output:** Last iterate $\alpha^k$, $w^k$

### A.9.2 $\tau$–Block TCS

Here we provide Algorithm 7 which is a detailed implementation of Algorithm 5 in a more efficient way. Similar to Algorithm 6 but with sketch sizes $\tau_d$ and $\tau_n$, we also store a $d \times d$ matrix `cov`, but not a vector `sample`. We refer to Algorithm 7 as the $\tau$-block TCS method.

From Algorithm 7, the cost of solving the $\tau_d \times \tau_d$ system (see Algorithm 7 Line 10) is $\mathcal{O}(\tau_d^3)$ for a direct solver and the cost of updating $\alpha$ and $\overline{\alpha}$ (see Algorithm 7 Line 13 and Line 14) are $\mathcal{O}(\tau_d n)$ and $\mathcal{O}(\tau_d d)$ respectively. Overall, this implies that the cost of executing the sketching of the first $d$ rows is

$$c_d \overset{\text{def}}{=} \mathcal{O}(\max(\tau_d^3, \tau_d n, \tau_d d)). \tag{A.49}$$

Similarly, the dominant cost of executing the last $n$ rows sketch comes from forming the $\tau_n \times \tau_n$ linear system or solving such system (see Line 25), which gives

$$c_n \overset{\text{def}}{=} \mathcal{O}(\max(\tau_n^3, \tau_n^2 d)). \tag{A.50}$$

In average, which means taking the Bernoulli parameter $b$ into account, the total cost per iteration of the TCS updates in expectation is

$$\begin{aligned} c_{avg} &\overset{\text{def}}{=} c_d \times (1-b) + c_n \times b \\ &= \mathcal{O}(\max(\tau_d^3, \tau_d n, \tau_d d)) \times (1-b) + \mathcal{O}(\max(\tau_n^3, \tau_n^2 d)) \times b. \end{aligned} \tag{A.51}$$

Depending on the sketch sizes $(\tau_d, \tau_n)$ and the Bernoulli parameter $b$, the nature of $c_{avg}$ can be different from $\mathcal{O}(d)$ (see Kaczmarz-TCS in Algorithm 6) to $\mathcal{O}(d^2)$ (see the cost per iteration analysis paragraph in the next section). We discuss the total cost per iteration of the TCS method in practice in different cases in the next section.

## A.10 Additional experimental details

All the sampling of the methods was pre-computed before starting counting the wall-clock time for each method and each dataset. We also paused the timing when the algorithms were under process of the performance evaluation of the gradient norm or of the logistic regression loss that were necessary to generate the plots.

In the following, from the experimental results for GLM in Section 2.8.1, we discuss the parameters' choices for TCS in practice, including the sketch sizes $(\tau_d, \tau_n)$, the Bernoulli parameter $b$, the stepsize $\gamma$ and the analysis of total cost per iteration. See Table A.1 for the parameters we chose for TCS in the experiments in Figure 2.1. Such choices are due to TCS's cost per iteration.

**Algorithm 7:** $\tau$–Block TCS

**Input:** $\mathcal{D}$ = distribution over random matrices

**store in memory:**

  $\texttt{cov}$: $\frac{1}{\lambda^2 n^2}\mathbf{A}\mathbf{A}^\top \in \mathbb{R}^{d\times d}$

**initialization:**

  Choose $(\alpha^0, w^0) \in \mathbb{R}^n \times \mathbb{R}^d$ and a step size $\gamma \in \mathbb{R}^{++}$

  Set $\overline{\alpha}^0 = \frac{1}{\lambda n}\mathbf{A}\alpha^0$

**for** $k = 0, 1, \cdots$ **do**

  Sample a fresh tossing-coin sketching matrix: $\mathbf{S}_k \sim \mathcal{D}$

  **if** $\mathbf{S}_k = \begin{bmatrix} \mathbf{S}_d & 0 \\ 0 & 0 \end{bmatrix}$ *with* $\mathbf{S}_d = \mathbf{I}_{B_d}$ *and* $|B_d| = \tau_d$ **then**

    **Update** (A.45): \\ *Sketch a linear system based on the first $d$ rows of the Jacobian*

      Compute $y_d \in \mathbb{R}^{\tau_d}$ the least norm solution to the $\tau_d \times \tau_d$ linear system

      $\left(\texttt{cov}[B_d,\ B_d] + \mathbf{I}_{\tau_d}\right) y_d = -\left(\overline{\alpha}^k_{B_d} - w^k_{B_d}\right)$

      Compute the updates

      $w^{k+1}_{B_d} = w^k_{B_d} - \gamma \cdot y_d$ \\ *$\tau_d$ coordinates' update of the variable $w^k$*

      $\alpha^{k+1} = \alpha^k + \gamma \cdot \frac{1}{\lambda n}\mathbf{A}^\top_{B_d,:} y_d$ \\ *full vector's update of the auxiliary variable $\alpha^k$*

      $\overline{\alpha}^{k+1} = \overline{\alpha}^k + \gamma \cdot \texttt{cov}[:,\ B_d] y_d$

  **else**

    $\mathbf{S}_k = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{S}_n \end{bmatrix}$ with $\mathbf{S}_n = \mathbf{I}_{B_n}$ and $|B_n| = \tau_n$

    **Update** (A.46): \\ *Sketch a system based on the last $n$ rows of the Jacobian*

      $\texttt{temp} = \mathbf{A}^\top_{:,B_n} w^k \in \mathbb{R}^{\tau_n}$ \\ *Temporal vector*

      $\mathbf{D}^k_{B_n} = \mathbf{Diag}\left(\phi''_{B_n}(\texttt{temp})\right) \in \mathbb{R}^{\tau_n \times \tau_n}$ \\ *Compute $\phi''_i(a_i^\top w^k)$ element-wise $\forall i \in B_n$*

      $\nabla\Phi^k_{B_n} = \mathbf{A}_{:,B_n}\mathbf{D}^k_{B_n} \in \mathbb{R}^{d\times\tau_n}$

      $\Phi^k_{B_n} = \phi'_{B_n}(\texttt{temp})$ \\ *Compute $\phi'_i(a_i^\top w^k)$ element-wise $\forall i \in B_n$*

      Compute $y_n \in \mathbb{R}^{\tau_n}$ the least norm solution to the $\tau_n \times \tau_n$ linear system

      $\left([\nabla\Phi^k_{B_n}]^\top \nabla\Phi^k_{B_n} + \mathbf{I}_{\tau_n}\right) y_n = -\left(\alpha^k_{B_n} + \Phi^k_{B_n}\right)$

      Compute the updates

      $w^{k+1} = w^k + \gamma \cdot \nabla\Phi^k_{B_n} y_n$ \\ *full vector's update of the variable $w^k$*

      $\alpha^{k+1}_{B_n} = \alpha^k_{B_n} + \gamma \cdot y_n$ \\ *$\tau_n$ coordinates' update of the auxiliary variable $\alpha^k$*

      $\overline{\alpha}^{k+1} = \overline{\alpha}^k + \gamma \cdot \frac{1}{\lambda n}\mathbf{A}_{:,B_n} y_n$

**Output:** Last iterate $\alpha^k$, $w^k$

**Table A.1** – Details of the parameters' choices ($\gamma$ and $b$) for 50-TCS, 150-TCS and 300-TCS

| | | 50-TCS | 150-TCS | 300-TCS |
|---|---|---|---|---|
| dataset | stepsize | Bernoulli | Bernoulli | Bernoulli |
| covetype | 1.0 | $\frac{n}{n+\tau_n*3}$ | $\frac{n}{n+\tau_n*3}$ | $\frac{n}{n+\tau_n*3}$ |
| a9a | 1.5 | $\frac{n}{n+\tau_n} - 0.03$ | $\frac{n}{n+\tau_n} - 0.03$ | $\frac{n}{n+\tau_n} - 0.11$ |
| fourclass | 1.0 | $\frac{n}{n+\tau_n} - 0.11$ | $\frac{n}{n+\tau_n} - 0.11$ | $\frac{n}{n+\tau_n} - 0.11$ |
| artificial | 1.0 | $\frac{n}{n+\tau_n} - 0.03$ | $\frac{n}{n+\tau_n} - 0.11$ | $\frac{n}{n+\tau_n} - 0.11$ |
| ijcnn1 | 1.8 | $\frac{n}{n+\tau_n} - 0.03$ | $\frac{n}{n+\tau_n} - 0.11$ | $\frac{n}{n+\tau_n} - 0.11$ |
| webspam | 1.8 | $\frac{n}{n+\tau_n*3}$ | $\frac{n}{n+\tau_n*3}$ | $\frac{n}{n+\tau_n*3}$ |
| epsilon | 1.8 | $\frac{n}{n+\tau_n*3}$ | $\frac{n}{n+\tau_n*3}$ | $\frac{n}{n+\tau_n*3}$ |
| phishing | 1.8 | $\frac{n}{n+\tau_n} - 0.03$ | $\frac{n}{n+\tau_n} - 0.11$ | $\frac{n}{n+\tau_n} - 0.11$ |

**Choice of the sketch size $\tau_d$** For all of our experiments, $\tau_d = d$ performs always the best in time and in number of iterations. That means, when $\mathbf{S}_k = \begin{bmatrix} \mathbf{S}_d & 0 \\ 0 & 0 \end{bmatrix}$ at $k$th iteration, we choose $\mathbf{S}_d = \mathbf{I}_d$. Note also that the first $d$ rows (2.49) are linear, thus using $\mathbf{S}_d = \mathbf{I}_d$ gives an exact solution to these first $d$ equations. We found that such an exact solution from the linear part induces a fast convergence when $d < n$. We did not test datasets for which $d > n$ with $d$ very large.

**Choice of the Bernoulli parameter $b$ for uniform sampling** First, we calculate the probability of sampling one row of the function $F$ (2.49). Since there exists two types of sketching for TCS method depending on the *coin toss*, we address both of them. The probability of sampling one specific row of the first block (fist $d$ rows of (2.49)) is

$$p_d = \frac{\tau_d}{d} \times (1 - b)$$

and the one of the second block is

$$p_n = \frac{\tau_n}{n} \times b.$$

It is natural to choose $b$ such that the uniform sampling of the whole system, i.e. $p_d = p_n$, is guaranteed. This implies to set

$$p_{uniform} \overset{\text{def}}{=} \frac{\tau_d n}{\tau_d n + \tau_n d}.$$

As we choose $\tau_d = d$, this implies

$$p_{uniform} = \frac{n}{n + \tau_n}.$$

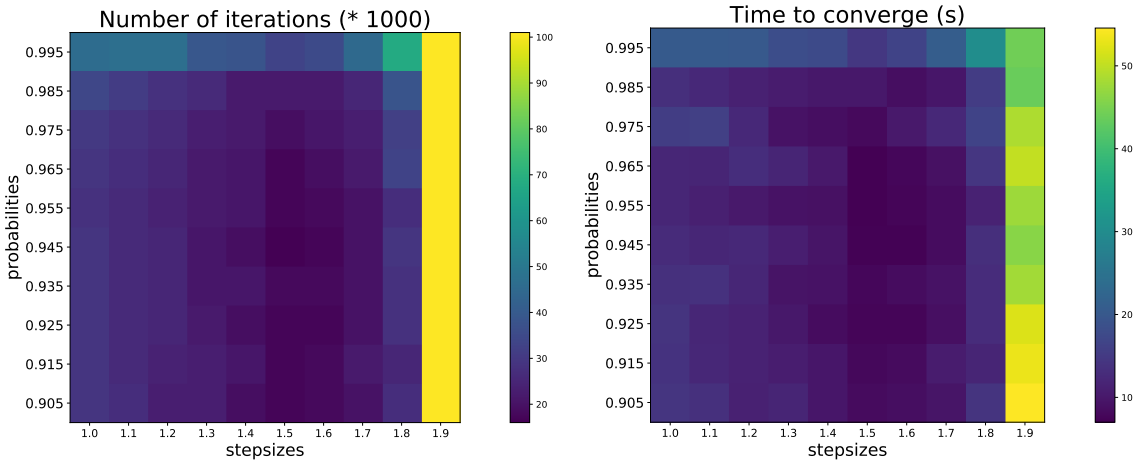


**Figure A.1** – a9a dataset: Grid search of the Bernoulli parameter $b$ and the stepsize $\gamma$ with 150-TCS method.The darker colors correspond to a resulting small gradient norm and thus a better solution.

However, we found through multiple experiments that when setting $b$ slightly smaller than $p_{uniform}$ (e.g. $-1\%$), this reduces significantly the number of iterations to get convergence. See in Figure A.1 for a grid search of the Bernoulli parameter $b$ and the stepsize $\gamma$ for a9a dataset with $b = p_{uniform} = 0.995$ in the first line of the figure. Before giving details about how to choose $b$ in practice, we first provide the cost per iteration analysis of the TCS method in detail.

**Total cost per iteration in expectation analysis for TCS in different cases** Recall two types of costs per iteration $c_d$ (A.49) and $c_n$ (A.50). In our cases, consider $b = \frac{n}{n+\tau_n}$ and $d = \tau_d < n$. To summarize, the cost per iteration in expectation can be one of the three following cases followed with their bounds:

1. If $\tau_n < \sqrt{n} < d < n$ such as *epsilon* dataset, then $c_d = \mathcal{O}(d^3) > c_n = \mathcal{O}(\tau_n^2 d)$, and

$$\begin{aligned} c_{avg1} &= \mathcal{O}(d^3) \times \left(1 - \frac{n}{n+\tau_n}\right) + \mathcal{O}(\tau_n^2 d) \times \frac{n}{n+\tau_n} = \mathcal{O}(\frac{\tau_n d}{n+\tau_n}(d^2 + \tau_n n)) \\ \implies \mathcal{O}(\tau_n^2 d) &\leq c_{avg1} \leq \mathcal{O}(\tau_n d^2); \end{aligned} \tag{A.52}$$

2. if $\tau_n < d < \sqrt{n}$ such as *webspam* dataset with 50-TCS and 150-TCS, *a9a*, *phishing* and *covtype* datasets with 50-TCS method, then $c_d = \mathcal{O}(dn) > c_n = \mathcal{O}(\tau_n^2 d)$, and

$$\begin{aligned} c_{avg2} &= \mathcal{O}(dn) \times \left(1 - \frac{n}{n+\tau_n}\right) + \mathcal{O}(\tau_n^2 d) \times \frac{n}{n+\tau_n} \\ &= \mathcal{O}(\frac{\tau_n n}{n+\tau_n}(d + \tau_n d)) = \mathcal{O}(\tau_n^2 d) \quad \text{as } \frac{1}{2} \leq \frac{n}{n+\tau_n} < 1; \end{aligned} \tag{A.53}$$

**Table A.2** – Cost per iteration for different datasets and different sketch sizes.

| dataset | 50-TCS | 150-TCS | 300-TCS |
|---|---|---|---|
| covetype | $c_d > c_n$ | $c_d > c_n$ | $c_d > c_n$ |
| a9a | $c_d > c_n$ | $c_d > c_n$ | $c_d < c_n$ |
| fourclass | $c_d < c_n$ | $c_d < c_n$ | $c_d < c_n$ |
| artificial | $c_d > c_n$ | $c_d < c_n$ | $c_d < c_n$ |
| ijcnn1 | $c_d > c_n$ | $c_d < c_n$ | $c_d < c_n$ |
| webspam | $c_d > c_n$ | $c_d > c_n$ | $c_d > c_n$ |
| epsilon | $c_d > c_n$ | $c_d > c_n$ | $c_d > c_n$ |
| phishing | $c_d > c_n$ | $c_d < c_n$ | $c_d < c_n$ |

3. if $d < \sqrt{n}$ and $d < \tau_n$ such as all the other experiments for TCS methods in Figure 2.1, then $c_d = \mathcal{O}(dn)$, $c_n = \mathcal{O}(\tau_n^3)$, and

$$\begin{aligned} c_{avg3} &= \mathcal{O}(dn) \times \left(1 - \frac{n}{n+\tau_n}\right) + \mathcal{O}(\tau_n^3) \times \frac{n}{n+\tau_n} \\ &= \mathcal{O}(\frac{\tau_n n}{n+\tau_n}(d+\tau_n^2)) \\ &= \mathcal{O}(\tau_n^3) > \mathcal{O}(d^3) \quad \text{as } \frac{1}{2} \leq \frac{n}{n+\tau_n} < 1. \end{aligned} \tag{A.54}$$

Notice that $c_{avg1}, c_{avg2}, c_{avg3} \ll \mathcal{O}(dn)$ in general for large scale datasets with large $n$. For example, $c_{avg1} < \mathcal{O}(dn)$ when $\tau_n d < n$. This justifies that TCS method is cheaper than the first-order method which requires evaluating the full gradient and thus has a cost per iteration of at least $\mathcal{O}(dn)$. From $c_{avg1}$ (A.52), we know that TCS method can have the same cost per iteration as the stochastic first-order methods which is $\mathcal{O}(d)$ in practice, such as SVRG (Johnson and Zhang, 2013), SAG (Schmidt et al., 2017), dfSDCA (Shalev-Shwartz, 2016) and Quartz (Qu et al., 2015).

Furthermore, from the above analysis of computational cost, we can easily obtain the comparisons between $c_d$ and $c_n$ for different datasets and different sketch sizes in Table A.2. These comparisons helped us to choose $b$ as we detail in the following.

**Choice of the Bernoulli parameter $b$ in practice** From the above discussion about $p_{uniform}$, heuristically, we decrease $b$ from $p_{uniform}$ to achieve faster convergence. For a large range of choices $b$, TCS converges. However, $b$ affects directly the computational cost per iteration. From (A.51), we know that if $c_d > c_n$, decreasing $b$ will increase the average cost of the method. In this case, there is a trade-off between the number of iterations and the average cost to achieve the fastest convergence in time (see Figure A.1). For a large dataset with $n$ large such as *epsilon, webspam and covtype*, we decrease $b$ slightly, as for a small dataset, we make a relatively big

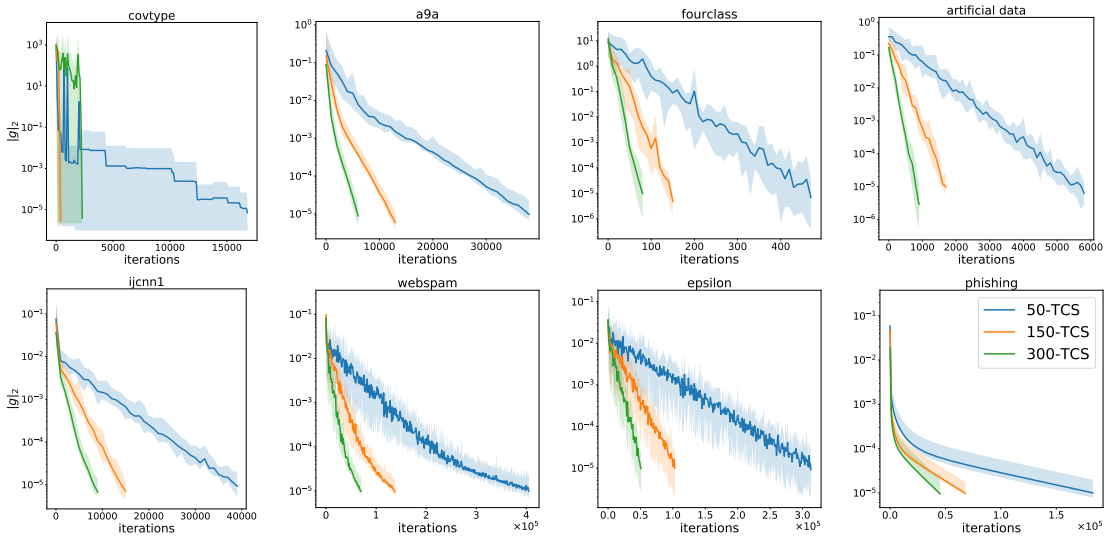


**Figure A.2** – Comparisons of different sketch sizes for TCS method in terms of the number of iterations.

decrease for $b$. If $c_d < c_n$, decreasing $b$ will also decrease the average cost. In this case, we tend to decrease $b$ even further. See Table A.1 the choices of $b$.

**Choice of the sketch size $\tau_n$** As for $\tau_n$, we observe that with bigger sketch size $\tau_n$, the method requires less number of iterations to get convergence. From Figure A.2, this is true for all the datasets except for *covtype* dataset. However, choosing bigger sketch size $\tau_n$ will also increase the cost per iteration. Consequently, there exists an optimal sketch size such that the method converges the fastest in time taking account the balance between the number of iterations and the cost per iteration. From the experiments in Figure 2.1, we show that $\tau_n = 150$ is in general a very good choice for any scale of $n$.

**Choice of the stepsizes** Different to our global convergence theories, in practice, choosing constant stepsize $\gamma > 1$ may converge faster (see Figure A.1) for certain datasets. Here we need to be careful that the stepsize we mentioned is the stepsize used for the sketch $\mathbf{S}_k = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{S}_n \end{bmatrix}$. As for the sketch $\mathbf{S}_k = \begin{bmatrix} \mathbf{S}_d & 0 \\ 0 & 0 \end{bmatrix}$, we always choose stepsize $\gamma = 1$. Because stepsize $\gamma = 1$ solves exactly the linear system. Henceforth, we use $\gamma$ to designate the stepsize used for the sketch of the last $n$ rows of $F$ (2.49). In our experiments, we find that the choice of the stepsize is related to the condition number (C.N.) of the model. If the dataset is ill-conditioned with a big C.N. of the model, $\gamma = 1$ is a good choice (see Figure 2.1 top row and Table A.1 first four lines except for a9a); if the dataset is well-conditioned with a small C.N. of the model, all $\gamma \in (1, 1.8]$ gets convergence (see Figure A.1). In practice, $\gamma = 1.8$ is a good choice for well-conditioned

datasets (see Figure 2.1). However, from the grid search of stepsizes for a9a (see Figure A.1), we know that the optimal stepsize for a9a is $\gamma = 1.5$. To avoid tuning the stepsizes, i.e. a grid search procedure, we will apply a stochastic line search process (Vaswani et al., 2019b) in the next Appendix A.11.

Furthermore, we observe that the stepsize $\gamma$ is highly related to the smoothness constant $L$. If $L$ is big, then we choose $\gamma$ close to 1, if $L$ is small, we increase $\gamma$ until $\gamma = 1.8$ (see Table A.1). Such observation remains conjecture.

Finally, to summarize in practice for the TCS method with $d < n$, we choose $\tau_d = d$ and $\tau_n = 150$, we choose $b$ following the guideline introduced above; we always choose stepsize $\gamma = 1$ for the sketch of first $d$ rows (2.49); as for the sketch of last $n$ rows (2.49), we choose stepsize $\gamma = 1$ if the dataset is ill-conditioned and we can choose stepsize $\gamma = 1.8$ if the dataset is well-conditioned.

## A.11 Stochastic line-search for TCS methods applied in GLM

In order to avoid tuning the stepsizes, we can modify Algorithm 1 by applying a stochastic line-search introduced by (Vaswani et al., 2019b). This is because again SNR can be interpreted as a SGD method. It is a *stochastic* line-search because on the $k$th iteration we sample a *stochastic* sketching matrix $\mathbf{S}_k$, and search for a stepsize $\gamma_k$ satisfying the following condition:

$$\begin{aligned} f_{\mathbf{S}_k,w^k}\left(w^k - \gamma_k \nabla f_{\mathbf{S}_k,w^k}(w^k)\right) &\leq f_{\mathbf{S}_k,w^k}(w^k) - c\cdot\gamma_k \left\|\nabla f_{\mathbf{S}_k,w^k}(w^k)\right\|^2 \\ &\overset{(2.13)}{=} (1-2c\gamma_k)\, f_{\mathbf{S}_k,w^k}(w^k). \end{aligned} \tag{A.55}$$

Here, $c > 0$ is a hyper-parameter, usually a value close to 0 is chosen in practice.

### A.11.1 Stochastic line-search for TCS method

Now we focus on GLMs, which means we develop the stochastic line-search based on (A.55) for TCS method. At $k$th iteration, if $\mathbf{S}_k = \begin{bmatrix} \mathbf{S}_d & 0 \\ 0 & 0 \end{bmatrix}$ with $\mathbf{S}_d = \mathbf{I}_{B_d}$, we sketch a linear system based on the first $d$ rows of the Jacobian (A.34). Because of this linearity, the function $f_{\mathbf{S}_k,(\alpha^k,w^k)}(\alpha^k;w^k)$ is quadratic. Thus (A.55) can be re-written as

$$\begin{aligned} f_{\mathbf{S}_k,k}\left(\begin{bmatrix}\alpha^k \\ w^k\end{bmatrix} - \gamma_k \nabla f_{\mathbf{S}_k,k}(\alpha^k;w^k)\right) &= (1-\gamma_k)^2 f_{\mathbf{S}_k,k}(\alpha^k;w^k) \\ &\leq (1-2c\gamma_k) f_{\mathbf{S}_k,k}(\alpha^k;w^k), \end{aligned} \tag{A.56}$$

where we use the shorthand $f_{\mathbf{S}_k,k}(\alpha^k;w^k) \equiv f_{\mathbf{S}_k,(\alpha^k;w^k)}(\alpha^k;w^k)$. To achieve the Armijo line-search condition (A.56), it suffices to take $\gamma = 1$ and $0 < c \leq \frac{1}{2}$ which is a common choice. Consequently, we do not need extra function evaluations. It is also well known that stepsize equal to 1 is optimal as for Newton's method applied in quadratic problems.

If $\mathbf{S}_k = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{S}_n \end{bmatrix}$ with $\mathbf{S}_n = \mathbf{I}_{B_n}$, we have

$$f_{\mathbf{S}_k,k}(\alpha^k;w^k) = \frac{1}{2}\left(\alpha^k_{B_n} + \Phi^k_{B_n}\right)^\top \left([\nabla\Phi^k_{B_n}]^\top \nabla\Phi^k_{B_n} + \mathbf{I}_{\tau_n}\right)^\dagger \left(\alpha^k_{B_n} + \Phi^k_{B_n}\right). \tag{A.57}$$

and

$$\begin{aligned} & f_{\mathbf{S}_k,k}\left(\begin{bmatrix}\alpha^k \\ w^k\end{bmatrix} - \gamma_k \nabla f_{\mathbf{S}_k,k}(\alpha^k;w^k)\right) \\ &= \frac{1}{2} F(\alpha^k + \gamma_k\Delta\alpha^k; w^k + \gamma_k\Delta w^k)^\top \mathbf{H}_{\mathbf{S}_k}(\alpha^k;w^k) F(\alpha^k + \gamma_k\Delta\alpha^k; w^k + \gamma_k\Delta w^k) \\ &= \frac{1}{2} F(\alpha^k + \gamma_k\Delta\alpha^k; w^k + \gamma_k\Delta w^k)^\top \begin{bmatrix} 0 \\ \mathbf{I}_{B_n} \end{bmatrix} \left([\nabla\Phi^k_{B_n}]^\top \nabla\Phi^k_{B_n} + \mathbf{I}_{\tau_n}\right)^\dagger \\ & \quad \begin{bmatrix} 0 \\ \mathbf{I}_{B_n} \end{bmatrix}^\top F(\alpha^k + \gamma_k\Delta\alpha^k; w^k + \gamma_k\Delta w^k) \end{aligned} \tag{A.58}$$

with

$$\begin{bmatrix} 0 \\ \mathbf{I}_{B_n} \end{bmatrix}^\top F(\alpha^k + \gamma_k\Delta\alpha^k; w^k + \gamma_k\Delta w^k) = \alpha^k_{B_n} + \gamma_k \mathbf{I}_{B_n}^\top \Delta\alpha^k + \phi'_{B_n}\left(\mathbf{A}^\top_{:,B_n} w^k + \gamma_k \mathbf{A}^\top_{:,B_n}\Delta w^k\right). \tag{A.59}$$

By (A.46), we recall that

$$\Delta\alpha^k = -\mathbf{I}_{B_n}\left([\nabla\Phi^k_{B_n}]^\top \nabla\Phi^k_{B_n} + \mathbf{I}_{\tau_n}\right)^\dagger \left(\alpha^k_{B_n} + \Phi^k_{B_n}\right), \tag{A.60}$$

$$\Delta w^k = -\nabla\Phi^k_{B_n}\left([\nabla\Phi^k_{B_n}]^\top \nabla\Phi^k_{B_n} + \mathbf{I}_{\tau_n}\right)^\dagger \left(\alpha^k_{B_n} + \Phi^k_{B_n}\right). \tag{A.61}$$

Note that the cost for evaluating (A.57) and (A.58) are $\mathcal{O}(\tau_n)$ and $\mathcal{O}\left(\max\left(\tau_n^3, \tau_n d\right)\right)$ respectively, which are not expensive. Because one part of them are essentially a by-product from the computation of $y_n$, $\Delta\alpha^k$ and $\Delta w^k$ in Algorithm 5. See Algorithm 8 the implementation of TCS combined with the stochastic Armijo line-search. $\beta \in (0,1)$ is a discount factor.

**Algorithm 8:** $\tau$-TCS+Armijo

**Input:** Stepsize $\gamma > 0$, line-search parameter $c > 0$, discount factor $\beta \in (0,1)$, sketch sizes $\tau_d, \tau_n \in \mathbb{N}$, probability $b \in (0,1)$, $v \sim B(b)$ be a Bernoulli random variable (the coin toss)

Initialize $(\alpha^0; w^0) \in \mathbb{R}^{n+d}$

**for** $k = 0, 1, \cdots$ **do**

  Sample $v \sim B(b)$ with $v \in \{0, 1\}$

  **if** $v = 0$ **then**

    Sample $B_d \subset \{1, \ldots, d\}$ with $|B_d| = \tau_d$ uniformly.

    Compute $y_d \in \mathbb{R}^{\tau_d}$ the least norm solution to

    $$\left(\frac{\mathbf{A}_{B_d,:}\mathbf{A}_{B_d,:}^\top}{\lambda^2 n^2} + \mathbf{I}_{\tau_d}\right) y_d = \frac{\mathbf{A}_{B_d,:}\alpha^k}{\lambda n} - w^k_{B_d}$$

    Compute the updates

    $$\begin{bmatrix}\Delta\alpha^k \\ \Delta w^k\end{bmatrix} = -\begin{bmatrix}\frac{1}{\lambda n}\mathbf{A}_{B_d,:}^\top \\ -\mathbf{I}_{B_d}\end{bmatrix} y_d$$

    $w^{k+1} = w^k + \Delta w^k$

    $\alpha^{k+1} = \alpha^k + \Delta\alpha^k$

  **else**

    Reset $\gamma$ to the initial stepsize.

    Sample $B_n \subset \{1, \ldots, n\}$ with $|B_n| = \tau_n$ uniformly.

    Compute $y_n \in \mathbb{R}^{\tau_n}$ the least norm solution to

    $$\left([\nabla\Phi^k_{B_n}]^\top \nabla\Phi^k_{B_n} + \mathbf{I}_{\tau_n}\right) y_n = \alpha^k_{B_n} + \Phi^k_{B_n}$$

    Compute the updates

    $$\begin{bmatrix}\Delta\alpha^k \\ \Delta w^k\end{bmatrix} = -\begin{bmatrix}\mathbf{I}_{B_n} \\ \nabla\Phi^k_{B_n}\end{bmatrix} y_n$$

    **while** $f_{\mathbf{S}_k,k}\left(\begin{bmatrix}\alpha^k \\ w^k\end{bmatrix} + \gamma\begin{bmatrix}\Delta\alpha^k \\ \Delta w^k\end{bmatrix}\right) > (1 - 2c\gamma) f_{\mathbf{S}_k,k}(\alpha^k, w^k)$ **do**

      $\gamma \leftarrow \beta \cdot \gamma$

    $w^{k+1} = w^k + \gamma\Delta w^k$

    $\alpha^{k+1} = \alpha^k + \gamma\Delta\alpha^k$

**Output:** Last iterate $\alpha^k$, $w^k$

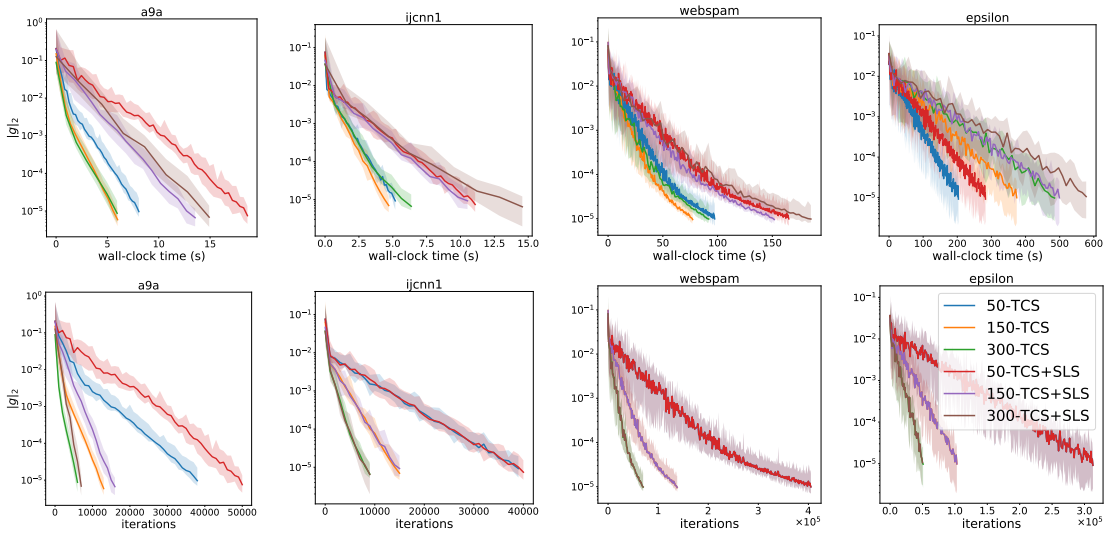


**Figure A.3** – Experiments for TCS method combined with the stochastic line-search.

### A.11.2 Experimental results for stochastic line search

For all experiments, we set the initial stepsize $\gamma = 2$ with $\gamma$ the stepsize for the last $n$ rows' sketch and reduce the stepsize by a factor $\beta = 0.9$ when the line-search (A.55) is not satisfied. We choose the stepsize $\gamma = 1$ with $\gamma$ the stepsize for the first $d$ rows' sketch and $c = 0.09$.

From Figure A.3, we observe that stochastic line search guarantees the convergence of the algorithm and does not tune any parameters. However, it slows down the convergence speed compared to the original algorithm with its rule of thumb parameters' choice. This is expected, as it does extra function evaluations at each step for the stochastic line search procedure.

# Appendix B

# Complements on Chapter 3

## Contents



The Appendix is organized as follows: In Appendix B.1, we carefully derive the closed form updates of SAN and SANA presented in Algorithm 2 and 3. In Appendix B.2, we specialize SAN and SANA for the case of regularized generalized linear models and provide more detailed and efficient pseudo-codes for such case. In Appendix B.3, we give further details on the numerical experiments and provide additional experiments for SANA to compare with SNM and IQN. In Appendix B.4 and B.5, we provide the proofs for the claims and results in Section 3.4.

## B.1 A closed form expression for SAN and SANA

In this section, we show that the updates of the SAN method given in Algorithm 2 are equivalent to the implicit formulation in (3.10)-(3.11). We then derive the closed form updates of the SANA method in Section B.1.2. Finally in Section B.1.3, we provide a useful lemma. It provides an alternatively way to directly deduce the closed form updates (3.10) and (3.11) of SAN.

### B.1.1 Closed form expression for SAN

We start with the following technical lemma.

**Lemma B.1.** *Let $j \in \{1, \dots, n\}$. Let $\hat{w} \in \mathbb{R}^d$, and $\hat{\alpha}_1, \dots, \hat{\alpha}_n \in \mathbb{R}^d$. Let $c_j \in \mathbb{R}^d$, and $\mathbf{H}_j \in \mathbb{R}^{d \times d}$ be a symmetric positive definite matrix. The optimization problem*

$$\min_{w, \alpha_1, \dots, \alpha_n \in \mathbb{R}^d} \frac{1}{2} \sum_{i=1}^{n} \|\alpha_i - \hat{\alpha}_i\|^2 + \frac{1}{2} \|w - \hat{w}\|^2_{\mathbf{H}_j}$$
$$\text{subject to } \mathbf{H}_j(w - \hat{w}) - \alpha_j = c_j,$$

*has a unique solution $(w, \alpha_1, \dots, \alpha_n)$ given by*

$$\begin{aligned} w &= \hat{w} + (\mathbf{I}_d + \mathbf{H}_j)^{-1}(c_j + \hat{\alpha}_j), \\ \alpha_j &= \hat{\alpha}_j - (\mathbf{I}_d + \mathbf{H}_j)^{-1}(c_j + \hat{\alpha}_j), \\ \alpha_i &= \hat{\alpha}_i \ \text{ for } \ i \neq j. \end{aligned}$$

*Proof.* Denoting $x = (w, \alpha_1, \dots, \alpha_n)$, let us define

$$\Phi(x) \overset{\text{def}}{=} \frac{1}{2} \sum_{i=1}^{n} \|\alpha_i - \hat{\alpha}_i\|^2 + \frac{1}{2} \|w - \hat{w}\|^2_{\mathbf{H}_j}, \quad \text{and} \quad \Psi_j(x) \overset{\text{def}}{=} \mathbf{H}_j(w - \hat{w}) - \alpha_j - c_j. \tag{B.1}$$

The fact that $\mathbf{H}_j$ is positive definite implies that $\Phi$ is strongly convex. Moreover $\Psi_j$ is affine, so we deduce that this problem has a unique solution. Moreover, this solution, let us call it $x = (w, \alpha_1, \dots, \alpha_n)$, is characterized as the unique vector in $\mathbb{R}^{(n+1)d}$ satisfying the following Karush-Kuhn-Tucker (KKT) conditions (Karush, 1939; Kuhn and Tucker, 1951):

$$(\exists \beta_j \in \mathbb{R}^d) \quad \text{such that} \quad \begin{cases} \nabla \Phi(x) + \nabla \Psi_j(x) \beta_j = 0, \\ \Psi_j(x) = 0. \end{cases} \tag{B.2}$$

The derivatives in the above KKT conditions are given by

$$\nabla \Phi(x) = \begin{bmatrix} \mathbf{H}_j(w - \hat{w}) \\ \alpha_1 - \hat{\alpha}_1 \\ \vdots \\ \alpha_n - \hat{\alpha}_n \end{bmatrix} \quad \text{and} \quad \nabla \Psi_j(x) = \begin{bmatrix} \mathbf{H}_j \\ \mathbf{0}_d \\ \vdots \\ \mathbf{I}_d \\ \vdots \\ \mathbf{0}_d \end{bmatrix} \leftarrow j + 1 \tag{B.3}$$

Using the expression for these derivatives, we can rewrite the KKT conditions (B.2) as

$$(\exists \beta_j \in \mathbb{R}^d) \quad \text{such that} \quad \begin{cases} \mathbf{H}_j(w - \hat{w}) + \mathbf{H}_j \beta_j = 0, \\ \alpha_j - \hat{\alpha}_j - \beta_j = 0, \\ \alpha_i - \hat{\alpha}_i + 0 = 0, \text{ for all } i \neq j \\ \mathbf{H}_j(w - \hat{w}) - \alpha_j - c_j = 0. \end{cases} \tag{B.4}$$

We immediately see that $\alpha_i = \hat{\alpha}_i$ for $i \neq j$. Combining the second and fourth equations in (B.4), we obtain

$$\beta_j = \alpha_j - \hat{\alpha}_j = \mathbf{H}_j(w - \hat{w}) - c_j - \hat{\alpha}_j.$$

Multiplying this new equality by $\mathbf{H}_j$ allows us to rewrite the first equation in (B.4) as:

$$\mathbf{H}_j(w - \hat{w}) + \mathbf{H}_j \beta_j = 0 \Leftrightarrow \mathbf{H}_j(w - \hat{w}) + \mathbf{H}_j^2(w - \hat{w}) = \mathbf{H}_j(c_j + \hat{\alpha}_j).$$

Using the fact that $\mathbf{H}_j$ is invertible, the latter is equivalent to write:

$$w = \hat{w} + (\mathbf{I}_d + \mathbf{H}_j)^{-1}(c_j + \hat{\alpha}_j).$$

Moreover, since $\mathbf{H}_j(\mathbf{I}_d + \mathbf{H}_j)^{-1} = \mathbf{I}_d - (\mathbf{I}_d + \mathbf{H}_j)^{-1}$, we can also turn the fourth equation in (B.4) into

$$\alpha_j = \mathbf{H}_j(w - \hat{w}) - c_j = \left(\mathbf{I}_d - (\mathbf{I}_d + \mathbf{H}_j)^{-1}\right)(c_j + \hat{\alpha}_j) - c_j = \hat{\alpha}_j - (\mathbf{I}_d + \mathbf{H}_j)^{-1}(c_j + \hat{\alpha}_j).$$

This proves the claim. □

**Lemma B.2.** *Let $\pi \in [0, 1]$ and $\gamma \in (0, 1]$ a step size. Algorithm 2 (SAN) is equivalent to the following algorithm:*
*With probability $\pi$, update according to*

$$\begin{cases} \bar{x}^{k+1} = \operatorname{argmin} \|w - w^k\|^2 + \sum_{i=1}^n \|\alpha_i - \alpha_i^k\|^2 \quad \textit{subject to} \quad \frac{1}{n}\sum_{i=1}^n \alpha_i = 0, \\ x^{k+1} = (1 - \gamma)x^k + \gamma \bar{x}^{k+1}, \end{cases} \tag{B.5}$$

*Otherwise with probability $(1 - \pi)$, sample $j \sim \{1, \dots, n\}$ uniformly and update according to*

$$\begin{cases} \bar{x}^{k+1} = & \operatorname{argmin} \|w - w^k\|^2_{\nabla^2 f_j(w^k)} + \sum_{i=1}^n \|\alpha_i - \alpha_i^k\|^2 \\ & \textit{subject to} \quad \nabla^2 f_j(w^k)(w - w^k) - \alpha_j = -\nabla f_j(w^k), \\ x^{k+1} = & (1 - \gamma)x^k + \gamma \bar{x}^{k+1}. \end{cases} \tag{B.6}$$

*Proof.* Suppose that we are in the case (which holds with probability $\pi$) given by (B.5). In the projection step, we see that $w$ is not present in the constraint, which implies that $\bar{w}^{k+1} = w^k$, and therefore $w^{k+1} = w^k$. On the other hand, $(\bar{\alpha}_1, \ldots, \bar{\alpha}_n)$ is the projection of $(\alpha_1, \ldots, \alpha_n)$ onto a simple linear constraint, and can be computed in closed form as

$$(\forall i \in \{1, \ldots, n\}) \quad \bar{\alpha}_i^{k+1} = \alpha_i^k - \frac{1}{n} \sum_{i=1}^{n} \alpha_i^k.$$

Consequently

$$(\forall i \in \{1, \ldots, n\}) \quad \alpha_i^{k+1} = (1-\gamma)\alpha_i^k + \gamma \left( \alpha_i^k - \frac{1}{n} \sum_{i=1}^{n} \alpha_i^k \right) = \alpha_i^k - \frac{\gamma}{n} \sum_{i=1}^{n} \alpha_i^k,$$

which gives us exactly the step 4 in Algorithm 2.

Let now $j$ be in $\{1, \ldots, n\}$ sampled uniformly, and suppose that we are in the case given by (B.6). Using Lemma B.1 we can compute an explicit form for $\bar{x}^{k+1}$ given by

$$\begin{aligned}
\bar{w}^{k+1} &= w^k + \left( \mathbf{I}_d + \nabla^2 f_j(w^k) \right)^{-1} (\alpha_j^k - \nabla f_j(w^k)), \\
\bar{\alpha}_j^{k+1} &= \alpha_j^k - \left( \mathbf{I}_d + \nabla^2 f_j(w^k) \right)^{-1} (\alpha_j^k - \nabla f_j(w^k)), \\
\bar{\alpha}_i^{k+1} &= \hat{\alpha}_i^k \quad \text{for all} \quad i \neq j.
\end{aligned}$$

Consequently, after applying the relaxation step we have

$$\begin{aligned}
w^{k+1} &= w^k + \gamma \left( \mathbf{I}_d + \nabla^2 f_j(w^k) \right)^{-1} (\alpha_j^k - \nabla f_j(w^k)), \\
\alpha_j^{k+1} &= \alpha_j^k - \gamma \left( \mathbf{I}_d + \nabla^2 f_j(w^k) \right)^{-1} (\alpha_j^k - \nabla f_j(w^k)), \\
\bar{\alpha}_i^{k+1} &= \hat{\alpha}_i^k \quad \text{for all} \quad i \neq j.
\end{aligned}$$

which is exactly the steps 8-10 in Algorithm 2. □

### B.1.2 Closed form expression for SANA

**Lemma B.3.** *Let $j \in \{1, \ldots, n\}$. Let $c_j \in \mathbb{R}^d$, $\hat{w} \in \mathbb{R}^d$, and let $\hat{\alpha}_1, \ldots, \hat{\alpha}_n \in \mathbb{R}^d$ be such that $\sum_{i=1}^n \hat{\alpha}_i = 0$. Let $\mathbf{H}_j \in \mathbb{R}^{d \times d}$ be a positive definite matrix. Then the optimization problem*

$$\min_{w, \alpha_1, \ldots, \alpha_n \in \mathbb{R}^d} \frac{1}{2} \sum_{i=1}^{n} \|\alpha_i - \hat{\alpha}_i\|^2 + \frac{1}{2} \|w - \hat{w}\|_{\mathbf{H}_j}^2,$$
$$\text{subject to } \mathbf{H}_j(w - \hat{w}) - \alpha_j = c_j,$$

$$\sum_{i=1}^{n} \alpha_i = 0, \tag{B.7}$$

*has a unique solution* $(w, \alpha_1, \dots, \alpha_n)$ *given by*

$$\begin{aligned} d &= \left(\frac{n-1}{n}\mathbf{I}_d + \mathbf{H}_j\right)^{-1}(c_j + \hat{\alpha}_j), \\ w &= \hat{w} + d, \\ \alpha_j &= \hat{\alpha}_j - \frac{n-1}{n}d, \\ \alpha_i &= \hat{\alpha}_i + \frac{1}{n}d, \quad \textit{for } i \neq j. \end{aligned}$$

*Proof.* Noting $x = (w, \alpha_1, \dots, \alpha_n)$, let us define $\Phi(x)$ and $\Psi_j(x)$ as in (B.1), together with

$$\Psi_0(x) \overset{\text{def}}{=} \sum_{i=1}^{n} \alpha_i.$$

The fact that $\mathbf{H}_j$ is positive definite implies that we are minimizing a strongly convex function over a set of affine equations. We deduce that this problem has a unique solution. Moreover, this solution, let us call it $x = (w, \alpha_1, \dots, \alpha_n)$, is characterized as the unique vector in $\mathbb{R}^{(n+1)d}$ satisfying the following KKT conditions:

$$(\exists \beta_0, \beta_j \in \mathbb{R}^d) \quad \text{such that} \quad \begin{cases} \nabla\Phi(x) + \nabla\Psi_0(x)\beta_0 + \nabla\Psi_j(x)\beta_j = 0, \\ \Psi_0(x) = 0, \\ \Psi_j(x) = 0. \end{cases}$$

Here, we can compute $\nabla\Phi(x)$ and $\nabla\Psi_j(x)$ as in (B.3), together with

$$\nabla\Psi_0(x) = \begin{bmatrix} \mathbf{0}_d \\ \mathbf{I}_d \\ \vdots \\ \mathbf{I}_d \end{bmatrix}.$$

Therefore, we can rewrite the KKT conditions as

$$(\exists \beta_0, \beta_j \in \mathbb{R}^d) \quad \text{such that} \quad \begin{cases} \mathbf{H}_j(w - \hat{w}) + 0 + \mathbf{H}_j\beta_j = 0, \\ \alpha_j - \hat{\alpha}_j + \beta_0 - \beta_j = 0, \\ \alpha_i - \hat{\alpha}_i + \beta_0 + 0 = 0, \text{ for all } i \neq j \\ \sum_{i=1}^n \alpha_i = 0 \\ \mathbf{H}_j(w - \hat{w}) - \alpha_j - c_j = 0. \end{cases} \tag{B.8}$$

The last equation in (B.8) can be rewritten as

$$\alpha_j = \mathbf{H}_j(w - \hat{w}) - c_j. \tag{B.9}$$

Summing the equations involving $\alpha_i$ for $i \neq j$, and using the fact that $\sum_{i=1}^n \alpha_i = \sum_{i=1}^n \hat{\alpha}_i = 0$, together with (B.9), we can deduce that

$$0 = \sum_{i \neq j} (\alpha_i - \hat{\alpha}_i + \beta_0) = -\alpha_j + \hat{\alpha}_j + (n-1)\beta_0 = -\mathbf{H}_j(w - \hat{w}) + c_j + \hat{\alpha}_j + (n-1)\beta_0.$$

In other words, we obtain that

$$\beta_0 = \frac{1}{n-1} \left( \mathbf{H}_j(w - \hat{w}) - (c_j + \hat{\alpha}_j) \right). \tag{B.10}$$

Injecting the above expression into the second equation of (B.8), and using again (B.9), gives

$$\begin{aligned} \beta_j &= \mathbf{H}_j(w - \hat{w}) - c_j - \hat{\alpha}_j + \frac{1}{n-1} \left( \mathbf{H}_j(w - \hat{w}) - (c_j + \hat{\alpha}_j) \right) \\ &= \frac{n}{n-1} \left( \mathbf{H}_j(w - \hat{w}) - (c_j + \hat{\alpha}_j) \right) \end{aligned}$$

Combining this expression of $\beta_j$ with the first equation in (B.8) leads to

$$\begin{aligned} 0 &= \mathbf{H}_j(w - \hat{w}) + \frac{n}{n-1} \mathbf{H}_j \left( \mathbf{H}_j(w - \hat{w}) - (c_j + \hat{\alpha}_j) \right) \\ &= \mathbf{H}_j \left( \left( \mathbf{I}_d + \frac{n}{n-1} \mathbf{H}_j \right) (w - \hat{w}) - \frac{n}{n-1} (c_j + \hat{\alpha}_j) \right) \end{aligned}$$

Using the fact that $\mathbf{H}_j$ is positive definite, we obtain that:

$$\begin{aligned} w &= \hat{w} + \frac{n}{n-1} \left( \mathbf{I}_d + \frac{n}{n-1} \mathbf{H}_j \right)^{-1} (c_j + \hat{\alpha}_j) \\ &= \hat{w} + \left( \frac{n-1}{n} \mathbf{I}_d + \mathbf{H}_j \right)^{-1} (c_j + \hat{\alpha}_j), \\ &= \hat{w} + d, \end{aligned}$$

where $d$ is defined as $\left(\frac{n-1}{n}\mathbf{I}_d + \mathbf{H}_j\right)^{-1}(c_j + \hat{\alpha}_j)$.

Going back now to (B.9) we can write

$$\begin{aligned}
\alpha_j &= \mathbf{H}_j d - c_j = \mathbf{H}_j \left(\frac{n-1}{n}\mathbf{I}_d + \mathbf{H}_j\right)^{-1}(c_j + \hat{\alpha}_j) - c_j \\
&= \left(\mathbf{I}_d - \frac{n-1}{n}\left(\frac{n-1}{n}\mathbf{I}_d + \mathbf{H}_j\right)^{-1}\right)(c_j + \hat{\alpha}_j) - c_j \\
&= \hat{\alpha}_j - \frac{n-1}{n}\left(\frac{n-1}{n}\mathbf{I}_d + \mathbf{H}_j\right)^{-1}(c_j + \hat{\alpha}_j) \\
&= \hat{\alpha}_j - \frac{n-1}{n}d.
\end{aligned}$$

It remains to compute $\alpha_i$, for $i \neq j$. Start with the first equation of (B.8) and see that $w - \hat{w} + \beta_j = 0$. This implies that $\beta_j = -d$. We can therefore use the second equation of (B.8) to write that

$$\beta_0 = \beta_j - (\alpha_j - \hat{\alpha}_j) = -d + \frac{n-1}{n}d = -\frac{1}{n}d.$$

We can finally call the third equation of (B.8) and write that $\alpha_i = \hat{\alpha}_i - \beta_0 = \hat{\alpha}_i + \frac{1}{n}d$. □

**Lemma B.4.** *Let $\gamma \in (0, 1]$ be a step size. Algorithm 3 (SANA) is equivalent to the following algorithm: update the iterates according to*

$$\begin{cases}
\bar{x}^{k+1} = & \operatorname{argmin} \|w - w^k\|^2_{\nabla^2 f_j(w^k)} + \sum_{i=1}^n \|\alpha_i - \alpha_i^k\|^2 \\
& \quad \textit{subject to} \begin{cases} \frac{1}{n}\sum_{i=1}^n \alpha_i = 0, \\ \nabla^2 f_j(w^k)(w - w^k) - \alpha_j = -\nabla f_j(w^k), \end{cases} \\
x^{k+1} = & (1-\gamma)x^k + \gamma \bar{x}^{k+1}.
\end{cases} \tag{B.11}$$

*Proof.* Consider the iterates defined by (B.11). Using Lemma B.3, we can compute an explicit form for $\bar{x}^{k+1}$:

$$\begin{aligned}
d^k &= \left(\frac{n-1}{n}\mathbf{I}_d + \nabla^2 f_j(w^k)\right)^{-1}(\alpha_j^k - \nabla f_j(w^k)), \\
\bar{w}^{k+1} &= w^k + d^k, \\
\bar{\alpha}_j^{k+1} &= \alpha_j^k - \frac{n-1}{n}d^k, \\
\bar{\alpha}_i^{k+1} &= \alpha_i^k + \frac{1}{n}d^k, \quad \text{for } i \neq j.
\end{aligned}$$

After applying the relaxation step $x^{k+1} = (1-\gamma)x^k + \gamma\bar{x}^{k+1}$, we obtain exactly the steps 5-8 in Algorithm 3. □

### B.1.3 Generic projection onto linear systems

Here we provide a useful lemma that can directly deduce the closed form updates of (3.10) and (3.11) of SAN. It will also be used later in the appendix.

**Lemma B.5.** *Let* $\mathbf{A} \in \mathbb{R}^{n\times d}$, $\mathbf{S} \in \mathbb{R}^{n\times\tau}$, $b \in \mathbf{Im}(\mathbf{A})$, *and* $\mathbf{H}$ *be a symmetric positive definite matrix. The optimization problem*

$$x^* = \arg\min_{x\in\mathbb{R}^d} \quad \frac{1}{2}\|x\|^2_{\mathbf{H}}, \\ \text{subject to } \mathbf{S}^\top\mathbf{A}x = \mathbf{S}^\top b,$$

*has a unique solution, called the weighted sketch-and-project optimal solution:*

$$x^* = \mathbf{H}^{-1}\mathbf{A}^\top\mathbf{S}(\mathbf{S}^\top\mathbf{A}\mathbf{H}^{-1}\mathbf{A}^\top\mathbf{S})^\dagger\mathbf{S}^\top b. \tag{B.12}$$

*Proof.* First, note that this problem is strongly convex because $\mathbf{H}$ is supposed to be positive definite, and therefore admits a unique solution $x^* \in \mathbb{R}^d$. Second, since $\mathbf{H}$ is invertible, we can do the change of variables $y \overset{\text{def}}{=} \mathbf{H}^{1/2}x$. This allows us to write that $x^* = (\mathbf{H})^{-1/2}y^*$ where $y^*$ is the unique solution of

$$\arg\min_{y\in\mathbb{R}^d} \frac{1}{2}\|y\|^2, \\ \text{subject to } \mathbf{S}^\top\mathbf{A}\mathbf{H}^{-1/2}y = \mathbf{S}^\top b.$$

The unique solution to the above problem is the minimal-norm solution of the linear system $\mathbf{S}^\top\mathbf{A}(\mathbf{H})^{-1/2}y = \mathbf{S}^\top b$, which can be simply expressed by using the pseudo-inverse (Ben-Israel and Charnes, 1963, Definition 1) :

$$y^* = \left(\mathbf{S}^\top\mathbf{A}\mathbf{H}^{-1/2}\right)^\dagger\mathbf{S}^\top b.$$

Using the relation $\mathbf{M}^\dagger = \mathbf{M}^\top(\mathbf{M}\mathbf{M}^\top)^\dagger$ (Penrose, 1955, Lemma 1 & Equation (10)), we obtain

$$y^* = (\mathbf{H})^{-1/2}\mathbf{A}^\top\mathbf{S}(\mathbf{S}^\top\mathbf{A}\mathbf{H}^{-1}\mathbf{A}^\top\mathbf{S})^\dagger\mathbf{S}^\top b.$$

Multiplying this equality by $\mathbf{H}^{-1/2}$ gives us the desired expression for $x^*$. □

This lemma is useful. Later it will be applied in Lemma B.14 and consequently provide the explicit updates of (3.10) and (3.11) of SAN in Section B.4.2. Thus this is a different way to obtain the closed form updates of SAN, compared to Section B.1.1.

## B.2 Implementations for regularized GLMs

### B.2.1 Definition and examples

Here we specify our algorithms for the case of regularized generalized linear models. Throughout this section, we assume that our finite sum minimization problem (3.1) is a GLM (generalized linear model) defined as follows.

**Assumption B.6** (Regularized GLM)**.** *Our problem* (3.1) *writes as*

$$\min_{w\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1} f_i(w) \overset{def}{=} \phi_i(\langle a_i, w\rangle) + R(w), \tag{B.13}$$

*where* $\{a_i\}_{i=1}^n \subset \mathbb{R}^d$ *are* data points, $\{\phi_i\}_{i=1}^n$ *are twice differentiable real convex* loss *functions with* $\phi_i''(t) > 0$*, and* $R$ *is a separable* regularizer *with* $R(w) = \sum_{j=1}^d R_j(w_j)$ *where* $R_j$ *is a twice differentiable real convex function with* $R_j''(t) > 0$*, for all* $t \in \mathbb{R}$*.*

Some classic examples of GLMs include ridge regression where $\phi_i(t) = \frac{1}{2}(t - y_i)^2$ and $R_j(t) = \frac{\lambda}{2}t^2$ where $\lambda > 0$ is a regularization parameter. L2-regularized logistic regression, the example on which we perform most of our experiments, is also a GLM with

$$\phi_i(t) = \ln\left(1 + e^{-y_i t}\right) \quad \text{and} \quad R_j(t) = \frac{\lambda}{2}t^2. \tag{B.14}$$

We also consider other forms of separable regularizers such as the pseudo-huber regularizer where $R_j(t) = \lambda\delta^2\left(\sqrt{1 + \left(\frac{t}{\delta}\right)^2} - 1\right)$ where $\delta$ is a parameter.

In the next section, we will show that for GLMs, our methods can be efficiently implemented. But first we need the following preliminary results.

**Lemma B.7** (Simple computations with Regularized GLMs)**.** *For GLMs* (*Assumption B.6*) *we have for all* $j \in \{1, \dots, n\}$*, all* $w \in \mathbb{R}^d$ *and every* $\mu \geq 0$ *that*

$$(1) \quad \begin{aligned} \nabla R(w) &= [R_1'(w_1) \dots R_d'(w_d)]^\top, \\ \nabla^2 R(w) &= \mathbf{Diag}\left(R_1''(w_1), \dots, R_d''(w_d)\right). \end{aligned}$$

(2) $$\begin{aligned} \nabla f_j(w) &= \nabla R(w) + \phi_j'(\langle a_j, w\rangle) a_j, \\ \nabla^2 f_j(w) &= \nabla^2 R(w) + \phi_j''(\langle a_j, w\rangle) a_j a_j^\top. \end{aligned}$$

(3) *With* $\hat{a}_j := \left(\mu \mathbf{I}_d + \nabla^2 R(w_k)\right)^{-1} a_j$, *we have*

$$\left(\mu \mathbf{I}_d + \nabla^2 f_j(w)\right)^{-1} = \left(\mu \mathbf{I}_d + \nabla^2 R(w_k)\right)^{-1} - \frac{\phi_j''(\langle a_j, w\rangle)}{1 + \phi_j''(\langle a_j, w\rangle)\langle \hat{a}_j, a_j\rangle} \hat{a}_j \hat{a}_j^\top.$$

(4) *If* $R(w) = \frac{\lambda}{2}\|w\|^2$, *with* $\lambda > 0$, *then*

$$\left(\mu \mathbf{I}_d + \nabla^2 f_j(w)\right)^{-1} = \frac{1}{\mu + \lambda}\left(\mathbf{I}_d - \frac{\phi_j''(\langle a_j, w\rangle)}{\mu + \lambda + \phi_j''(\langle a_j, w\rangle)\|a_j\|^2} a_j a_j^\top\right).$$

*Proof.* (1) and (2) are trivial. For (3), let $\Phi := \phi_j''(\langle a_j, w\rangle)$, which is nonnegative because of the Assumption B.6. Consider now the Sherman–Morrison formula:

$$(\mathbf{M} + uu^\top)^{-1} = \mathbf{M}^{-1} - \frac{1}{1 + \langle \mathbf{M}^{-1}u, u\rangle}(\mathbf{M}^{-1}u)(\mathbf{M}^{-1}u)^\top.$$

This allows us to write, for $\mathbf{D} = (\mu \mathbf{I}_d + \nabla^2 R(w))^{-1}$ and $\mathbf{M} = \Phi^{-1}(\mu \mathbf{I}_d + \nabla^2 R(w))$, that

$$\begin{aligned} \left(\mu \mathbf{I}_d + \nabla^2 f_j(w)\right)^{-1} &= \left(\mu \mathbf{I}_d + \nabla^2 R(w) + \Phi a_j a_j^\top\right)^{-1} = \Phi^{-1}\left(\mathbf{M} + a_j a_j^\top\right)^{-1} \\ &= \Phi^{-1}\left(\mathbf{M}^{-1} - \frac{1}{1 + \langle \mathbf{M}^{-1}a_j, a_j\rangle}(\mathbf{M}^{-1}a_j)(\mathbf{M}^{-1}a_j)^\top\right) \\ &= \mathbf{D} - \frac{\Phi}{1 + \Phi\langle \mathbf{D}a_j, a_j\rangle}(\mathbf{D}a_j)(\mathbf{D}a_j)^\top. \end{aligned}$$

(4) is a direct consequence of the fact that $\mathbf{D} = \left(\mu \mathbf{I}_d + \nabla^2 R(w_k)\right)^{-1} = \frac{1}{\mu+\lambda}\mathbf{I}_d$. □

### B.2.2 SAN with GLMs

Here we give the detailed derivation of our implementation of SAN for GLMs, see Algorithm 9. Upon examination, we can see that every step of Algorithm 9 has a cost of $\mathcal{O}(d)$, except on line 4. As explained in Section 3.2.1, the averaging cost on line 4 costs $\mathcal{O}(d)$ in which $\pi$ is of the order of $\mathcal{O}(1/n)$. The only step that we have left an implicit computation is on lines 9 and 10 which require inverting $(\mathbf{I}_d + \nabla^2 R(w^k))$. But this to comes at a cost of $\mathcal{O}(d)$ since in our Assumption B.6 the regularizer is separable, and thus the Hessian is a diagonal matrix whose inversion also costs $\mathcal{O}(d)$.

**Algorithm 9:** SAN for regularized GLMs

**Input:** Data $\{a_i\}_{i=1}^n$, loss functions $\{\phi_i\}_{i=1}^n$, regularizer $R$, $\pi \in (0,\, 1)$, step size $\gamma \in (0,\, 1]$, max iteration $T$

1 Initialize $\alpha_1^0, \cdots, \alpha_n^0, w^0 \in \mathbb{R}^d$ and $\overline{\alpha}^0 = \frac{1}{n}\sum_{i=1}^n \alpha_i^0$.

2 **for** $k = 0, \ldots, T-1$ **do**

3 **With probability** $\pi$ **update:**

4 $\alpha_i^{k+1} = \alpha_i^k - \gamma\overline{\alpha}^k, \quad$ for all $i \in \{1, \cdots, n\}$

5 $w^{k+1} = w^k$

6 **Otherwise with probability** $(1-\pi)$**:**

7 Sample $j \in \{1, \ldots, n\}$ uniformly

8 $g^k = \nabla R(w^k) + \phi_j'(\langle a_j, w^k\rangle)a_j - \alpha_j^k$

9 $\hat{a}_j = (\mathbf{I}_d + \nabla^2 R(w^k))^{-1} a_j$

10 $d^k = \frac{\phi_j''(\langle a_j, w^k\rangle)\langle \hat{a}_j, g^k\rangle}{1+\phi_j''(\langle a_j, w^k\rangle)\langle \hat{a}_j, a_j\rangle}\hat{a}_j - \left(\mathbf{I}_d + \nabla^2 R(w^k)\right)^{-1} g^k$

11 $w^{k+1} = w^k + \gamma d^k$

12 $\alpha_j^{k+1} = \alpha_j^k - \gamma d^k$

13 $\alpha_i^{k+1} = \alpha_i^k$ for $i \neq j$

14 $\overline{\alpha}^{k+1} = \overline{\alpha}^k - \frac{\gamma}{n} d^k$

**Output:** Last iterate $w^T$

Next we formalize the costs of Algorithm 9 in the following remark. By computational cost, we refer to the total number of floating point operations, that is the number of scalar multiplications and additions.

**Remark B.8.** *The average costs of SAN (Algorithm 9) per iteration under Assumption B.6 are:*

- *`Memory storage` of* $\mathcal{O}(nd)$ *scalars.*
- *`Memory access` of* $\mathcal{O}(\pi nd + (1-\pi)d)$ *which is* $\mathcal{O}(d)$ *when* $\pi \simeq 1/n$.
- *`Data access` of* $\mathcal{O}(1)$ .
- *`Computational cost` of* $\mathcal{O}(\pi dn + (1-\pi)d)$ *which is* $\mathcal{O}(d)$ *when* $\pi \simeq 1/n$.

In calculating the average computational cost per iteration, we used that in expectation the updates on lines 4–5 occur with probability $\pi$, while the updates on lines 7–14 occur with probability $(1-\pi)$.

**Lemma B.9.** *The SAN Algorithm 2 applied to Regularized GLMs (in the sense of Assumption B.6) is Algorithm 9.*

*Proof.* Let $k \in \{0, \ldots, T-1\}$. With probability $\pi$ from Algorithm 2 we have

$$\alpha_i^{k+1} \;=\; \alpha_i^k - \frac{\gamma}{n}\sum_{j=1}^n \alpha_j^k, \quad \text{for all } i \in \{1, \cdots, n\}.$$

This can be rewritten as $\alpha_i^{k+1} = \alpha_i^k - \gamma\bar{\alpha}^k$, which is the update on line 4 in Algorithm 9.

With probability $(1-\pi)$ from Algorithm 2 we have

$$\begin{aligned} d^k &= -\left(\mathbf{I}_d + \nabla^2 f_j(w^k)\right)^{-1}\left(\nabla f_j(w^k) - \alpha_j^k\right), \\ w^{k+1} &= w^k + \gamma d^k, \\ \alpha_j^{k+1} &= \alpha_j^k - \gamma d^k. \end{aligned}$$

Using Lemma B.7, we see that

$$g^k := \nabla f_j(w^k) - \alpha_j^k = \nabla R(w^k) + \phi_j'(\langle a_j, w^k\rangle)a_j - \alpha_j^k.$$

Still using Lemma B.7, and introducing the notation $\hat{a}_j = (\mathbf{I}_d + \nabla^2 R(w^k))^{-1}a_j$, we see that

$$\left(\mathbf{I}_d + \nabla^2 f_j(w^k)\right)^{-1} = \left(\mathbf{I}_d + \nabla^2 R(w^k)\right)^{-1} - \frac{\phi_j''(\langle a_j, w^k\rangle)}{1 + \phi_j''(\langle a_j, w^k\rangle)\langle \hat{a}_j, a_j\rangle}\hat{a}_j\hat{a}_j^\top.$$

Therefore,

$$d^k = \frac{\phi_j''(\langle a_j, w^k\rangle)\langle \hat{a}_j, g^k\rangle}{1 + \phi_j''(\langle a_j, w^k\rangle)\langle \hat{a}_j, a_j\rangle}\hat{a}_j - \left(\mathbf{I}_d + \nabla^2 R(w^k)\right)^{-1} g^k,$$

which concludes the proof. □

Finally, when the regularizer is the L2 norm, then we can implement SAN even more efficiently as follows.

**Example B.10** (Ridge regularization). *If $R(w) = \frac{\lambda}{2}\|w\|^2$ with $\lambda > 0$, then the stochastic Newton direction $d^k$ can be computed explicitly (see also Lemma B.7):*

$$d^k = \frac{\phi_j''(r^k)}{1+\lambda} \cdot \frac{\langle a_j, \alpha_j^k\rangle - \phi_j'(r^k)\|a_j\|^2 - \lambda r^k}{1 + \lambda + \phi_j''(r^k)\|a_j\|^2}a_j - \frac{1}{1+\lambda}\left(\lambda w^k + \phi_j'(r^k)a_j - \alpha_j^k\right),$$

*where $r^k = \langle a_j, w^k\rangle$.*

### B.2.3 SANA with GLMs

In Algorithm 10 we give the specialized implementation of SANA (Algorithm 3) for GLMs.

**Algorithm 10:** SANA for regularized GLMs

**Input:** Data $\{a_i\}_{i=1}^n$, loss functions $\{\phi_i\}_{i=1}^n$, regularizer $R$, step size $\gamma \in (0,\, 1]$, max iteration $T$

Initialize $\alpha_1^0, \cdots, \alpha_n^0, w^0 \in \mathbb{R}^d$, with $\sum_{i=1}^n \alpha_i^0 = 0$;
Pre-compute $\mu = \frac{n-1}{n}$;
**for** $k = 0, \ldots, T-1$ **do**
  Sample $j \in \{1, \ldots, n\}$ uniformly;
  $g^k = \nabla R(w^k) + \phi_j'(\langle a_j, w^k \rangle) a_j - \alpha_j^k$
  $\hat{a}_j = (\mu \mathbf{I}_d + \nabla^2 R(w^k))^{-1} a_j$
  $d^k = \frac{\phi_j''(\langle a_j, w^k \rangle)\langle \hat{a}_j, g^k \rangle}{1+\phi_j''(\langle a_j, w^k \rangle)\langle \hat{a}_j, a_j \rangle} \hat{a}_j - \left(\mu \mathbf{I}_d + \nabla^2 R(w^k)\right)^{-1} g^k$
  $w^{k+1} = w^k + \gamma d^k$
  $\alpha_j^{k+1} = \alpha_j^k + \gamma \mu d^k$
  $\alpha_i^{k+1} = \alpha_i^k - \frac{\gamma}{n} d^k, \quad$ for $i \neq j$

**Output:** Last iterate $w^T$

Next we formalize the costs of Algorithm 10 in the following remark.

**Remark B.11.** *The costs of SANA (Algorithm 10) per iteration under Assumption B.6 are:*

- *`Memory storage` of* $\mathcal{O}(nd)$ *scalars.*
- *`Memory access` of* $\mathcal{O}(nd)$.
- *`Data access` of* $\mathcal{O}(1)$ .
- *`Computational cost` of* $\mathcal{O}(nd)$.

**Lemma B.12.** *The SANA Algorithm 3 applied to Regularized GLMs (in the sense of Assumption B.6) is Algorithm 10.*

*Proof.* Let $k \in \{0, \ldots, T-1\}$, and $\mu := 1 - n^{-1}$. Let $j$ be sampled over $\{1, \ldots, n\}$ uniformly. From Algorithm 3 we have

$$
\begin{aligned}
g^k &= \nabla f_j(w^k) - \alpha_j^k, \\
d^k &= -\left(\mu \mathbf{I}_d + \nabla^2 f_j(w^k)\right)^{-1} g^k, \\
w^{k+1} &= w^k + \gamma d^k, \\
\alpha_j^{k+1} &= \alpha_j^k - \gamma \mu d^k, \\
\alpha_i^{k+1} &= \alpha_i^k + \tfrac{\gamma}{n} d^k, \quad \text{for } i \neq j.
\end{aligned}
$$

Using Lemma B.7, we see that

$$g^k = \nabla R(w^k) + \phi_j'(\langle a_j, w^k \rangle) a_j - \alpha_j^k.$$

Still using Lemma B.7, and introducing the notation $\hat{a}_j = (\mu \mathbf{I}_d + \nabla^2 R(w^k))^{-1} a_j$, we have that

$$\left(\mu \mathbf{I}_d + \nabla^2 f_j(w^k)\right)^{-1} = \left(\mu \mathbf{I}_d + \nabla^2 R(w^k)\right)^{-1} - \frac{\phi_j''(\langle a_j, w^k \rangle)}{1 + \phi_j''(\langle a_j, w^k \rangle)\langle \hat{a}_j, a_j \rangle} \hat{a}_j \hat{a}_j^\top.$$

Therefore,

$$d^k = \frac{\phi_j''(\langle a_j, w^k \rangle)\langle \hat{a}_j, g^k \rangle}{1 + \phi_j''(\langle a_j, w^k \rangle)\langle \hat{a}_j, a_j \rangle} \hat{a}_j - \left(\mu \mathbf{I}_d + \nabla^2 R(w^k)\right)^{-1} g^k,$$

which concludes the proof.

□

## B.3 Experimental details in Section 3.3 and additional experiments

We present the details of the experiments in Section 3.3, in order to guide readers to reproduce the exact same results in Figure 3.1 and Figure 3.2. We also explain some grid search results about sensitivity of hyperparameters in Section B.3.3, showing in particular that SAN does not require parameter tuning. Then we provide additional experiments for SANA, SNM and IQN in Section B.3.4 which are not included in Chapter 3. These experimental results support that SANA introduced in Section 3.2.2 is also a reasonable method. Finally, we provide experiments to compare SAN and SAN without variable metric in Section B.3.5 to illustrate the importance of such variable metric.

### B.3.1 Experimental details in Section 3.3

All experiments in Section 3.3 were run in Python 3.7.7 on a laptop with an Intel Core i9-9980HK CPU and 32 Gigabyte of DDR4 RAM running OSX 11.3.1.

All datasets were taken directly from LibSVM (Chang and Lin, 2011) on https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/ and the scaled versions were used if provided. All datasets were preprocessed by adding an intercept, i.e. a constant feature one. For the datasets whose binary labels are not in $\{-1, 1\}$, e.g., `phishing`, `mushrooms` and `covtype`, we assigned $-1$ to the smallest labels and $+1$ to those largest ones. All learnable parameters were initialized by zeros, e.g., $w^0 = 0 \in \mathbb{R}^d$ and $\alpha_i^0 = 0 \in \mathbb{R}^d$ for $i = 1, \cdots, n$ for SAN.

Table B.1 provides the details of the datasets we used in Section 3.3, including the condition number and $L_{\max}$. For a given dataset, let $\mathbf{A} = [a_1 \cdots a_n] \in \mathbb{R}^{d \times n}$ be the data matrix, the

**Table B.1** – Details of the binary data sets used in the logistic regression experiments

| dataset | dimension ($d$) | samples ($n$) | $L_{\max}$ | sparsity | condition number |
|---|---|---|---|---|---|
| phishing | $68+1$ | 11055 | 0.5001 | 0.5588 | $4.1065 \times 10^{18}$ |
| mushrooms | $112+1$ | 8124 | 5.5001 | 0.8125 | $1.3095 \times 10^{19}$ |
| ijcnn1 | $22+1$ | 49990 | 1.2342 | 0.4091 | 25.6587 |
| covtype | $54+1$ | 581012 | 2.154 | 0.7788 | $9.6926 \times 10^{17}$ |
| webspam | $254+1$ | 350000 | 0.5 | 0.6648 | $6.9973 \times 10^{255}$ |
| epsilon | $2000+1$ | 400000 | 0.5 | 0.0 | $3.2110 \times 10^{10}$ |
| rcv1 | $47236+1$ | 20242 | 0.5 | 0.9984 | $5.3915 \times 10^{25}$ |
| real-sim | $20958+1$ | 72309 | 0.5 | 0.9976 | $1.3987 \times 10^{20}$ |

condition number in Table B.1 is computed by

$$\text{condition number of the dataset} \overset{\text{def}}{=} \sqrt{\frac{\lambda_{\max}\left(\mathbf{A}\mathbf{A}^\top\right)}{\lambda_{\min}^{+}\left(\mathbf{A}\mathbf{A}^\top\right)}},$$

where $\lambda_{\max}$ and $\lambda_{\min}^{+}$ are the largest and smallest non-zero eigenvalue operators respectively. $L_{\max}$ is defined as $L_{\max} = \max_{i=1,\dots,n} L_i$, where $L_i = \frac{1}{4}\|a_i\|^2 + \lambda$ is the smoothness constant of the regularized logistic regression $f_i$. Notice that the step size's choice for SAG and SVRG is of the order of $\mathcal{O}(1/L_{\max})$.

From Table B.1, note that we have datasets that are middle scale (top row of Figure 3.1) and large scale (bottom row of Figure 3.1), well conditioned (`ijcnn1`) and ill conditioned (`webspam` and `rcv1`), sparse (`rcv1` and `real-sim`) and dense (`epsilon`), under-parametrized (`phishing`, `mushrooms`, `ijcnn1`, `covtype`, `webspam`, `epsilon` and `real-sim`) and over-parametrized (`rcv1`).

**Pseudo-Huber function.** Recall the definition of the pseudo-Huber function used as the regularizer in our experiments in Figure 3.2: $R(w) = \sum_{i=1}^{d} R_i(w_i)$ with

$$R_i(w_i) = \delta^2 \left( \sqrt{1 + \left(\frac{w_i}{\delta}\right)^2} - 1 \right).$$

When $w_i$ is large, $R_i(w_i) \to \delta|w_i|$ for all $i = 1, \cdots, n$, i.e. $R(w)$ approximates L1 loss with a factor $\delta$; when $w_i$ is closed to zero, $R_i(w_i) \to \frac{1}{2}w_i^2$ for all $i = 1, \cdots, n$, i.e. $R(w)$ approximates L2 loss. This function can be served as a regularizer to promote the sparsity of the solution (Fountoulakis and Gondzio, 2016).

Besides, the pseudo-Huber is $\mathcal{C}^\infty$. The gradient of the pseudo-Huber is given by

$$\nabla R(w) = \left[ \frac{w_1}{\sqrt{1 + \left(\frac{w_1}{\delta}\right)^2}} \cdots \frac{w_d}{\sqrt{1 + \left(\frac{w_d}{\delta}\right)^2}} \right]^\top \in \mathbb{R}^d$$

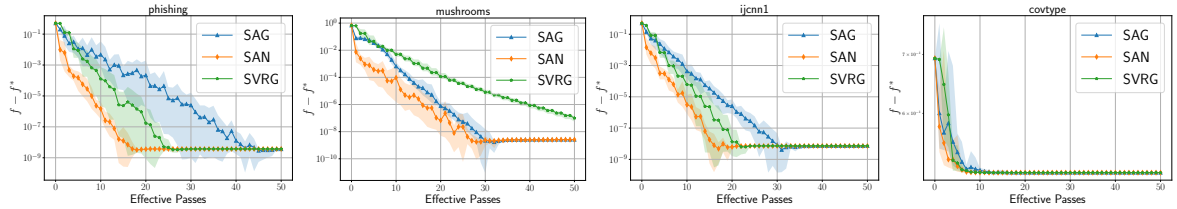


**Figure B.1** – Function sub-optimality of logistic regression with L2 regularization.

and the Hessian is given by

$$\nabla^2 R(w) \;=\; \mathbf{Diag}\left(\left(1+\left(\frac{w_1}{\delta}\right)^2\right)^{-3/2}, \cdots, \left(1+\left(\frac{w_d}{\delta}\right)^2\right)^{-3/2}\right) \;\leq\; \mathbf{I}_d.$$

Thus the pseudo-Huber is 1-smooth which is the same as L2 regularizer. Consequently, $L_{\max}$ for the pseudo-Huber regularized logistic regression is the same as the L2-regularized one.

### B.3.2 Function sub-optimality plots

The performance of an algorithm for solving a convex problem is usually done by measuring one of the following quantities: the solution gap $\|w^k - w^*\|$ where $w^*$ is the solution of the problem, the optimization gap $f(w^k) - \inf f$, and the stationarity gap $\|\nabla f(w^k)\|$. In Chapter 3 we choose to measure and compare performance of algorithms in terms of $\|\nabla f(w^k)\|$. The main reason for this is that the solution gap $\|w^k - w^*\|$ and the optimization gap $f(w^k) - \inf f$ both require to compute the solution of the problem to a high precision. While this is possible to do for small problems, it quickly becomes intractable for large problems (see Figure B.1 for `covtype`), which we want to address in Chapter 3 (see Table B.1 for more bigger datasets than `covtype`). The flat curves appeared in Figure B.1 after certain effective passes, especially for `covtype` dataset, are due to the imprecise computation of $\inf f$ from the solver `scipy.optimize.fmin_l_bfgs_b`. Indeed, the curves in Figure B.1 are in logarithmic scale. When $f(w^k) - \widehat{\inf} f < 0$ with $\widehat{\inf} f$ the tentative solution of the problem computed by the solver, it means that the solution $\widehat{\inf} f$ is imprecise, i.e. the solver performs worse than the tested algorithms. In this case, Figure B.1 plots $\left|f(w^k) - \widehat{\inf} f\right| = \widehat{\inf} f - f(w^k) > 0$ where the curves remain flat in logarithmic scale.

We argue that the quantity $\|\nabla f(w^k)\|^2$ is a fair and good proxy for the more classical optimization gap $f(w^k) - \inf f$. Our argument for this is twofold. First, we observe empirically on small problems (for which we can compute $\inf f$ with precision) that the curves for $\|\nabla f(w^k)\|^2$ and $f(w^k) - \inf f$ behave the same (see Figure B.1). Second, we verify theoretically that $\|\nabla f(w^k)\|^2$ and $f(w^k) - \inf f$ are of the same order. Indeed, Assumption 3.2 implies that $f$ is strongly convex on every compact. In particular, it verifies on every compact a Lojasiewicz

**Table B.2** – `covtype` dataset: grid search of $\pi$ and $\gamma$ for SAN

| $\pi$ \ $\gamma$ | 0.7 | 0.8 | 0.9 | 1.0 | 1.1 | 1.2 | 1.3 |
|---|---|---|---|---|---|---|---|
| $1/2n$ | 27 | 25 | 23 | 21 | 21 | 22 | 24 |
| $1/n$ | 26 | 26 | 25 | 22 | 22 | 23 | 24 |
| $10/n$ | 28 | 24 | 24 | 23 | 23 | 22 | 22 |
| $100/n$ | 28 | 26 | 23 | 23 | 22 | 23 | 24 |
| $1000/n$ | 28 | 26 | 27 | 27 | 25 | 24 | 26 |

inequality:

$$(\forall R > 0)(\exists \mu > 0)(\forall w \in \mathbb{B}(0, R)) \quad f(w) - \inf f \leq \frac{1}{2\mu}\|\nabla f(w)\|^2.$$

Moreover, $f$ is convex, so if we assume that $f$ has a $L$-Lipschitz gradient, we obtain the following inequality:

$$(\forall w \in \mathbb{R}^d) \quad \frac{1}{2L}\|\nabla f(w)\|^2 \leq f(w) - \inf f.$$

Note that this assumption is verified for the functions considered in our experiments (see Section B.3.1).

### B.3.3 Effect of hyperparameters

As we discussed in Section 3.3, SAN involves neither prior knowledge of the datasets (e.g., $L_{\max}$), nor the hyperparameter tuning, while both SAG (Schmidt et al., 2017) and SVRG (Johnson and Zhang, 2013) do. To support this conclusion, under different hyperparameters setting, we measure the performance of the given algorithm by monitoring the number of effective passes over the data required to reach below a threshold (e.g., $10^{-4}$ in our case) of $\|\nabla f\|$. We repeat this procedure 5 times and report the average results.

**Grid search for SAN.** SAN has two hyperparameters: the probability $\pi$ doing the averaging step in Algorithm 2 and the step size $\gamma$. We searched $\pi$ in a wide range

$$\pi \in \{\frac{1}{2n}, \frac{1}{n}, \frac{10}{n}, \frac{100}{n}, \frac{1000}{n}\};$$

as for $\gamma$, through our extensive experiments, we observed that SAN works consistently well when $\gamma$ is around one as we expected for second order methods, thus we tried

$$\gamma \in \{0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3\}.$$

**Table B.3** – `ijcnn1` dataset: grid search of $\pi$ and $\gamma$ for SAN

| $\pi$ \ $\gamma$ | 0.7 | 0.8 | 0.9 | 1.0 | 1.1 | 1.2 | 1.3 |
|---|---|---|---|---|---|---|---|
| $1/2n$ | 13 | 13 | 14 | 12 | 11 | 12 | 13 |
| $1/n$ | 14 | 13 | 12 | 12 | 12 | 12 | 13 |
| $10/n$ | 13 | 13 | 12 | 12 | 12 | 12 | 13 |
| $100/n$ | 13 | 11 | 12 | 13 | 11 | 12 | 14 |
| $1000/n$ | 16 | 13 | 13 | 13 | 14 | 14 | 14 |

**Table B.4** – Grid search of the step size $\gamma$ for SVRG on four datasets

| Datasets \ $\gamma$ | $\frac{1}{10L_{\max}}$ | $\frac{1}{5L_{\max}}$ | $\frac{1}{3L_{\max}}$ | $\frac{1}{2L_{\max}}$ | $\frac{1}{L_{\max}}$ | $\frac{2}{L_{\max}}$ | $\frac{5}{L_{\max}}$ |
|---|---|---|---|---|---|---|---|
| `covtype` | 44 | 24 | 18 | **14** | 20 | × | × |
| `ijcnn1` | 22 | 12 | **10** | **10** | 15 | 25 | × |
| `phishing` | 14 | 11 | **10** | 14 | 18 | 44 | × |
| `mushrooms` | × | × | 44 | 36 | 28 | **20** | 46 |

Table B.2 and B.3 show the grid search results on datasets `covtype` and `ijcnn1`. We see that the average effective data passes required to reach the threshold is stable. It means that SAN is not sensitive to these hyperparameters. This advantage allows us to use $\pi = \frac{1}{n+1}$ and $\gamma = 1.0$ as default choice in our experiments shown in Section 3.3.

**Grid search for SAG and SVRG.** Additionally we evaluated the effect of step size $\gamma$ which is a crucial hyperparameter for first order methods. Let $f_i$ be $L_i$-smooth for all $i \in \{1, \dots, n\}$ and $L_{\max} = \max_{i \in \{1,\dots,n\}} L_i$. As $\gamma = \frac{1}{L_{\max}}$ is thought as the rule of thumb choice in practice (Pedregosa et al., 2011) for SAG and SVRG, we searched over the values given by

$$\gamma \in \left\{ \frac{1}{10L_{\max}}, \frac{1}{5L_{\max}}, \frac{1}{3L_{\max}}, \frac{1}{2L_{\max}}, \frac{1}{L_{\max}}, \frac{2}{L_{\max}}, \frac{5}{L_{\max}} \right\}$$

on different datasets.

**Table B.5** – Grid search of the step size $\gamma$ for SAG on four datasets

| Datasets \ $\gamma$ | $\frac{1}{10L_{\max}}$ | $\frac{1}{5L_{\max}}$ | $\frac{1}{3L_{\max}}$ | $\frac{1}{2L_{\max}}$ | $\frac{1}{L_{\max}}$ | $\frac{2}{L_{\max}}$ | $\frac{5}{L_{\max}}$ |
|---|---|---|---|---|---|---|---|
| `covtype` | 21 | **19** | 23 | 24 | 40 | × | × |
| `ijcnn1` | **14** | 16 | 17 | 17 | 22 | 34 | × |
| `phishing` | **14** | 17 | 21 | 21 | 30 | 48 | × |
| `mushrooms` | × | 47 | 32 | 24 | **18** | 25 | × |

From our observations to Table B.4 and B.5 ,[1] we can draw the conclusions that compared to SAN, there is no universal step size choice for SAG and SVRG which gives a consistent good performance on different datasets. This point is one of our original motivations to develop a second order method that requires neither prior knowledge from datasets nor the hyperparameter tuning.

### B.3.4 Additional experiments for SANA, SNM and IQN applied for L2 logistic regression

We present some additional results of SANA, SNM (Kovalev et al., 2019) and IQN (Mokhtari et al., 2018) compared to SAN on L2 logistic regression scenario.

First, we show the results on middle size datasets, `phishing` and `mushrooms` in Figure B.2. On the one hand, in terms of effective passes of data, SANA has a similar performance as SAN despite the fact that SANA is unbiased and SAN is a biased estimate. Both methods are less efficient than SNM and IQN. Notice that the initialization process of SNM is expensive, as it requires a computation of the full Newton system. Such process is not counted into the effective passes. On the other hand, in terms of computational time, we observe that SAN does as well as IQN and SNM: SAN's cheap iteration cost compensates for its slower convergence rate. On the other hand, we observe for SANA that it is not competitive with respect to the other methods in terms of time taken. This is coherent in a regime where $d \ll n$ since SANA has a computation cost of $\mathcal{O}(nd)$ per iteration (see Table 3.1), while the cost for SAN and SNM, IQN is respectively $\mathcal{O}(d)$ and $\mathcal{O}(d^2)$, However, it shows that SANA is still a meaningful incremental second order method that satisfies our objective statement. This supports our general approach to design algorithms via function splitting.

In our second set of experiments, we compare those algorithms on large scale datasets. As shown in Figure B.3, we tested two datasets `webspam` and `epsilon`. As we discuss below, both SNM and IQN are limited in this case, while SAN is able to efficiently solve the problem. IQN is disqualified in this large scale setting, because its memory cost of $\mathcal{O}(nd^2)$ is prohibitive and makes it impossible to run on a laptop. This cost comes from the fact that IQN maintains and updates $n$ approximations of the hessians $\nabla^2 f_i(w^k)$, each of size $d^2$, and these matrices are not low-rank even for a GLM, preventing from using GLM implementation tricks (as it is the case for SNM, see Remark B.13 below). We also did not run SANA, since we already know that it performs similarly to SAN in terms of effective passes, but suffers from a cost per iteration scaling with $n$, which is too large here. It is possible to run SNM, but it is not efficient in terms of computational time due to its expensive cost per iteration. For the dataset `epsilon`, just after

[1] The symbol $\times$ in these tables means that the algorithm can not reach below the threshold $10^{-4}$ after 50 data passes.

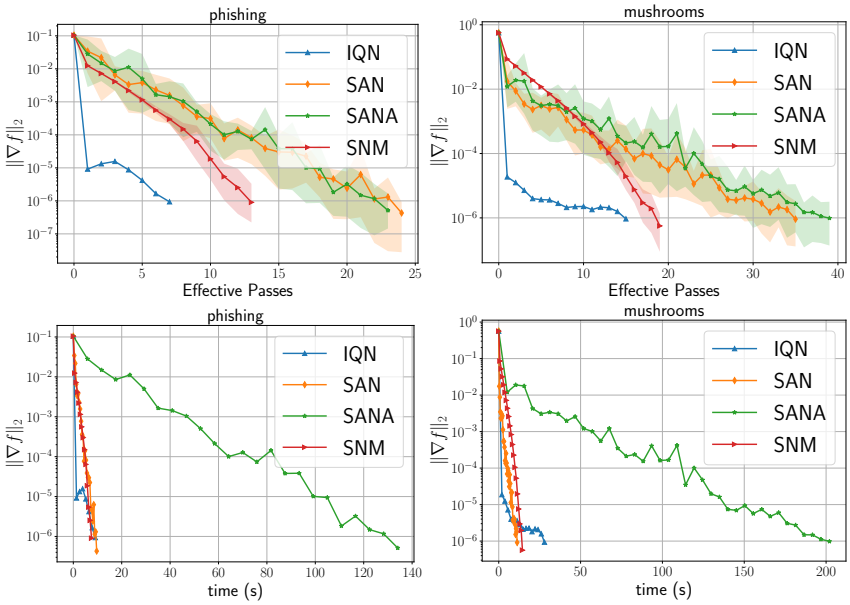


**Figure B.2** – L2-regularized logistic regression for SAN, SANA, SNM and IQN on middle size datasets. Top row is evaluated in terms of effective data passes and bottom row is evaluated in terms of computational time.

one pass over the data, the running time of SNM exceeded our maximum allowed time while at the same time SAN has run 25 data passes and reached a solution with a $10^{-6}$ precision.

Furthermore, note that we are running experiments in a setting which is favorable to SNM. Indeed, its cost per iteration $\mathcal{O}(d^2)$ is only valid when using $L2$ regularization. If we were to consider another separable regularizer, its cost per iteration would be $\mathcal{O}(d^3)$, making SNM infeasible for large dimensional problems. The next remark details those considerations about the complexity of SNM.

**Remark B.13** (On the cost of SNM). *The updates of SNM can be written in closed form as*

$$
\begin{aligned}
w^{k+1} &= \left(\frac{1}{n}\sum_{i=1}^{n}\nabla^2 f_i(\alpha_i^k)\right)^{-1}\left(\frac{1}{n}\sum_{i=1}^{n}\nabla^2 f_i(\alpha_i^k)\alpha_i^k - \nabla f_i(\alpha_i^k)\right),\\
\alpha_j^{k+1} &= w^{k+1},\\
\alpha_i^{k+1} &= \alpha_i^k \quad \textit{for } \; i \neq j,
\end{aligned}
$$

*where $w, \alpha_i \in \mathbb{R}^d$ for $i = 1, \cdots, n$ are variables defined in (3.5) using a variable splitting trick. The main cost of SNM is to update the following inverse matrix $\left(\frac{1}{n}\sum_{i=1}^{n}\nabla^2 f_i(\alpha_i^k)\right)^{-1}$ after updating a single $\alpha_j$.*

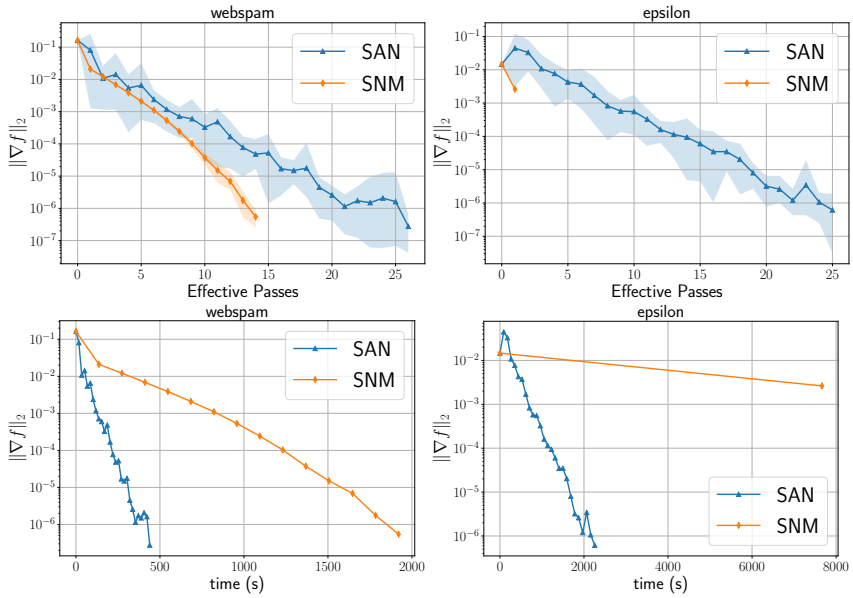


**Figure B.3** – L2-regularized logistic regression for SAN and SNM on large size datasets. Top row is evaluated in terms of effective data passes and bottom row is evaluated in terms of computational time.

*For L2-regularized GLMs, by using the Sherman-Morrison formula, the above term can be implemented efficiently in $\mathcal{O}(d^2)$ (See Algorithm 3 in Kovalev et al. (2019)), exploiting rank one updates of the matrix.*

*For other separable regularizers, such a formula is no longer available, as the perturbation becomes rank $d$ due to the diagonal Hessian of the regularizer derived by Lemma B.7 (1). The inversion of the matrix, therefore costs $\mathcal{O}(d^3)$ over all. Note that the memory cost is also impacted in this case: for general separable regularizers the memory cost will be $\mathcal{O}(nd+d^2)$, instead of $\mathcal{O}(n+d^2)$ as can be seen in Table 3.1 for L2-regularized GLMs.*

### B.3.5 SAN vs SAN without the variable metric

One of the main design features of SAN is that at every iteration we project our iterates onto an affine space with respect to a metric induced by the Hessian of one sampled function. One could ask whether this is worth it, given that it makes the theoretical analysis much more difficult. Let us consider again the problem introduced in (3.11) where the Hessian induced norm has been replaced by the L2 norm as following:

$$\alpha_j^{k+1}, w^{k+1} = \arg \min_{\alpha_j \in \mathbb{R}^d, w \in \mathbb{R}^d} \left\|\alpha_j - \alpha_j^k\right\|^2 + \left\|w - w^k\right\|^2 \quad \text{(B.15)}$$
$$\text{subject to } \nabla f_j(w^k) + \nabla^2 f_j(w^k)(w - w^k) = \alpha_j.$$

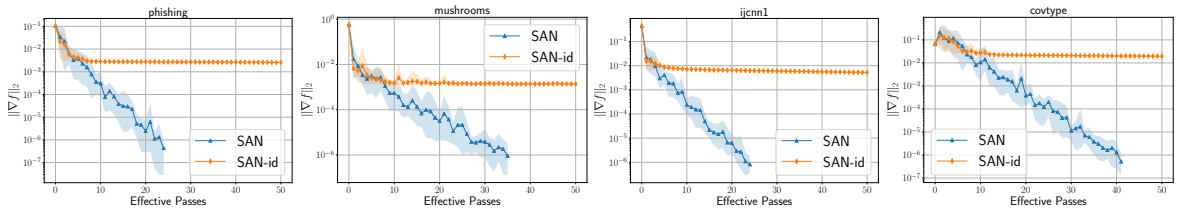


**Figure B.4** – L2-regularized logistic regression for SAN and SAN without the variable metric.

Using Lemma B.5, we can compute the closed form update of (B.15) (the details are left to readers):

$$\alpha_j^{k+1} = \alpha_j^k - \left(\mathbf{I} + (\nabla^2 f_j(w^k))^2\right)^{-1} \left(\alpha_j^k - \nabla f_j(w^k)\right), \tag{B.16}$$

$$w^{k+1} = w^k - \nabla^2 f_j(w^k) \left(\alpha_j^{k+1} - \alpha_j^k\right). \tag{B.17}$$

We call this algorithm *SAN-id*[2] for short. However, we observe from Figure B.4 that SAN-id only performs well at the early stage and stops converging to the optimum after the first few passes over the data. This motivated us to develop the version with the variable metric as introduced in the main text.

## B.4 SAN and SANA viewed as a sketched Newton Raphson method with variable metric

Here we provide a more detailed, step by step, introduction of the SAN and SANA methods. We also detail how SAN and SANA are particular instances of the Variable Metric Sketched Newton Raphson method introduced in the Section 3.4.

### B.4.1 A sketched Newton Raphson point of view

Here we clarify how the SAN and SANA methods are special cases of the sketched Newton Raphson method with a variable metric detailed in Section 3.4.

Let $x = \left[w\,;\alpha_1\,;\cdots\,;\alpha_n\right] \in \mathbb{R}^{(n+1)d}$ and $F : \mathbb{R}^{(n+1)d} \to \mathbb{R}^{(n+1)d}$ defined as

$$F(x) \overset{\text{def}}{=} \left[\tfrac{1}{n}\textstyle\sum \alpha_i\,;\nabla f_1(w) - \alpha_1\,;\cdots\,;\nabla f_n(w) - \alpha_n\right]. \tag{B.18}$$

[2]because this algorithm fits also in our SNRVM framework with $\mathbf{W}_k \equiv \mathbf{I}$ in (3.18).

Therefore $w^* \in \mathbb{R}^d$ is a minimizer of (3.1) if and only if there exists $x^* = [w^*; \alpha_1^*; \dots; \alpha_n^*] \in \mathbb{R}^{(n+1)d}$ such that $F(x^*) = 0$. The Jacobian $\nabla F(x)$ is given by

$$\nabla F(x) = \begin{bmatrix} 0 & \nabla^2 f_1(w) & \cdots & \nabla^2 f_n(w) \\ \frac{1}{n}\mathbf{I}_d & & & \\ \vdots & & -\mathbf{I}_{nd} & \\ \frac{1}{n}\mathbf{I}_d & & & \end{bmatrix} \in \mathbb{R}^{(n+1)d \times (n+1)d}, \tag{B.19}$$

To find a zero of the function $F$, one could use the damped Newton Raphson method

$$x^{k+1} = x^k - \gamma \nabla F(x^k)^\top \, {}^\dagger F(x^k), \quad \gamma \in (0,\, 1]. \tag{B.20}$$

This can be equivalently rewritten as a projection-and-relaxation step given by

$$\begin{cases} \bar{x}^{k+1} = \operatorname{argmin} \|x - x^k\|^2 \quad \text{subject to} \quad \nabla F(x^k)^\top (x - x^k) = -F(x^k), \\ x^{k+1} = (1-\gamma)x^k + \gamma \bar{x}^{k+1}. \end{cases} \tag{B.21}$$

Using the definition of our function $F$ in (3.6), we see that each iteration of the Newton Raphson method requires to project onto the following set of linear equations:

$$\begin{aligned} & \nabla F(x^k)^\top (x - x^k) = -F(x^k), \\ \Leftrightarrow \quad & \begin{cases} \frac{1}{n}\sum_{i=1}^n (\alpha_i - \alpha_i^t) = -\frac{1}{n}\sum_{i=1}^n \alpha_i^t, \\ \nabla^2 f_i(w^t)(w - w^t) - (\alpha_i - \alpha_i^t) = \alpha_i^t - \nabla f_i(w^t) \text{ for } i \in \{1, \dots, n\}, \end{cases} \\ \Leftrightarrow \quad & \begin{cases} \frac{1}{n}\sum_{i=1}^n \alpha_i = 0, \\ \nabla^2 f_i(w^t)(w - w^t) - \alpha_i = -\nabla f_i(w^t) \text{ for } i \in \{1, \dots, n\}. \end{cases} \end{aligned} \tag{B.22}$$

Projecting onto (B.22) is challenging for two reasons: first it accesses all of the data (every function $f_i$ is involved) and second it requires solving a large linear system.

One approach to circumvent this bottleneck is to *sketch* this linear system: at every iteration, instead of considering (B.22), we will project onto a random row compression of this system. Sketching can be for instance as simple as sampling one of the equations appearing in (B.22). In its more general form, a sketch corresponds to any linear transformation of the equations. In our context, this can be written as

$$\mathbf{S}^\top \nabla F(x^k)^\top (x - x^k) = -\mathbf{S}^\top F(x^k), \tag{B.23}$$

where $\mathbf{S} \in \mathbb{R}^{(n+1)d \times \tau}$ is called the sketching matrix, and its number of columns $\tau$ is typically small.

This idea is at the core of the Sketched Newton Raphson method (Yuan et al., 2022b), which aims at finding a zero of the function $F$ by iterating:

$$\begin{cases} \bar{x}^{k+1} = \operatorname{argmin} \|x - x^k\|^2 \quad \text{subject to} \quad \mathbf{S}_k^\top \nabla F(x^k)^\top (x - x^k) = -\mathbf{S}_k^\top F(x^k), \\ x^{k+1} = (1-\gamma)x^k + \gamma \bar{x}^{k+1}, \end{cases} \tag{B.24}$$

where $\mathbf{S}_k$ is a sketching matrix randomly sampled at each iteration with respect to some distribution.

As we detailed in Section 3.4, the algorithms proposed in Chapter 3 can be seen as particular instances of a *Variable Metric* Sketched Newton Raphson method. This more general framework allows, at every iteration, to project the previous iterate onto (B.23) with respect to some non-euclidean metric. The algorithm writes as follows:

$$\begin{cases} \bar{x}^{k+1} = \operatorname{argmin} \|x - x^k\|^2_{\mathbf{W}_k} \quad \text{subject to} \quad \mathbf{S}_k^\top \nabla F(x^k)^\top (x - x^k) = -\mathbf{S}_k^\top F(x^k), \\ x^{k+1} = (1-\gamma)x^k + \gamma \bar{x}^{k+1}. \end{cases} \tag{B.25}$$

Here, both $\mathbf{S}_k$ and $\mathbf{W}_k$ are randomly sampled with respect to a distribution which may depend on $x^k$. Besides, $\mathbf{W}_k$ is positive-definite. The closed form solution to (B.25) is given in (3.18), thanks to the following Lemma.

**Lemma B.14.** *If the iterates in (3.17) are well defined, then they are equivalent to (3.18).*

*Proof.* Let $x^{k+1}$ be the iterate defined by (3.17), where we assumed that the linear system $\mathbf{S}_k^\top \nabla F(x^k)^\top (x - x^k) = -\mathbf{S}_k^\top F(x^k)$ has a solution. Let us do a change of variable $u = x - x^k$, and write $\bar{x}^{k+1} = x^k + u^*$ where

$$u^* = \operatorname{argmin} \|u\|^2_{\mathbf{W}_k} \quad \text{subject to} \quad \mathbf{S}_k^\top \nabla F(x^k)^\top u = -\mathbf{S}_k^\top F(x^k).$$

We can call Lemma B.5 to obtain that

$$u^* = -\mathbf{W}_k^{-1} \nabla F(x^k) \mathbf{S}_k \left( \mathbf{S}_k^\top \nabla F(x^k)^\top \mathbf{W}_k^{-1} \nabla F(x^k) \mathbf{S}_k \right)^\dagger \mathbf{S}_k^\top F(x^k).$$

The claim follows after writing that $x^{k+1} = (1-\gamma)x^k + \gamma(x^k + u^*) = x^k + \gamma u^*$. □

### B.4.2 SAN is a particular case of SNRVM

Let us consider SAN, described in Algorithm 2, and rewrite it as an instance of the Variable Metric Sketched Newton Raphson method (B.25). Given a probability $\pi \in (0, 1)$, we define for all $x \in \mathbb{R}^{(n+1)d}$ a distribution $\mathcal{D}_x^{\text{SAN}}$ as follows: $(\mathbf{S}, \mathbf{W}) \sim \mathcal{D}_x^{\text{SAN}}$ means that

- with probability $\pi$ we have

$$\mathbf{S} = \begin{bmatrix} \mathbf{I}_d \\ \mathbf{0}_d \\ \vdots \\ \mathbf{0}_d \end{bmatrix} \quad \text{and} \quad \mathbf{W} = \mathbf{I}_{(n+1)d}, \tag{B.26}$$

- with probability $1-\pi$, we sample $j \in \{1, \cdots, n\}$ uniformly and set

$$\mathbf{S} = \begin{bmatrix} \mathbf{0}_d \\ \vdots \\ \mathbf{I}_d \\ \vdots \\ \mathbf{0}_d \end{bmatrix} \leftarrow j+1 \quad \text{and} \quad \mathbf{W} = \begin{bmatrix} \nabla^2 f_j(w) & & & \\ & \mathbf{I}_d & & \\ & & \ddots & \\ & & & \mathbf{I}_d \end{bmatrix}. \tag{B.27}$$

**Lemma B.15.** *Let $\pi \in (0,\, 1)$ and $\gamma \in (0,\, 1]$ a step size. Algorithm 2 (SAN) is equivalent to the Variable Metric Sketched Newton Raphson method (B.25) applied to the function $F$ defined in (3.6), where at each iteration $(\mathbf{S}_k, \mathbf{W}_k)$ is sampled with respect to $\mathcal{D}_{x^k}^{SAN}$, as defined in (B.26)-(B.27).*

*Proof.* Let us consider the Variable Metric Sketched Newton Raphson method described in this Lemma. We consider two cases, corresponding to the two classes of events described in (B.26) and (B.27).

Suppose that we are in the case (which holds with probability $\pi$) given by (B.26). In this case we have

$$\begin{aligned} \mathbf{S}_k^\top \nabla F(x^k)^\top (x - x^k) &\overset{(B.19)+(B.26)}{=} \begin{bmatrix} \mathbf{0}_d & \frac{1}{n}\mathbf{I}_d & \cdots & \frac{1}{n}\mathbf{I}_d \end{bmatrix} (x - x^k) \\ &= \frac{1}{n}\sum_{i=1}^n (\alpha_i - \alpha_i^k), \\ \mathbf{S}_k^\top F(x^k) &\overset{(3.6)+(B.26)}{=} \frac{1}{n}\sum_{i=1}^n \alpha_i^k. \end{aligned}$$

Those expressions mean that the linearized equation (B.23) is equivalent to $\frac{1}{n}\sum_{i=1}^n \alpha_i = 0$. So the update of the variables is exactly given by (B.5).

Let now $j$ be in $\{1, \ldots, n\}$ sampled uniformly, and suppose that we are in the case given by (B.27). We can then compute

$$\mathbf{S}_k^\top \nabla F(x^k)^\top (x - x^k) \overset{(B.19)+(B.27)}{=} \begin{bmatrix} \nabla^2 f_j(w^k) & \mathbf{0}_d & \cdots & -\mathbf{I}_d & \cdots & \mathbf{0}_d \end{bmatrix} (x - x^k)$$

$$\begin{aligned} &= && \nabla^2 f_j(w^k)(w - w^k) - (\alpha_j - \alpha_j^k), \\ \mathbf{S}_k^\top F(x^k) &\overset{(3.6)+(B.27)}{=} && \nabla f_j(w^k) - \alpha_j^k. \end{aligned}$$

Those expressions mean that the linearized equation (B.23) is equivalent to $\nabla^2 f_j(w^k)(w - w^k) - \alpha_j = -\nabla f_j(w^k)$. So the update of the variables is exactly given by (B.6). The conclusion follows Lemma B.2. □

### B.4.3 SANA is a particular case of SNRVM

Let us consider SANA, described in Algorithm 3, and rewrite it as an instance of the Variable Metric Sketched Newton Raphson method (B.25). We define for all $x \in \mathbb{R}^{(n+1)d}$ a distribution $\mathcal{D}_x^{\text{SANA}}$ as follows: $(\mathbf{S}, \mathbf{W}) \sim \mathcal{D}_x^{\text{SANA}}$ means that, with probability $1/n$ we sample $j \in \{1, \cdots, n\}$ and we have

$$\mathbf{S} = \begin{bmatrix} \mathbf{I}_d & \mathbf{0}_d \\ \mathbf{0}_d & \vdots \\ \vdots & \mathbf{I}_d \\ \vdots & \vdots \\ \mathbf{0}_d & \mathbf{0}_d \end{bmatrix} \leftarrow j+1 \quad \text{and} \quad \mathbf{W} = \begin{bmatrix} \nabla^2 f_j(w) & & & \\ & \mathbf{I}_d & & \\ & & \ddots & \\ & & & \mathbf{I}_d \end{bmatrix}. \tag{B.28}$$

**Lemma B.16.** *Let $\gamma \in (0, 1]$ be a step size. Algorithm 3 (SANA) is equivalent to the Variable Metric Sketched Newton Raphson method (B.25) applied to the function $F$ defined in (3.6), where at each iteration $(\mathbf{S}_k, \mathbf{W}_k)$ is sampled with respect to $\mathcal{D}_{x^k}^{SANA}$, as defined in (B.28).*

*Proof.* Let $k \in \mathbb{N}$, and suppose that we have sampled $j \in \{1, \ldots, n\}$ and $\mathbf{S}_k$ and $\mathbf{W}_k$ according to (B.28). Therefore,

$$\begin{aligned} \mathbf{S}_k^\top \nabla F(x^k)^\top (x - x^k) &\overset{(B.19)+(B.28)}{=} \begin{bmatrix} \mathbf{0}_d & \frac{1}{n}\mathbf{I}_d & \cdots & \cdots & \cdots & \frac{1}{n}\mathbf{I}_d \\ \nabla^2 f_j(w^k) & \mathbf{0}_d & \cdots & -\mathbf{I}_d & \cdots & \mathbf{0}_d \end{bmatrix} (x - x^k) \\ &= \begin{bmatrix} \frac{1}{n}\sum_{i=1}^n (\alpha_i - \alpha_i^k) \\ \nabla^2 f_j(w^k)(w - w^k) - (\alpha_j - \alpha_j^k) \end{bmatrix} \\ \mathbf{S}_k^\top F(x^k) &\overset{(3.6)+(B.28)}{=} \begin{bmatrix} \frac{1}{n}\sum_{i=1}^n \alpha_i^k \\ \nabla f_j(w^k) - \alpha_j^k \end{bmatrix}. \end{aligned}$$

Those expressions mean that the linearized equation (B.23) is equivalent to the two equations $\sum_{i=1}^{n} \alpha_i = 0$ and $\nabla^2 f_j(w^k)(w - w^k) - \alpha_j = -\nabla f_j(w^k)$. So the update of the variables is exactly given by (B.11). The conclusion follows Lemma B.4.

□

## B.5 Proofs for the results in Section 3.4, including Theorems 3.8 and 3.12

### B.5.1 Proof of Proposition 3.6

*Proof.* Let us start by showing that Assumption 3.4 is satisfied for SAN and SANA. The distribution $\mathcal{D}_x^{\text{SAN}}$ (resp. $\mathcal{D}_x^{\text{SANA}}$) defined in the Section B.4.2 (resp. Section B.4.3) is clearly finite and proper so long as $\pi \in (0, 1)$. It remains to compute $\mathbb{E}[\mathbf{S}\mathbf{S}^\top]$. We can see that it is a block-diagonal matrix $\mathbf{Diag}\left(\pi \mathbf{I}_d, \frac{1-\pi}{n}\mathbf{I}_d, \cdots, \frac{1-\pi}{n}\mathbf{I}_d\right)$ (resp. $\mathbf{Diag}\left(\mathbf{I}_d, \frac{1}{n}\mathbf{I}_d, \cdots, \frac{1}{n}\mathbf{I}_d\right)$), which is invertible since $\pi \in (0, 1)$.

Now let us turn on Assumption 3.5. To prove that $\nabla F(x)^\top \nabla F(x)$ is invertible, it is enough to show that $\nabla F(x)$ is injective. Let $x = (w\,; \alpha) \in \mathbb{R}^{d+dn}$, and let us first show that $\nabla F(x)$ is injective. Suppose there exists $\bar{x} = (\bar{w}\,; \bar{\alpha}) \in \mathbb{R}^{d+dn}$ such that $\nabla F(x)\bar{x} = 0$. Consequently from (B.19) we have that

$$\begin{aligned} \sum_{i=1}^{n} \nabla^2 f_i(w)\bar{\alpha}_i &= 0 \\ \frac{1}{n}\bar{w} &= \bar{\alpha}_i, \quad \text{for } i = 1, \ldots, n. \end{aligned} \tag{B.29}$$

Substituting out the $\bar{\alpha}_i$'s we have that

$$\nabla^2 f(w)\bar{w} = \frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(w)\bar{w} = 0.$$

Consequently, since $\nabla^2 f(w)$ is positive definite (recall Assumption 3.2), and in particular injective, we have that $\bar{w} = 0$. Thus it follows from (B.29) that $\bar{\alpha}_i = 0$ for $i = 1, \ldots, n$. This all shows that $\bar{x} = 0$, and concludes the proof that $\nabla F(x)$ is injective.

Furthermore, $\nabla F(x)$ is a square matrix, thus invertible. We have $F(x) \in \mathbf{Im}\left(\nabla F(x)^\top\right)$.

Finally $\nabla F(x)^\top \nabla F(x)$ is invertible since

$$\mathbf{Null}\left(\nabla F(x)^\top \nabla F(x)\right) = \mathbf{Null}\left(\nabla F(x)\right) = \{0\}.$$

□

### B.5.2 SNRVM is equivalent to minimizing a quadratic function over a random subspace

**Lemma B.17** (Lemma 10 in Gower et al. (2019a)). *For every matrix* $\mathbf{M}$ *and symmetric positive semi-definite matrix* $\mathbf{G}$ *such that* $\mathbf{Null}\left(\mathbf{G}\right) \subset \mathbf{Null}\left(\mathbf{M}\right)$, *we have that* $\mathbf{Null}\left(\mathbf{M}^\top\right) = \mathbf{Null}\left(\mathbf{M}\mathbf{G}\mathbf{M}^\top\right)$.

**Lemma B.18.** *Let Assumptions 3.4 and 3.5 hold. Then the iterates of SNRVM are equivalent to*

$$x^{k+1} = \operatorname{argmin}_{x\in\mathbb{R}^p} \ \hat{f}_k(x^k) + \left\langle \nabla \hat{f}_k(x^k), x - x^k \right\rangle + \frac{1}{2\gamma}\left\|x - x^k\right\|^2_{\mathbf{W}_k} \qquad \text{(B.30)}$$
$$\text{subject to } x \in x^k + \mathbf{Im}\left(\mathbf{W}_k^{-1}\nabla F(x^k)\mathbf{S}_k\right),$$

*where* $\hat{f}_k$ *is defined in* (3.19).

*Proof.* Start by observing that the problem in (B.30) is strongly convex, and therefore has a unique solution that we will note $x^*$. Let us prove that $x^*$ is exactly $x^{k+1}$ whose closed form expression is given in (3.18). For this, let $\tau$ be the number of columns for $\mathbf{S}_k$, and let $u \in \mathbb{R}^\tau$. We can then write that $x^* = x^k + \mathbf{W}_k^{-1}\nabla F(x^k)\mathbf{S}_k u^*$, where $u^*$ is any solution of the following unconstrained optimization problem:

$$u^* \in \operatorname{argmin}_{u\in\mathbb{R}^\tau} \left\langle \nabla \hat{f}_k(x^k), \mathbf{W}_k^{-1}\nabla F(x^k)\mathbf{S}_k u^* \right\rangle + \frac{1}{2\gamma}\left\|\mathbf{W}_k^{-1}\nabla F(x^k)\mathbf{S}_k u^*\right\|^2_{\mathbf{W}_k}.$$

Writing down the optimality conditions for this convex quadratic problem, we see that $u^*$ must verify:

$$\gamma\mathbf{S}_k^\top\nabla F(x^k)^\top\mathbf{W}_k^{-1}\nabla \hat{f}_k(x^k) + \mathbf{S}_k^\top\nabla F(x^k)^\top\mathbf{W}_k^{-1}\nabla F(x^k)\mathbf{S}_k u^*.$$

Let us choose the pseudo inverse solution of this linear system:

$$u^* = -\gamma\left(\mathbf{S}_k^\top\nabla F(x^k)^\top\mathbf{W}_k^{-1}\nabla F(x^k)\mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top\nabla F(x^k)^\top\mathbf{W}_k^{-1}\nabla \hat{f}_k(x^k). \qquad \text{(B.31)}$$

Using the definition of $\hat{f}_k$, we can write

$$\nabla \hat{f}_k(x^k) = \nabla F(x^k)\left(\nabla F(x^k)^\top\mathbf{W}_k^{-1}\nabla F(x^k)\right)^\dagger F(x^k). \qquad \text{(B.32)}$$

All we need to prove now is that

$$\nabla F(x^k)^\top \mathbf{W}_k^{-1} \nabla F(x^k) \left( \nabla F(x^k)^\top \mathbf{W}_k^{-1} \nabla F(x^k) \right)^\dagger F(x^k) = F(x^k). \tag{B.33}$$

To see why (B.33) is true, first notice that $\nabla F(x^k)^\top \mathbf{W}_k^{-1} \nabla F(x^k) \left( \nabla F(x^k)^\top \mathbf{W}^{-1} \nabla F(x^k) \right)^\dagger$ is the orthogonal projector onto the range of $\nabla F(x^k)^\top \mathbf{W}_k^{-1} \nabla F(x^k)$. Moreover, using the fact that $\mathbf{W}_k^{-1}$ is injective together with Lemma B.17, we can write that

$$\begin{aligned} \mathbf{Im}\left( \nabla F(x^k)^\top \mathbf{W}_k^{-1} \nabla F(x^k) \right) &= \left( \mathbf{Null}\left( F(x^k)^\top \mathbf{W}_k^{-1} \nabla F(x^k) \right) \right)^\perp \\ &= \left( \mathbf{Null}\left( \nabla F(x^k) \right) \right)^\perp \\ &= \mathbf{Im}\left( \nabla F(x^k)^\top \right). \end{aligned}$$

Since we know from Assumption 3.5 that $\nabla F(x^k)^\top$ is surjective, and so that $F(x^k)$ belongs in the range of $\nabla F(x^k)^\top$, we deduce that (B.33) is true. We can now inject (B.33) into (B.31), and obtain finally that

$$\begin{aligned} x^* &= x^k + \mathbf{W}_k^{-1} \nabla F(x^k) \mathbf{S}_k u^* \\ &= x^k - \gamma \mathbf{W}_k^{-1} \nabla F(x^k) \mathbf{S}_k \left( \mathbf{S}_k^\top \nabla F(x^k)^\top \mathbf{W}_k^{-1} \nabla F(x^k) \mathbf{S}_k \right)^\dagger \mathbf{S}_k^\top F(x^k), \end{aligned}$$

which is exactly (3.18). □

### B.5.3 About $\rho$ in Theorem 3.8

**Lemma B.19.** *If $A$, $B$ are two symmetric positive semi-definite matrices, then*

$$\mathbf{Null}\,(A+B) \;=\; \mathbf{Null}\,(A) \cap \mathbf{Null}\,(B)\,.$$

*Proof.* If $x \in \mathbf{Null}\,(A) \cap \mathbf{Null}\,(B)$ then it is trivial to see that $x \in \mathbf{Null}\,(A+B)$. If $x \in \mathbf{Null}\,(A+B)$, then

$$0 = \langle (A+B)x, x \rangle = \langle Ax, x \rangle + \langle Bx, x \rangle,$$

where by positive semi-definiteness we have $\langle Ax, x \rangle \geq 0$ and $\langle Bx, x \rangle \geq 0$. The sum of nonnegative numbers being nonegative, we deduce that

$$\langle Ax, x \rangle = \langle Bx, x \rangle = 0.$$

Since $\langle Ax, x\rangle = 0$, we deduce from the fact that $A$ is symmetric that $Ax = 0$. Similarly, $Bx = 0$, which concludes the proof. □

The following Lemma will be needed in the proof of Theorem 3.8.

**Lemma B.20.** *Recall the definition of* $\mathbf{H}(x)$ *given by*

$$\mathbf{H}(x) \overset{def}{=} \mathbb{E}\left[\mathbf{S}\left(\mathbf{S}^\top \nabla F(x)^\top \mathbf{W}^{-1} \nabla F(x) \mathbf{S}\right)^\dagger \mathbf{S}^\top\right], \tag{B.34}$$

*If Assumption 3.4 and 3.5 hold, then* $\mathbf{H}(x)$ *is invertible. Moreover, for every symmetric positive definite matrix* $\mathbf{W}$ *and* $x \in \mathbb{R}^p$ *we have that*

$$\min_{v \in \mathbf{Im}\left(\mathbf{W}^{-1/2}\nabla F(x)\right)\setminus\{0\}} \frac{\left\langle \mathbf{W}^{-1/2}\nabla F(x)\mathbf{H}(x)\nabla F(x)^\top \mathbf{W}^{-1/2} v, v\right\rangle}{\|v\|^2} \tag{B.35}$$

*is exactly the smallest positive eigenvalue of* $\mathbf{W}^{-1/2}\nabla F(x)\mathbf{H}(x)\nabla F(x)^\top \mathbf{W}^{-1/2}$.

*Proof.* Let $x \in \mathbb{R}^p$ and $(\mathbf{S}, \mathbf{W}) \sim \mathcal{D}_x$. Let $\mathbf{G} = \nabla F(x)^\top \mathbf{W}^{-1} \nabla F(x)$ which is symmetric positive semi-definite. Since $\nabla F(x)^\top \nabla F(x)$ and $\mathbf{W}$ are invertible we have that $\mathbf{G}$ is invertible. Consequently $\mathbf{Null}\left(\mathbf{G}\right) = \{0\} \subset \mathbf{Null}\left(\mathbf{S}^\top\right)$. Thus by Lemma B.17 (with $\mathbf{M} = \mathbf{S}^\top$) we have that

$$\mathbf{Null}\left(\left(\mathbf{S}^\top \nabla F(x)^\top \mathbf{W}^{-1} \nabla F(x) \mathbf{S}\right)^\dagger\right) = \mathbf{Null}\left(\mathbf{S}^\top \nabla F(x)^\top \mathbf{W}^{-1} \nabla F(x) \mathbf{S}\right) = \mathbf{Null}\left(\mathbf{S}\right).$$

Using Lemma B.17 once again with $\mathbf{G} = \left(\mathbf{S}^\top \nabla F(x)^\top \mathbf{W}^{-1} \nabla F(x) \mathbf{S}\right)^\dagger$ and $\mathbf{M} = \mathbf{S}$, we have that

$$\mathbf{Null}\Big(\underbrace{\mathbf{S}\left(\mathbf{S}^\top \nabla F(x)^\top \mathbf{W}^{-1} \nabla F(x) \mathbf{S}\right)^\dagger \mathbf{S}^\top}_{\overset{\text{def}}{=}\mathbf{H}_{\mathbf{S},\mathbf{W}}(x)}\Big) = \mathbf{Null}\left(\mathbf{S}^\top\right) = \mathbf{Null}\left(\mathbf{S}\mathbf{S}^\top\right). \tag{B.36}$$

Observe that with our notations and from Assumption 3.4,

$$\mathbf{H}(x) = \mathbb{E}_{\mathbf{S},\mathbf{W}\sim\mathcal{D}_x}\left[\mathbf{H}_{\mathbf{S},\mathbf{W}}(x)\right] = \sum_{i=1}^r p_i \mathbf{H}_{\mathbf{S}_i(x),\mathbf{W}_i(x)}(x).$$

As $\mathbf{H}_{\mathbf{S}_i(x),\mathbf{W}_i(x)}(x)$ is symmetric positive semi-definite, we can use Lemma B.19 to write

$$\mathbf{Null}\left(\mathbf{H}(x)\right) = \mathbf{Null}\left(\sum_{i=1}^r p_i \mathbf{H}_{\mathbf{S}_i(x),\mathbf{W}_i(x)}(x)\right) \quad = \quad \bigcap_{i=1}^r \mathbf{Null}\left(\mathbf{H}_{\mathbf{S}_i(x),\mathbf{W}_i(x)}(x)\right)$$

$$\overset{(B.36)}{=} \bigcap_{i=1}^{r} \mathbf{Null}\left(\mathbf{S}_i(x)\mathbf{S}_i(x)^\top\right)$$
$$= \mathbf{Null}\left(\mathbb{E}_{\mathbf{S}\sim\mathcal{D}_x}\left[\mathbf{S}\mathbf{S}^\top\right]\right) = \{0\}$$

This means that $\mathbf{H}(x)$ is invertible for all $x \in \mathbb{R}^p$.

Now, take any $x \in \mathbb{R}^p$, and a symmetric positive definite matrix $\mathbf{W}$. Then the matrix

$$\mathbf{W}^{-1/2}\nabla F(x)\mathbf{H}(x)\nabla F(x)^\top\mathbf{W}^{-1/2}$$

is symmetric, semi-definite positive. Since $\mathbf{H}(x)$ and $\mathbf{W}$ are invertible, we can apply Lemma B.17 again to obtain

$$\mathbf{Null}\left(\nabla F(x)^\top\mathbf{W}^{-1/2}\right) = \mathbf{Null}\left(\mathbf{W}^{-1/2}\nabla F(x)\mathbf{H}(x)\nabla F(x)^\top\mathbf{W}^{-1/2}\right). \tag{B.37}$$

Consequently

$$\begin{aligned}\mathbf{Im}\left(\mathbf{W}^{-1/2}\nabla F(x)\right) &= \left(\mathbf{Null}\left(\nabla F(x)^\top\mathbf{W}^{-1/2}\right)\right)^\perp \\ &\overset{(B.37)}{=} \left(\mathbf{Null}\left(\mathbf{W}^{-1/2}\nabla F(x)\mathbf{H}(x)\nabla F(x)^\top\mathbf{W}^{-1/2}\right)\right)^\perp. \end{aligned} \tag{B.38}$$

Therefore, we conclude that (B.35) is equal to

$$\min_{v\in\left(\mathbf{Null}\left(\mathbf{W}^{-1/2}\nabla F(x)\mathbf{H}(x)\nabla F(x)^\top\mathbf{W}^{-1/2}\right)\right)^\perp\setminus\{0\}} \frac{\left\langle\mathbf{W}^{-1/2}\nabla F(x)\mathbf{H}(x)\nabla F(x)^\top\mathbf{W}^{-1/2}v, v\right\rangle}{\|v\|^2}$$
$$= \lambda_{\min}^{+}\left(\mathbf{W}^{-1/2}\nabla F(x)\mathbf{H}(x)\nabla F(x)^\top\mathbf{W}^{-1/2}\right) > 0.$$

□

### B.5.4 Proof of Theorem 3.8

*Proof.* Let $k \in \mathbb{N}$. In this proof, we will write $\nabla F_k$ as a shorthand for $\nabla F(x^k)$, and we introduce the notation $\nabla^{\mathbf{W}} F_k \overset{\text{def}}{=} \mathbf{W}_k^{-1/2}\nabla F(x^k)$. First we aim to establish a relationship between $\hat{f}_k(x^k) = \left\|F(x^k)\right\|^2_{(\nabla F_k^\top\mathbf{W}_k^{-1}\nabla F_k)^\dagger}$ and $\left\|F(x^k)\right\|^2_{\mathbf{H}(x^k)}$. Observe that Assumption 3.5 allows us to write that

$$(\forall x \in \mathbb{R}^p) \quad F(x) = \nabla F(x)^\top\mathbf{W}_k^{-1/2}(\nabla F(x)^\top\mathbf{W}_k^{-1/2})^\dagger F(x). \tag{B.39}$$

This is due to the fact that $\nabla F(x)^\top \mathbf{W}_k^{-1/2}(\nabla F(x)^\top \mathbf{W}_k^{-1/2})^\dagger$ is the projection matrix onto $\mathbf{Im}\left(\nabla F(x)^\top \mathbf{W}_k^{-1/2}\right)$, where $\mathbf{W}_k^{-1/2}$ is surjective, meaning that

$$\mathbf{Im}\left(\nabla F(x)^\top\right) = \mathbf{Im}\left(\nabla F(x)^\top \mathbf{W}_k^{-1/2}\right).$$

From (B.39) we have that

$$\begin{aligned}
\left\|F(x^k)\right\|^2_{\mathbf{H}(x^k)} &= \left\langle F(x^k), \mathbf{H}(x^k)F(x^k)\right\rangle \\
&\overset{(B.39)}{=} \left\langle \nabla^{\mathbf{W}} F_k^\top (\nabla^{\mathbf{W}} F_k^\top)^\dagger F(x^k), \mathbf{H}(x^k)\nabla^{\mathbf{W}} F_k^\top (\nabla^{\mathbf{W}} F_k^\top)^\dagger F(x^k)\right\rangle \\
&= \left\langle (\nabla^{\mathbf{W}} F_k^\top)^\dagger F(x^k), \nabla^{\mathbf{W}} F_k \mathbf{H}(x^k)\nabla^{\mathbf{W}} F_k^\top \left((\nabla^{\mathbf{W}} F_k^\top)^\dagger F(x^k)\right)\right\rangle \\
&\overset{\text{Lemma B.20}}{\geq} \rho \left\|(\nabla^{\mathbf{W}} F_k^\top)^\dagger F(x^k)\right\|^2 \qquad \text{(B.40)} \\
&= \rho \left\|F(x^k)\right\|^2_{(\nabla^{\mathbf{W}} F_k^\top \nabla^{\mathbf{W}} F_k)^\dagger} \qquad \text{(B.41)} \\
&\overset{(3.19)}{=} 2\rho \hat{f}_k(x^k), \qquad \text{(B.42)}
\end{aligned}$$

where in (B.40) we used that $\mathbf{Im}\left((\nabla^{\mathbf{W}} F_k^\top)^\dagger\right) = \mathbf{Im}\left(\nabla^{\mathbf{W}} F_k\right)$ together with Lemma B.20, and in (B.41) we used that $(\mathbf{M})^{\dagger\top}(\mathbf{M})^\dagger = (\mathbf{M}^\top)^\dagger(\mathbf{M})^\dagger = (\mathbf{M}\mathbf{M}^\top)^\dagger$ for every matrix $\mathbf{M}$. Now, we turn onto the study the term $\hat{f}_k(x^k)$. Compute its gradient with respect to the metric induced by $\mathbf{W}_k$ at $x^k$:

$$\begin{aligned}
\nabla^{\mathbf{W}_k} \hat{f}_k(x^k) &= \mathbf{W}_k^{-1}\nabla F_k(\nabla F_k^\top \mathbf{W}_k^{-1}\nabla F_k)^\dagger F(x^k) \\
&= \mathbf{W}_k^{-1/2}(\nabla F_k^\top \mathbf{W}_k^{-1/2})^\dagger F(x^k). \qquad \text{(B.43)}
\end{aligned}$$

This, together with the Assumption 3.7, allows us to write that

$$\begin{aligned}
& \hat{f}_{k+1}(x^{k+1}) \\
\leq\ & \hat{f}_k(x^k) + \left\langle \nabla \hat{f}_k(x^k), x^{k+1} - x^k\right\rangle + \frac{L}{2}\left\|x^{k+1} - x^k\right\|^2_{\mathbf{W}_k} \\
\overset{(3.18)+(B.43)}{=}\ & \hat{f}_k(x^k) - \gamma \left\langle n^{\mathbf{W}_k}(x^k), \mathbf{W}_k^{-1}\nabla F_k \mathbf{S}_k \left(\mathbf{S}_k^\top \nabla F_k^\top \mathbf{W}_k^{-1}\nabla F_k \mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top F(x^k)\right\rangle_{\mathbf{W}_k} \\
& + \frac{\gamma^2 L}{2}\left\|\mathbf{W}_k^{-1}\nabla F_k \mathbf{S}_k \left(\mathbf{S}_k^\top \nabla F_k^\top \mathbf{W}_k^{-1}\nabla F_k \mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top F(x^k)\right\|^2_{\mathbf{W}_k} \\
\overset{(B.39)}{=}\ & \hat{f}_k(x^k) - \gamma \left\langle F(x^k), \mathbf{S}_k \left(\mathbf{S}_k^\top \nabla F_k^\top \mathbf{W}_k^{-1}\nabla F_k \mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top F(x^k)\right\rangle \\
& + \frac{\gamma^2 L}{2}\left\|\mathbf{W}_k^{-1}\nabla F_k \mathbf{S}_k \left(\mathbf{S}_k^\top \nabla F_k^\top \mathbf{W}_k^{-1}\nabla F_k \mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top F(x^k)\right\|^2_{\mathbf{W}_k} \\
=\ & \hat{f}_k(x^k) - \gamma\left(1 - \frac{\gamma L}{2}\right)\left\|F(x^k)\right\|^2_{\mathbf{S}_k(\mathbf{S}_k^\top \nabla F_k^\top \mathbf{W}_k^{-1}\nabla F_k \mathbf{S}_k)^\dagger \mathbf{S}_k^\top}
\end{aligned}$$

$$\overset{\gamma=1/L}{=} \quad \hat{f}_k(x^k) - \frac{\gamma}{2}\left\|F(x^k)\right\|^2_{\mathbf{S}_k(\mathbf{S}_k^\top \nabla F_k^\top \mathbf{W}_k^{-1} \nabla F_k \mathbf{S}_k)^\dagger \mathbf{S}_k^\top} \tag{B.44}$$

where in (B.44) we use the identity $\mathbf{M}^\dagger \mathbf{M} \mathbf{M}^\dagger = \mathbf{M}^\dagger$ with $\mathbf{M} = \mathbf{S}_k^\top \nabla F_k^\top \mathbf{W}_k^{-1} \nabla F_k \mathbf{S}_k$. Taking the expectation conditioned on $x_k$ in the inequality (B.44) gives

$$\begin{aligned}\mathbb{E}\left[\hat{f}_{k+1}(x^{k+1}) \mid x^k\right] &\leq \mathbb{E}\left[\hat{f}_k(x^k) \mid x^k\right] - \frac{\gamma}{2}\left\|F(x^k)\right\|^2_{\mathbf{H}(x^k)} \\ &\leq \mathbb{E}\left[\hat{f}_k(x^k) \mid x^k\right] - \rho\gamma \hat{f}_k(x^k).\end{aligned}$$

Taking full expectation and expanding the recurrence gives finally

$$\mathbb{E}\left[\hat{f}_{k+1}(x^{k+1})\right] \leq (1-\rho\gamma)\mathbb{E}\left[\hat{f}_k(x^k)\right].$$

□

### B.5.5 SNRVM for solving linear systems

Here we consider the simplified case in which our objective function (3.1) is a quadratic function. In this case, the stationarity condition (3.5) is a linear system. To simplify the notation, let us denote in this section the resulting linear system as

$$\mathbf{A}x = b, \quad \text{where } \mathbf{A} \in \mathcal{M}_p(\mathbb{R}),\ p = d(n+1). \tag{B.45}$$

In other words, our nonlinear map is given by $F(x) = \mathbf{A}x - b$. Because $\nabla F(x) = \mathbf{A}^\top$, where $\mathbf{A}$ is a square matrix, we see that the assumptions on $F$ in Assumption 3.5 are verified if and only if $\mathbf{A}$ is invertible.

In this setting the SNR method (3.16) is known as the sketch-and-project method (Gower and Richtárik, 2015b). The sketch-and-project method has been shown to converge linearly at a fast rate (Gower and Richtárik, 2015b; Gower and Richtárik, 2015a). Thus this quadratic case serves as a good sanity check to verify if our rate of convergence in Theorem 3.8 recovers the well known fast linear rate of the sketch-and-project method. This is precisely what we investigate in the next lemma. It only remains to reformulate Assumption 3.7, which we do in the following lemma.

**Lemma B.21.** *If $\{\mathbf{W}_k\}_{k\in\mathbb{N}}$ is a sequence of invertible matrices such that*

$$\mathbf{W}_{k+1} \preceq \mathbf{W}_k, \tag{B.46}$$

*and $\mathbf{A}$ is invertible, then Assumption 3.7 holds with $L = 1$.*

*Proof.* Using the fact that $F(x) = \mathbf{A}x - b$ and $\nabla F(x) = \mathbf{A}^\top$, we can rewrite definition (3.19) as

$$\hat{f}_k(x) \;=\; \frac{1}{2}\,\|\mathbf{A}x - b\|^2_{(\mathbf{A}\mathbf{W}_k^{-1}\mathbf{A}^\top)^\dagger}\,. \tag{B.47}$$

Now, since $\mathbf{A}$ is invertible, we have $(\mathbf{A}\mathbf{W}_k^{-1}\mathbf{A}^\top)^\dagger = \mathbf{A}^{\top -1}\mathbf{W}_k\mathbf{A}^{-1}$. So, if $x^*$ is the unique solution to (B.45), then we obtain

$$\hat{f}_k(x) \;=\; \frac{1}{2}\,\|x - x^*\|^2_{\mathbf{W}_k}\,. \tag{B.48}$$

Using Assumption B.46 together with the fact that $\hat{f}_k$ is quadratic to conclude that

$$\begin{aligned}
\hat{f}_{k+1}(x^{k+1}) &\overset{(B.46)}{\leq} \hat{f}_k(x^{k+1}) \\
&= \hat{f}_k(x^k) + \langle \nabla \hat{f}_k(x^k), x^{k+1} - x^k\rangle + \frac{1}{2}\|x^{k+1} - x^k\|^2_{\nabla^2 \hat{f}_k(x^k)} \\
&= \hat{f}_k(x^k) + \langle \nabla \hat{f}_k(x^k), x^{k+1} - x^k\rangle + \frac{1}{2}\|x^{k+1} - x^k\|^2_{\mathbf{W}_k}.
\end{aligned}$$

□

**Proposition B.22.** *Let $\mathbf{A} \in \mathcal{M}_p(\mathbb{R})$ be invertible, $b \in \mathbb{R}^p$, and $x^*$ be the solution to (B.45). Let $(x^k)_{k\in\mathbb{N}}$ be a sequence generated from SNRVM (3.18), with $F(x) = \mathbf{A}x - b$, and $\gamma = 1$. We assume that, at every iteration $k \in \mathbb{N}$, the matrices $(\mathbf{S}_k, \mathbf{W}_k)$ are sampled from a finite proper distribution (see Assumption 3.4) such that for all $x$, $\mathcal{D}_x$ is independant of $x$, that $\mathbb{E}_{\mathcal{D}_x}[\mathbf{S}\mathbf{S}^\top]$ is invertible and $\mathbf{W}_k$ is constant and equal to some invertible matrix $\mathbf{W} \in \mathcal{M}_p(\mathbb{R})$.*

*Let*

$$\rho \overset{def}{=} \lambda_{\min}\left(\mathbf{W}^{-1/2}\mathbf{A}^\top \mathbb{E}_{\mathcal{D}_{x^0}}[\mathbf{S}(\mathbf{S}^\top\mathbf{A}\mathbf{W}^{-1}\mathbf{A}^\top\mathbf{S})^\dagger\mathbf{S}^\top]\mathbf{A}\mathbf{W}^{-1/2}\right).$$

*It follows that $\rho \in (0, 1)$, and*

$$\mathbb{E}\left[\|x^k - x^*\|^2_{\mathbf{W}}\right] \;\leq\; (1-\rho)\|x^2 - x^*\|^2_{\mathbf{W}}. \tag{B.49}$$

*Proof.* We are going to apply the result in Theorem 3.8, so we start by checking its assumptions. First, our assumptions on the sampling ensure that Assumption 3.4 is verified. Second, as discussed earlier in this section, the fact that $\mathbf{A}$ is invertible ensures that Assumption 3.5 holds true. Third, our assumption that $\mathbf{W}_k \equiv \mathbf{W}$ together with Lemma B.21 tells us that Assumption 3.7 holds with $L = 1$, meaning that we take a stepsize $\gamma = 1$. Let $\mathbf{H} \overset{\text{def}}{=} \mathbb{E}_{\mathcal{D}_{x^0}}[\mathbf{S}(\mathbf{S}^\top\mathbf{A}\mathbf{W}^{-1}\mathbf{A}^\top\mathbf{S})^\dagger\mathbf{S}^\top]$. Note that this matrix is independant of $k$, because we assumed the distribution $\mathcal{D}_x$ to be independant of $x$. We also know that $\mathbf{H}$ is invertible, thanks to Lemma B.20. Therefore, $\rho = \lambda_{min}\left(\mathbf{W}^{-1/2}\mathbf{A}^\top\mathbf{H}\mathbf{A}\mathbf{W}^{-1/2}\right) > 0$.

To prove that $\rho \leq 1$, observe that

$$\begin{aligned}
&\mathbf{W}^{-1/2}\mathbf{A}^\top \mathbb{E}[\mathbf{S}(\mathbf{S}^\top \mathbf{A}\mathbf{W}^{-1}\mathbf{A}^\top \mathbf{S})^\dagger \mathbf{S}^\top]\mathbf{A}\mathbf{W}^{-1/2} \\
&= \mathbb{E}[\mathbf{W}^{-1/2}\mathbf{A}^\top \mathbf{S}(\mathbf{S}^\top \mathbf{A}\mathbf{W}^{-1/2}\mathbf{W}^{-1/2}\mathbf{A}^\top \mathbf{S})^\dagger \mathbf{S}^\top \mathbf{A}\mathbf{W}^{-1/2}] \\
&= \mathbb{E}[(\mathbf{S}^\top \mathbf{A}\mathbf{W}^{-1/2})^\top ((\mathbf{S}^\top \mathbf{A}\mathbf{W}^{-1/2})(\mathbf{S}^\top \mathbf{A}\mathbf{W}^{-1/2}))^\dagger (\mathbf{S}^\top \mathbf{A}\mathbf{W}^{-1/2})] \\
&= \mathbb{E}[(\mathbf{S}^\top \mathbf{A}\mathbf{W}^{-1/2})^\dagger (\mathbf{S}^\top \mathbf{A}\mathbf{W}^{-1/2})],
\end{aligned}$$

where $(\mathbf{S}^\top \mathbf{A}\mathbf{W}^{-1/2})^\dagger (\mathbf{S}^\top \mathbf{A}\mathbf{W}^{-1/2})$ is the orthogonal projection onto the range of $\mathbf{W}^{-1/2}\mathbf{A}^\top \mathbf{S}$. Consequently $\lambda_{max}((\mathbf{S}^\top \mathbf{A}\mathbf{W}^{-1/2})^\dagger (\mathbf{S}^\top \mathbf{A}\mathbf{W}^{-1/2})) \leq 1$, and from Jensen's inequality, we deduce that the eigenvalues of its expectation also are in $(0, 1]$. Whence $\rho \in (0, 1]$.

To conclude the proof, see that under our assumptions, the quantity $\rho(x)$ defined in Theorem 3.8 is independent of $x$, and equal to $\rho$. We have verified all the assumptions needed to call Theorem 3.8, which proves the claim. □

The rate of convergence given in (B.49) is exactly the well known linear rate of convergence given in Theorem 4.6 in Gower and Richtárik (2015a). For example, if $\mathbf{A}$ is symmetric positive definite, and we can set $\mathbf{W}_k \equiv \mathbf{A}$ and sample the sketching matrix $\mathbf{S} \in \mathbb{R}^{p\times 1}$ according to

$$\Pr \mathbf{S} = e_i \;=\; \frac{\mathbf{A}_{ii}}{\mathbf{Trace}\,(\mathbf{A})}, \quad \text{for } i = 1, \ldots, m.^{3}$$

With this choice of sketch and metric, the resulting method (3.18) is known as coordinate descent (Leventhal and Lewis, 2010; Gower and Richtárik, 2015b). In this case, our resulting rate in (B.49) is controlled by

$$\rho = \lambda_{\min}\left(\mathbf{A}^{1/2}\mathbb{E}\left[e_i\,(\mathbf{A}_{ii})^\dagger\, e_i^\top\right]\mathbf{A}^{1/2}\right) = \frac{\lambda_{\min}\,(\mathbf{A})}{\mathbf{Trace}\,(\mathbf{A})},$$

which is exactly the celebrated linear convergence rate of coordinate descent first given in Leventhal and Lewis (2010).

Proposition B.22 shows that our main convergence theory in Theorem 3.8 is *tight* in this quadratic setting. That is, when specialized to a linear mapping $F(x)$ and a fixed metric $\mathbf{W} \equiv \mathbf{W}_k$, our Theorem 3.8 recovers the best known convergences results as a special case.

### B.5.6 Proof of Theorem 3.10

[3] Here $e_i \in \mathbb{R}^m$ is the $i$-th unit coordinate vector and $\mathbf{Trace}\,(\mathbf{A}) = \sum_{i=1}^m \mathbf{A}_{ii}$ is the trace of $\mathbf{A}$.

**Lemma B.23.** *Let Assumption 3.9 hold. Let $x \in \Omega$, let $(\hat{\mathbf{S}}, \hat{\mathbf{W}})$ be in the domain of $\mathcal{D}_x$, and consider $\mathbf{A} := \hat{\mathbf{W}}^{-1/2}\nabla F(x)\mathbf{H}(x)\nabla F(x)^\top \hat{\mathbf{W}}^{-1/2}$, where $\mathbf{H}(x)$ is defined in (B.34). Then*

$$\lambda_{min}(\mathbf{A}) \geq \frac{\mu_{\nabla F}^2}{L_{\nabla F}^2}\frac{\mu_W}{L_W}\frac{\bar{\mu}_S}{L_S}.$$

*Proof.* Let us write $\mathbf{H} := \mathbf{H}(x)$, $J := \nabla F(x)$ and $U := J^\top \hat{\mathbf{W}}^{-1/2}$, so that $\mathbf{A} = U^\top \mathbf{H} U$. Therefore,

$$\lambda_{min}(\mathbf{A}) \geq \lambda_{min}(U^\top U)\lambda_{min}(\mathbf{H}).$$

From Assumption 3.5 we know that $\hat{\mathbf{W}}$ is invertible, and also that $J$ is injective, so from Assumption 3.9 and the fact that $J$ is a square matrix, we deduce that $J$ is invertible, and therefore deduce that $U$ is invertible as well. This means that $\lambda_{min}(U^\top U) > 0$ and that

$$\begin{aligned}\lambda_{min}(U^\top U) = \lambda_{min}(UU^\top) &= \lambda_{min}(J^\top \hat{\mathbf{W}}^{-1} J) \\ &\geq \lambda_{min}(J^\top J)\lambda_{min}(\hat{\mathbf{W}}^{-1}) = \frac{\sigma_{min}(J)^2}{\lambda_{max}(\mathbf{W})} \geq \frac{\mu_{\nabla F}^2}{L_W}.\end{aligned}$$

Now we turn to $\mathbf{H}$, and write $\mathbf{H} = \mathbb{E}\left[\mathbf{S}B\mathbf{S}^\top\right]$, where $B = (\mathbf{S}^\top G \mathbf{S})^\dagger$, with $G = J^\top \mathbf{W}^{-1} J$. From the same arguments as above, we know that that $G$ is invertible under our assumptions. So, using properties of the pseudo inverse with the fact that $G$ is symmetric and Lemma B.17, we can write that

$$\mathbf{Null}\left(B\right) = \mathbf{Null}\left(\left(\mathbf{S}^\top G \mathbf{S}\right)^\top\right) = \mathbf{Null}\left(\mathbf{S}^\top G \mathbf{S}\right) = \mathbf{Null}\left(\mathbf{S}\right).$$

Therefore, for all $x \in \mathbb{R}^p$ we have $\mathbf{S}^\top x \in \mathbf{Null}\left(B\right)^\perp$. So, by noting $\lambda_{min}^*(B)$ the smallest nonzero eigenvalue of $B$, we can write that

$$\langle \mathbf{S}B\mathbf{S}^\top x, x\rangle = \langle B\mathbf{S}^\top x, B\mathbf{S}^\top\rangle \geq \lambda_{min}^*(B)\|\mathbf{S}^\top x\|^2 = \lambda_{min}^*(B)\langle \mathbf{S}\mathbf{S}^\top x, x\rangle.$$

Here

$$\begin{aligned}\lambda_{min}^*(B) = \|B^\dagger\|^{-1} &= \|\mathbf{S}^\top G \mathbf{S}\|^{-1} \geq \|\mathbf{S}\mathbf{S}^\top\|^{-1}\|G\|^{-1} \\ &\geq \|\mathbf{S}\mathbf{S}^\top\|^{-1}\|J^\top J\|^{-1}\|\mathbf{W}^{-1}\|^{-1} \geq L_S^{-1}L_{\nabla F}^{-2}\mu_W,\end{aligned}$$

where we used the fact that $\|\mathbf{W}^{-1}\|^{-1} = \lambda_{min}(\mathbf{W})$ and $\|J^\top J\|^{-1} = \sigma_{max}(J)^{-2}$. By combining those last inequalities we obtain that

$$\langle \mathbf{H}x, x\rangle = \mathbb{E}\left[\langle \mathbf{S}B\mathbf{S}^\top x, x\rangle\right]$$

$$\begin{aligned}
&\geq L_S^{-1} L_{\nabla F}^{-2} \mu_W \mathbb{E}\left[\langle \mathbf{S}\mathbf{S}^\top x, x\rangle\right] \\
&= L_S^{-1} L_{\nabla F}^{-2} \mu_W \langle \mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right] x, x\rangle \\
&\geq L_S^{-1} L_{\nabla F}^{-2} \mu_W \bar{\mu}_S \|x\|^2.
\end{aligned}$$

This means that $\lambda_{min}(\mathbf{H}) \geq L_S^{-1} L_{\nabla F}^{-2} \mu_W \bar{\mu}_S$. If we recombine all our inequalities, we ultimately obtain that

$$\lambda_{min}(\mathbf{A}) \geq \frac{\mu_{\nabla F}^2}{L_W} L_S^{-1} L_{\nabla F}^{-2} \mu_W \bar{\mu}_S,$$

which is what we needed. □

**Lemma B.24.** *Let Assumption 3.9 hold. Let $x \in \Omega$, let $(\hat{\mathbf{S}}, \hat{\mathbf{W}})$ be in the domain of $\mathcal{D}_x$. Then*

$$\lambda_{min}\left((\nabla F(x)^\top \mathbf{W}^{-1} \nabla F(x))^\dagger\right) \geq L_{\nabla F}^{-2} \mu_W > 0.$$

*In particular, for all $k \in \mathbb{N}$, if $x^k \in \Omega$ almost surely, then*

$$\mathbb{E}\left[\hat{f}_k(x^k)\right] \geq \frac{\mu_W}{2L_{\nabla F}^2} \mathbb{E}\left[\|F(x^k)\|^2\right] \quad \textit{almost surely.}$$

*Proof.* We have $\lambda_{min}\left((\nabla F(x)^\top \mathbf{W}^{-1} \nabla F(x))^\dagger\right) = \|\nabla F(x)^\top \mathbf{W}^{-1} \nabla F(x)\|^{-1}$ where

$$\|\nabla F(x)^\top \mathbf{W}^{-1} \nabla F(x)\| \leq \sigma_{max}(\nabla F(x))^2 \lambda_{min}(\mathbf{W})^{-1} \leq L_{\nabla F}^2 \mu_W^{-1},$$

which gives the desired lower bound on the eigenvalues. Now, given $x^k \in \Omega$ we immediately deduce that

$$\hat{f}_k(x^k) \geq \frac{\mu_W}{2L_{\nabla F}^2} \|F(x^k)\|^2. \tag{B.50}$$

The conclusion follows by taking the expectation on this inequality. □

*Proof of Theorem 3.10.* Keeping the notations of Theorem 3.8, we see from Lemma B.23 that we can take

$$\rho = \frac{\mu_{\nabla F}^2}{L_{\nabla F}^2} \frac{\mu_W}{L_W} \frac{\bar{\mu}_S}{L_S} > 0,$$

from which we obtain that

$$(\forall k \in \mathbb{N}) \quad \mathbb{E}\left[\hat{f}_k(x^k)\right] \quad \leq \quad (1 - \rho\gamma)^k \, \mathbb{E}\left[\hat{f}_0(x^0)\right] \quad \text{almost surely.}$$

We now can use Lemma B.24 to lower bound the left member of that inequality, and obtain

$$(\forall k \in \mathbb{N}) \quad \frac{\mu_W}{2L_{\nabla F}^2}\mathbb{E}\left[\|F(x^k)\|^2\right] \leq (1-\rho\gamma)^k\,\mathbb{E}\left[\hat{f}_0(x^0)\right] \quad \text{almost surely.} \tag{B.51}$$

The conclusion follows by taking

$$C = \mathbb{E}\left[\hat{f}_0(x^0)\right]\frac{L_{\nabla F}^2}{\mu_W}.$$

□

### B.5.7 Proof of convergence for SAN and SAN for bounded sequences

**Proposition B.25.** *Let Assumption 3.2 hold. For SAN and SANA, Assumption 3.9 holds on every compact set $\Omega$.*

*Proof.* First, remember that Assumption 3.5 holds for SAN and SANA (see Proposition 3.6), and that $m = p = (n+1)d$. Now, let $\Omega$ be a compact set, and verify that the bounds in Assumption 3.9 hold.

We start with the sketching matrices $\mathbf{S}$, for which we know (see the proof of Proposition 3.6 in Section B.5.1) that

$$\|\mathbf{S}\mathbf{S}^\top\| = 1 \quad \text{and} \quad \mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right] = \mathbf{Diag}\left(\pi\mathbf{I}_d, \frac{1-\pi}{n}\mathbf{I}_d, \cdots, \frac{1-\pi}{n}\mathbf{I}_d\right) \text{ or } \mathbf{Diag}\left(\mathbf{I}_d, \frac{1}{n}\mathbf{I}_d, \cdots, \frac{1}{n}\mathbf{I}_d\right).$$

In both cases, we see that we can take $L_S = 1$ and $\bar{\mu}_L = \min\{\frac{1}{n}, \frac{1-\pi}{n}, \pi\}$.

Second, let $\mathbf{W}_i$ be in the domain of $\mathcal{D}_x$. According to their definition in (B.27, B.28), and because the $f_i$ is of class $C^2$ (see Assumption 3.2), we know that each $\mathbf{W}_i$ is continuous with respect to $x$. Moreover, we know (again from Assumption 3.2) that $\mathbf{W}_i$ is definite positive : $\lambda_{min}(\mathbf{W}_i) > 0$. This is true for every $x \in \Omega$, so by continuity of $\lambda_{min}$ and the compactness of $\Omega$, we deduce that $\inf_{x\in\Omega} \lambda_{min}(\mathbf{W}_i) > 0$. Similarly, $\sup_{x\in\Omega} \lambda_{max}(\mathbf{W}_i) < +\infty$. This means that the constants $\mu_W$ and $L_W$ are well defined in $(0, +\infty)$.

Finally, we need to control the singular values of $\nabla F(x)$ over $\Omega$. We use here the same arguments that we used for $\mathbf{W}_i$ : $\nabla F(x)$ is continuous with respect to $x$, and it is invertible (because it is square and injective, see Proposition 3.6). □

**Theorem B.26.** *Let Assumptions 3.2 and 3.7 hold. Let $\{x^k\}_{k\in\mathbb{N}}$ be a sequence generated by SAN with $\pi = 1/(n+1)$, or by SANA, with $\gamma = 1/L$. Suppose that $\{x^k\}_{k\in\mathbb{N}}$ is bounded almost surely.*

*Then there exists $C, \rho > 0$ such that, for every $k \in \mathbb{N}$,*

$$\mathbb{E}\left[\|F(x^k)\|^2\right] \leq C(1-\gamma\rho)^k \text{ a.s.}$$

*Proof.* There exists a compact set $\Omega$ containing almost surely the sequence. So it remains to combine Theorem 3.10 together with Proposition B.25 and Proposition 3.6. □

### B.5.8 Proof of Theorem 3.12

The proof of Theorem 3.12, which can be found at the end of this section, will combine Theorem 3.10 with the forthcoming Propositions B.29 and B.30.

**Lemma B.27.** *Let $\varphi : [0, +\infty) \longrightarrow [1, +\infty)$ be defined as*

$$\varphi(t) := \sqrt{1 + \frac{1}{2}\left(t + \sqrt{4t + t^2}\right)}. \tag{B.52}$$

1. *$\varphi(t)$ is well defined and increasing on $[0, +\infty[$.*
2. *$\varphi(t)^{-1} = \sqrt{1 + \frac{1}{2}\left(t - \sqrt{4t + t^2}\right)}$.*
3. *For all $a \in (0, +\infty)$, $\varphi(at)\varphi(t^{-1})t^{-1/2}$ is decreasing on $(0, +\infty)$.*
4. *For all $a \in (0, +\infty)$, and all $t \geq 1$, $\varphi(at)\varphi(t^{-1}) \leq \varphi(1)\sqrt{t}\sqrt{2 + a}$.*

*Proof.* 1 : It is well defined because $t + \sqrt{4t + t^2} \geq 0$. It is increasing because it is the composition, sum and product of increasing functions on $[0, +\infty[$. Point 2 is a simple exercise. Point 3 is a bit more technical. Let $\phi(t) = \varphi(at)^2\varphi(t^{-1})^2 t^{-1}$, which is the square of the quantity of interest. We can compute its derivative, and a some effort we obtain that

$$\begin{aligned} t^3\phi'(t) = &- \left[t + \frac{1}{2}\left(1 + \sqrt{1 + 4t}\right)\right]\left[1 + \frac{at}{\sqrt{4at + a^2t^2}}\right] \\ &- \frac{1}{2}\left[1 + \frac{1 + 2t}{\sqrt{1 + 4t}}\right]\left[1 + \frac{1}{2}\left(at + \sqrt{4at + a^2t^2}\right)\right]. \end{aligned}$$

It is clear that the above expression is the sum of two negative terms, implying that $\phi$ is decreasing. For item 4, we use the monotonicity of item 3 to get

$$\varphi(at)\varphi(t^{-1}) \leq \varphi(a \cdot 1)\varphi(1)\sqrt{t}.$$

From the inequality $a + \sqrt{a^2 + 4a} \leq 2a + 2$ (it is easy to prove it by rearranging the terms and taking the square), we deduce that

$$\varphi(a) \leq \sqrt{1 + \frac{1}{2}(2a + 2)} = \sqrt{2 + a}.$$

□

**Lemma B.28.** *Let $\mathbf{A} \in \mathbb{R}^{m \times d}$ be an injective matrix. Let $\varphi$ be defined as in (B.52) and consider:*

$$A := \begin{bmatrix} \mathbf{I}_d & \mathbf{0}_{d,m} \\ \mathbf{A} & \mathbf{I}_m \end{bmatrix}.$$

*Then $\|A\| = \varphi\left(\|\mathbf{A}^\top \mathbf{A}\|\right)$.*

*Proof.* We start by remembering that $\|A\|$ is the largest singular value of $A$. The singular values of $A$ are exactly the square root of the eigenvalues of $A^\top A$, which is given by

$$A^\top A := \begin{bmatrix} \mathbf{I}_d + \mathbf{A}^\top \mathbf{A} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{I}_m \end{bmatrix}.$$

We compute its eigenvalues by finding the roots of its characteristic polynomial, that we note $P \in \mathbb{R}[X]$. Using a simple formula for computing the determinant of a $2 \times 2$ block matrix, we can write, for all $X \neq -1$ :

$$\begin{aligned} P(X) &= \det \begin{bmatrix} (1-X)\mathbf{I}_d + \mathbf{A}^\top \mathbf{A} & \mathbf{A}^\top \\ \mathbf{A} & (1-X)\mathbf{I}_m \end{bmatrix} \\ &= \det\left((1-X)\mathbf{I}_m\right) \det\left((1-X)\mathbf{I}_d + \mathbf{A}^\top \mathbf{A} - \mathbf{A}^\top ((1-X)\mathbf{I}_m)^{-1} \mathbf{A}\right) \\ &= (1-X)^m \det\left((1-X)\mathbf{I}_d + \mathbf{A}^\top \mathbf{A} - \frac{1}{1-X}\mathbf{A}^\top \mathbf{A}\right) \\ &= (1-X)^{m-d} \det\left((1-X)^2 \mathbf{I}_d + (1-X)\mathbf{A}^\top \mathbf{A} - \mathbf{A}^\top \mathbf{A}\right) \\ &= (1-X)^{m-d} \det\left((1-X)^2 \mathbf{I}_d - X\mathbf{A}^\top \mathbf{A}\right). \end{aligned}$$

The right member of this equality is polynomial in $X$, since the determinant of a matrix is polynomial in its coefficients, and our assumption that $\mathbf{A}$ is injective implies that $m - d \geq 0$. In particular this right member is well defined and continuous at $X = -1$, which means that the equality holds true for every $X \in \mathbb{R}$.

We see that 1 is a root of $P$, with multiplicity $m-d$. The other roots are the zeroes of $\det\left((1-X)^2\mathbf{I}_d - X\mathbf{A}^\top\mathbf{A}\right)$, for which we see that

$$\begin{aligned}\det\left((1-X)^2\mathbf{I}_d - X\mathbf{A}^\top\mathbf{A}\right) = 0 \quad &\Leftrightarrow \quad (1-X)^2 \text{ is an eigenvalue of } X\mathbf{A}^\top\mathbf{A}\\ &\Leftrightarrow \quad (1-X)^2 = X\lambda \text{ for } \lambda \in \operatorname{spec}(\mathbf{A}^\top\mathbf{A})\\ &\Leftrightarrow \quad X = 1 + \frac{1}{2}\left(\lambda \pm \sqrt{4\lambda+\lambda^2}\right) \text{ for } \lambda \in \operatorname{spec}(\mathbf{A}^\top\mathbf{A}),\end{aligned}$$

which gives us the remaining $2d$ roots (counted with multiplicity). This proves that the singular values of $A$ are 1 (with multiplicity $m-d$) and (see Lemma B.27.2)

$$\{\varphi(\lambda), \varphi(\lambda)^{-1} \mid \lambda \in \operatorname{spec}\left(\mathbf{A}^\top\mathbf{A}\right)\}.$$

Since $\varphi$ is increasing (Lemma B.27.1), and $\varphi(\lambda) \geq 1$, we conclude that the largest singular value of $A$ is $\varphi\left(\|\mathbf{A}^\top\mathbf{A}\|\right)$.

□

**Proposition B.29.** *Let Assumption 3.11 hold, and consider the SAN (resp. SANA) algorithm. Let* $c = \sqrt{\frac{3+\sqrt{5}}{2}}$. *Then Assumption 3.9 is verified, with* $\Omega = \mathbb{R}^p$ *and:*

$$\begin{aligned}\mu_W &= \min\{1, \mu_f\}, & L_W &= \max\{1, L_f\},\\ \bar{\mu}_S &= \min\left\{\tfrac{1-\pi}{n}, \pi\right\} \left(\textit{resp. } \bar{\mu}_S = \tfrac{1}{n}\right), & L_S &= 1,\\ \mu_{\nabla F} &= \frac{\mu_W}{c\sqrt{n}\sqrt{2+L_f^2}}, & L_{\nabla F} &= L_W c\sqrt{n}\sqrt{2+L_f^2}.\end{aligned}$$

*Proof.* Let $x \in \mathbb{R}^p$ be fixed, $J := \nabla F(x) \in \mathbb{R}^{p\times p}$, and $(\mathbf{S}, \mathbf{W})$ in the domain of $\mathcal{D}_x$. We need to find uniform spectral bounds on those three matrices. For $\mathbf{S}$, we have seen already in the proof of Proposition B.25 that we can take $L_S = 1$, and $\bar{\mu}_L = \frac{1}{n}$ for SAN, or $\min\{\frac{1-\pi}{n}, \pi\}$ for SANA. For $\mathbf{W}$, we see directly from (B.27, B.28) that it is a block-diagonal matrix, whose eigenvalues are included in $[\mu_W, L_W]$, with $\mu_W = \min\{1, \mu_f\}$ and $L_W = \max\{1, L_f\}$. The rest of this proof is dedicated to the study of $J$, which requires more work.

Remember that the expression for $J$ is given in (B.19). We write for convenience that

$$J = \begin{bmatrix} \mathbf{0}_d & \mathbf{H} \\ \frac{1}{n}\mathbf{E}^\top & -\mathbf{I}_{nd} \end{bmatrix},$$

where

$$\mathbf{E} := [\mathbf{I}_d, \cdots, \mathbf{I}_d] \in \mathbb{R}^{d\times nd}, \quad \mathbf{H} := [H_1, \cdots, H_n] \in \mathbb{R}^{d\times nd}, \quad H_i := \nabla^2 f_i(w).$$

Let us now introduce a few more matrices. Let $\bar{H} := \nabla^2 f(w)$, which can equivalently be written as $\bar{H} = \frac{1}{n}\sum_i H_i = \frac{1}{n}\mathbf{H}\mathbf{E}^\top$. Now consider

$$U := \begin{bmatrix} \mathbf{I}_d & \mathbf{0}_{d,nd} \\ -\mathbf{H}^\top & \mathbf{I}_{nd} \end{bmatrix}, \quad D := \begin{bmatrix} \bar{H} & \mathbf{0}_{d,nd} \\ \mathbf{0}_{nd,d} & -\mathbf{I}_{nd} \end{bmatrix}, \quad V := \begin{bmatrix} \mathbf{I}_d & \mathbf{0}_{d,nd} \\ \frac{-1}{n}\mathbf{E}^\top & \mathbf{I}_{nd} \end{bmatrix}.$$

Note that those three matrices are triangular, and invertible because $\bar{H}$ is invertible. It is easy to see that $J = U^\top DV$. Indeed,

$$DV = \begin{bmatrix} \bar{H} & \mathbf{0}_{d,nd} \\ \frac{1}{n}\mathbf{E}^\top & -\mathbf{I}_{nd} \end{bmatrix}, \quad U^\top DV = \begin{bmatrix} \bar{H} - \mathbf{H}\frac{1}{n}\mathbf{E}^\top & \mathbf{H} \\ \frac{1}{n}\mathbf{E}^\top & -\mathbf{I}_{nd} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_d & \mathbf{H} \\ \frac{1}{n}\mathbf{E}^\top & -\mathbf{I}_{nd} \end{bmatrix},$$

where the last equality comes from the fact that $\mathbf{H}\frac{1}{n}\mathbf{E}^\top = \bar{H}$. Therefore, it remains to upper bound the right member of

$$\sigma_{max}(J) = \sqrt{\lambda_{max}(J^\top J)} = \|J\| \leq \|D\|\|U\|\|V\|.$$

Bounding the smallest singular value will follow the same argument. Indeed, from our assumptions, $J^T J$ is invertible (see Proposition 3.6), but $J$ is a square matrix, therefore $J$ itself is invertible. In consequence, we can write that $J^{-1} = {V^{-1}}^\top D^{-1} U^{-1}$, so that

$$\sigma_{min}(J) = \frac{1}{\|J^{-1}\|} \geq \frac{1}{\|D^{-1}\|\|U^{-1}\|\|V^{-1}\|},$$

where one easily computes that

$$U^{-1} = \begin{bmatrix} \mathbf{I}_d & \mathbf{0}_{d,nd} \\ \mathbf{H}^\top & \mathbf{I}_{nd} \end{bmatrix}, \quad D^{-1} = \begin{bmatrix} \bar{H}^{-1} & \mathbf{0}_{d,nd} \\ \mathbf{0}_{nd,d} & -\mathbf{I}_{nd} \end{bmatrix}, \quad V^{-1} = \begin{bmatrix} \mathbf{I}_d & \mathbf{0}_{d,nd} \\ \frac{1}{n}\mathbf{E}^\top & \mathbf{I}_{nd} \end{bmatrix}.$$

It is easy to see, given our smoothness and strong convexity assumptions, that

$$\|D\| = \max\{1, L\} = L_W \quad \text{and} \quad \|D^{-1}\| = \max\{1, \mu^{-1}\} = \mu_W^{-1}.$$

Now, observe that $V$ and $V^{-1}$ share the same structure, so we can call Lemma B.28 with $\mathbf{A} = \frac{1}{n}\mathbf{E}^T$ or $-\frac{1}{n}\mathbf{E}^T$. In both cases $\mathbf{A}^\top\mathbf{A} = n^{-1}\mathbf{I}_d$, meaning that $\|\mathbf{A}^\top\mathbf{A}\| = n^{-1}$, and so we deduce that

$$\|V\| = \|V^{-1}\| = \varphi(n^{-1}).$$

Finally, we do the same for $U$ and $U^{-1}$ : we use Lemma B.28 with $\mathbf{A} = \pm\mathbf{H}^\top$. In both cases $\mathbf{A}^\top\mathbf{A} = \mathbf{H}\mathbf{H}^\top = \sum_{i=1}^n H_i^2$, whose eigenvalues belong to $[n\mu^2, nL^2]$. Due to the monotonicity of $\varphi$, we deduce that

$$\|U\| = \|U^{-1}\| \leq \varphi(nL^2).$$

As a result, we conclude that

$$\sigma_{min}(J) \geq \frac{1}{\mu_W^{-1}\varphi(n^{-1})\varphi(nL^2)} = \frac{\mu_W}{\varphi(n^{-1})\varphi(nL^2)},$$

while

$$\sigma_{max}(J) \leq L_W\varphi(n^{-1})\varphi(nL^2).$$

We can then conclude by using Lemma B.27.4 and noting $c = \varphi(1)$. □

**Proposition B.30.** *Let Assumption 3.11 hold, and consider the SAN (resp. SANA) algorithm. Then*

1. $F : \mathbb{R}^p \longrightarrow \mathbb{R}^p$ *is a diffeomorphism*
2. $F^{-1} : \mathbb{R}^p \longrightarrow \mathbb{R}^p$ *is Lipschitz continuous :*

$$(\forall x, y \in \mathbb{R}^p) \quad \|x - y\| \leq \frac{c\sqrt{n}\sqrt{2 + L_f^2}}{\min\{1, \mu_f\}}\|F(x) - F(y)\| \quad \textit{with} \quad c = \sqrt{\frac{3+\sqrt{5}}{2}}.$$

3. $F^{-1}(0) = [w^*,\ \nabla f_1(w^*),\ \ldots,\ \nabla f_n(w^*)]$, *with* $w^* = \operatorname{argmin} f$.

*Proof.* Let us start by showing that $F$ is injective. For $x, \hat{x} \in \mathbb{R}^p$, we have

$$F(x) = F(\hat{x}) \ \Rightarrow\ \frac{1}{n}\sum_i \alpha_i = \frac{1}{n}\sum_i \hat{\alpha}_i \ \text{ and } \ \nabla f_i(w) - \alpha_i = \nabla f_i(\hat{w}) - \hat{\alpha}_i,\ \forall i = 1, \ldots, n. \quad \text{(B.53)}$$

Summing the right member over $i$, we obtain that $\nabla f(w) - \nabla f(\hat{w}) = \frac{1}{n}\sum_i \alpha_i - \frac{1}{n}\sum_i \hat{\alpha}_i = 0$. In other words, we obtained that $\nabla f(w) = \nabla f(\hat{w})$. Now, we assumed that $f$ is strongly convex, therefore $\nabla f$ is injective (this can be seen from the fact that $\nabla f$ is strongly monotone). So we deduce that $w = \hat{w}$. Going back to (B.53), we see now that $\alpha_i = \hat{\alpha}_i$, from which we conclude that $x = \hat{x}$, and that $F$ is indeed injective.

We know that $\nabla F(x)$ is invertible for all $x \in \mathbb{R}^p$ (see Proposition 3.6), so we can use the global inversion theorem to deduce that $F$ is a diffeomorphism between $\mathbb{R}^p$ and $F(\mathbb{R}^p)$. Let us prove now that $F(\mathbb{R}^p) = \mathbb{R}^p$. For this we will use an argument analog to what we used in the

proof of Proposition B.29. Let $u, d, v^\top : \mathbb{R}^p \longrightarrow \mathbb{R}^p$ be defined as

$$\begin{aligned} u(x) &:= (w, -\nabla f_1(w) + \alpha_1, \dots, -\nabla f_n(w) + \alpha_n), \\ v^\top(x) &:= (w - \frac{1}{n}\sum_{i=1}^n \alpha_i, \alpha_1, \dots, \alpha_n), \\ d(x) &:= (\nabla f(w), -\alpha_1, \dots, -\alpha_n). \end{aligned}$$

It is a simple exercise to verify that $F = v^\top \circ d \circ u$. We will see now that those three functions are invertible and that their inverses are given by

$$\begin{aligned} u^{-1}(x) &:= (w, \nabla f_1(w) + \alpha_1, \dots, \nabla f_n(w) + \alpha_n), \\ v^{\top -1}(x) &:= (w + \frac{1}{n}\sum_{i=1}^n \alpha_i, \alpha_1, \dots, \alpha_n), \\ d^{-1}(x) &:= (\nabla f^*(w), -\alpha_1, \dots, -\alpha_n). \end{aligned}$$

For $u$, it is again a simple exercise to verify that $u \circ u^{-1} = u^{-1} \circ u = id_{\mathbb{R}^p}$. Same for $v$ (which actually is linear). For $d$, we used the notation $f^*$ which refers to the Fenchel transform of $f$,

$$f^*(u) := \sup_{w \in \mathbb{R}^d} \langle u, w \rangle - f(w).$$

We assumed $f$ to be strongly convex and smooth, which means that $f^*$ is well-defined and differentiable on $\mathbb{R}^p$, and that $(\nabla f)^{-1} = \nabla f^*$ (Bauschke and Combettes, 2017, Theorems 13.37, 16.29 & 18.15). This proves that our expression for $d^{-1}$ is correct. Now we conclude, by seeing that $u^{-1}, v^{\top -1}, d^{-1}$ are well-defined on $\mathbb{R}^p$, that $F^{-1}$ is well-defined on $\mathbb{R}^p$, and so that $F(\mathbb{R}^p) = \mathbb{R}^p$.

Now, we focus on $F^{-1} : \mathbb{R}^p \longrightarrow \mathbb{R}^p$. It is differentiable everywhere, so we can use the mean value theorem to deduce that $F^{-1}$ is Lipschitz continuous on $\mathbb{R}^p$, with a Lipschitz constant being bounded by:

$$\sup_{y \in \mathbb{R}^p} \|\nabla F^{-1}(y)\| = \sup_{x \in \mathbb{R}^p} \|\nabla F^{-1}(F(x))\| = \sup_{x \in \mathbb{R}^p} \|\nabla F(x)^{-1}\| = \sup_{x \in \mathbb{R}^p} \frac{1}{\sigma_{min}(\nabla F(x))} \leq \frac{1}{\mu_{\nabla F}},$$

where $\mu_{\nabla F}$ was computed in Proposition B.29. We obtain that

$$\|x - y\| = \|F^{-1}(F(x)) - F^{-1}(y)\| \leq \frac{1}{\mu_{\nabla F}} \|F(x) - F(y)\|.$$

To conclude about the expression of $F^{-1}(0)$, compute

$$F^{-1}(x) = (u^{-1} \circ d^{-1} \circ v^{\top -1})(x)$$

$$= [\hat{w},\ \nabla f_1(\hat{w}) - \alpha_1,\ \cdots,\ \nabla f_n(\hat{w}) - \alpha_n],\ \text{with } \hat{w} = \nabla f^*\left(w + \frac{1}{n}\sum_{i=1}^n \alpha_i\right),$$

and use the fact that $\nabla f^*(0) = w^*$. □

*Proof of Theorem 3.12.* Let $w^* = \operatorname{argmin} f$, and $x^* = [w^*, \nabla f_1(w^*), \ldots, \nabla f_n(w^*)]$, such that $F(x^*) = 0$ according to Proposition B.30. Using again Proposition B.30, we obtain for all $k \in \mathbb{N}$ that

$$\|x^k - x^*\|^2 \leq \frac{c^2 n(2 + L_f^2)}{\min\{1, \mu_f^2\}} \|F(x^k)\|^2.$$

Here we have

$$\|x^k - x^*\|^2 = \|w^k - w^*\|^2 + \sum_{i=1}^n \|\alpha_i^k - \nabla f_i(w^*)\|^2.$$

Taking the expectation on the above expressions, and using the fact that $c^2 \leq 3$, we obtain that

$$\mathbb{E}\left[\|w^k - w^*\|^2\right] + \sum_{i=1}^n \mathbb{E}\left[\|\alpha_i^k - \nabla f_i(w^*)\|^2\right] \leq \frac{3n(2 + L_f^2)}{\min\{1, \mu_f^2\}} \mathbb{E}\left[\|F(x^k)\|^2\right].$$

Now we can use Theorem 3.10 together with Proposition B.29, and combine the constants into play. For the sake of the presentation, we assume $\pi = 1/(n+1)$, so that we can have a unique lower bound for SAN and SANA : $\bar{\mu}_S = \min\{\frac{1}{n}, \frac{1}{n+1}\} \geq \frac{1}{2n}$. We also simplify the expression of $c$, by again using bounds like $c^2 \leq 3$ or $c^4 \leq 7$. This allows us to write that

$$\mathbb{E}\left[\|F(x^k)\|^2\right] \leq C'(1 - \gamma\rho)^k \quad \text{almost surely,}$$

with $\rho = \frac{\min\{1, \mu_f^3\}}{14n^3(2+L_f^2)^2 \max\{1, L_f^3\}}$, and $C' = 6n\mathbb{E}\left[\hat{f}_0(x^0)\right] \frac{\max\{1, L_f^2\}(2+L_f^2)}{\min\{1, \mu_f\}}$. The conclusion follows by taking $C = C' \frac{3n(2+L_f^2)}{\min\{1, \mu_f^2\}}$:

$$\mathbb{E}\left[\|w^k - w^*\|^2\right] + \sum_{i=1}^n \mathbb{E}\left[\|\alpha_i^k - \nabla f_i(w^*)\|^2\right] \leq 18n^2 \mathbb{E}\left[\hat{f}_0(x^0)\right] \frac{\max\{1, L_f^2\}(2 + L_f^2)^2}{\min\{1, \mu_f^3\}} (1 - \gamma\rho)^k.$$

□

# Appendix C

# Complements on Chapter 4

Contents



Here we provide the related work discussion, the missing proofs from Chapter 4 and some additional noteworthy observations made in Chapter 4.

## C.1 Related work

We provide an extended discussion for the context of our work, including a discussion comparing the technical novelty of the chapter to the finite sum minimization result in Khaled and Richtárik (2023), a comparison of the convergence theories of vanilla PG and the problem dependent constants.

### C.1.1 Technical contribution and novelty compared to Khaled and Richtárik (2023)

Our technical novelty compared to Khaled and Richtárik (2023) is threefold:

- First, Theorem 4.4 is not a direct application of Theorem 2 in Khaled and Richtárik (2023), which requires unbiased estimators of the gradient. Yet in PG methods, we have to deal with biased estimators due to the truncation of the trajectories. The first technical challenge was to adapt the proof technique to allow for biased gradients and a truncation error. This also explains the need of Assumption 4.2. Similarly, we need to handle the same challenge for the proof of Theorem C.22 when adapting the proof of Theorem 3 in Khaled and Richtárik (2023).

- Second, when considering the results we derived in specific cases in Section 4.4, the difference between our work and Khaled and Richtárik (2023) is even more significant. All cases studied in Khaled and Richtárik (2023) (e.g., finite-sum structure) are not applicable for PG methods and we had to derive specific analysis for our specialized settings (softmax with different regularizers, expected Lipschitz and smooth policies, Fisher-non-degenerate parametrized policies). Furthermore, our focus is on deriving explicit sample complexity, whereas the results in Khaled and Richtárik (2023) are concerned with convergence rates in terms of number of iterations. These dimensions are where most of the technical work was done. Without this work of developing sample complexity and studying specific cases found in PG literature, it was not clear at all that the (ABC) assumption proposed in Khaled and Richtárik (2023) would be relevant in RL.

- Finally, we also consider the setting where the relaxed weak gradient domination holds (Assumption 4.6 and Theorem C.8). This is an assumption that is unique to PG methods and had not been considered in Khaled and Richtárik (2023). Technically speaking, the proof of Theorem C.8 is unique and required a different approach (see the arguments following (C.20)).

### C.1.2 Sample complexity analysis of the vanilla policy gradient

Despite the success of PG methods in practice, a comprehensive theoretical understanding was lacking until recently.

**Global optimum convergence of vanilla PG with the exact full gradient.** We refer to global optimum convergence as an analysis that guarantees that $J^* - J(\theta_T) \leq \epsilon$ after $T$ iterations. The global optimum convergence results of PG with the exact full gradient have been developed under a number of different specific settings.

By using a gradient domination property of the expected return, which is also referred to as the Polyak-Lojasiewicz (PL) condition (Polyak, 1963; Łojasiewicz, 1963), Fazel et al. (2018) show that the linear-quadratic regulator (LQR) converges linearly to the global optimum for PG with the exact full gradient. However, in the LQR setting the function $J$ is not smooth, and thus does

not fit into the general setting we considered in this chapter. Notice that such (PL) condition is widely explored by Bhandari and Russo (2019) to identify more general MDP settings. When such (PL) condition holds, Bhandari and Russo (2019) show that any stationary point of the policy gradient of the expected return is a global optimum. More recently, Agarwal et al. (2021) leveraged a *weak* gradient domination property, also called the weak Polyak-Lojasiewicz condition which is exactly our condition (4.17) with $\epsilon' = 0$, to show that the projected PG converges to the global optimum with a $\mathcal{O}(\epsilon^{-2})$ convergence rate in tabular MDPs with tabular policies, also called direct policy parameterization. In later work, Xiao (2022) improve this result by a factor of $\epsilon$, i.e., they establish a $\mathcal{O}(\epsilon^{-1})$ convergence rate for the projected PG in the tabular setting when the exact full gradient is available. At the moment, we could not adapt our general ABC structure to analyze and derive a sample complexity guarantee for the projected PG. The same convergence rate $\mathcal{O}(\epsilon^{-1})$ is developed by Zhang et al. (2020a) by leveraging the hidden convex structure of the cumulative reward and consequently showing that all local optima (i.e., stationary points) are in fact global optima under certain bijection assumptions based on the occupancy measure space (Assumption 1 in Zhang et al. (2020a)). Notice that the assumptions proposed by Zhang et al. (2020a) are satisfied in the specific case of the tabular setting. We do not cover this specific assumption in our current analysis.

The global optimum convergence analysis with exact PG is also investigated in the case of softmax tabular policy with or without regularization. Agarwal et al. (2021) first provide an asymptotic convergence for the softmax tabular without regularization and a $\mathcal{O}(\epsilon^{-2})$ convergence rate for the softmax tabular with log barrier regularization. Even though the gradient domination property ((PL) or (4.17)) is not globally satisfied for the softmax tabular, Mei et al. (2020) prove that it is available by following the path of the iterations with the exact full gradient updates. Such a property is called the non-uniform Lojasiewicz inequality. Consequently, Mei et al. (2020) show a $\mathcal{O}(\epsilon^{-1})$ convergence rate for the softmax tabular without regularization by the weak gradient domination condition and a linear convergence rate for the softmax tabular with entropy regularization by the gradient domination condition. Finally, Li et al. (2021a) recently showed that the result of Mei et al. (2020) for softmax tabular policies may actually contain a term that is exponential in the discount factor, thus showing that exact PG may take an exponential time to converge.

*Our Contributions.* We provide a general sample complexity analysis which, when instantiated using specific settings given in the literature, recovers the same or even slightly improved convergence rates. Indeed, from Corollary C.14 we recover the $\mathcal{O}(\epsilon^{-2})$ convergence rate of Agarwal et al. (2021) for the softmax tabular with log barrier regularization and improve the rate by a factor of $1-\gamma$ through a better analysis of the smoothness constant. By leveraging the (relaxed weak) gradient domination properties which hold under the path of the iterations (Mei et al., 2020), we recover their results. That is, we recover the $\mathcal{O}(\epsilon^{-1})$ convergence rate for the softmax

tabular without regularization in Theorem C.8 and the linear convergence rate for the softmax tabular with entropy regularization in Theorem C.22.

**Sample complexity for FOSP convergence.** The convergence rates derived for exact PG are representative of the behavior of the algorithm but do not take into account the additional errors due to the stochastic nature of the actual algorithm used in practice. In this chapter we mostly focus on the sample complexity of the stochastic vanilla PG for FOSP convergence. The well known sample complexity for REINFORCE is $\widetilde{\mathcal{O}}(\epsilon^{-4})$ s.t. $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\widehat{\nabla}_m J(\theta_t)\right\|^2\right] \leq \epsilon^2$ after $T$ iterations. However, as Papini (2020) mentioned, *"formal proofs of this result are surprisingly hard to find both in the policy optimization and in the nonconvex optimization literature."* Papini (2020) give a proof of the result under the expected Lipschitz and smooth policy assumption (E-LS) in Theorem 7.1. When an estimate of the Q-function is available, Zhang et al. (2020b) also establish the same dependency on $\epsilon$ for the sample complexity of FOSP convergence for the policy gradient theorem (Sutton et al., 2000) under more restrictive Lipschitz and smooth policy assumption (LS). By adding an additional uniform ergodicity assumption (Mitrophanov, 2005), Xiong et al. (2021) improve the sample complexity of (Zhang et al., 2020b) by some factors of $1-\gamma$ but still has the same dependency on $\epsilon$.

*Our Contributions.* We establish the sample complexity analysis for the vanilla PG: REINFORCE (4.4) and GPOMDP (4.6). We improve the results of Papini (2020), Zhang et al. (2020b), and Xiong et al. (2021) by using weaker assumptions and allowing much wider range of hyper parameters (the batch size $m$ and the constant step size $\eta$) to achieve the optimal sample complexity. Overall, for both the exact and stochastic PG, our general sample complexity analysis recovers the state-of-the-art dependency on $\epsilon$ under the ABC assumption.

**Sample complexity for global optimum convergence.** We refer to sample complexity of global optimum convergence as an analysis that guarantees that $J^* - \mathbb{E}\left[J(\theta_T)\right] \leq \epsilon$ after $T$ iterations. To the best of our knowledge, there is no existing analysis that considers this type of convergence result for the stochastic vanilla PG. As for variance-reduced PG, by using Assumption 1 in Zhang et al. (2020a) about occupancy distribution, Zhang et al. (2021a) establish a $\widetilde{\mathcal{O}}(\epsilon^{-2})$ sample complexity to achieve the global optimum.

*Our Contributions.* Under the ABC assumption, the smoothness and an additional gradient domination type assumptions (4.17) and (PL), we establish the faster sample complexity analysis for the global optimum convergence in Section 4.3.2 and Section C.8. More precisely, when the relaxed weak gradient domination assumption (4.17) is available, we establish $\widetilde{\mathcal{O}}(\epsilon^{-3})$ sample complexity in Theorem C.8. We also show that one wide family of policies, the Fisher-non-degenerate parametrized policies, satisfy this relaxed weak gradient domination assumption. When the gradient domination assumption (PL) is available, we establish $\widetilde{\mathcal{O}}(\epsilon^{-1})$

sample complexity for the global optimum in Theorem C.22. It remains an open question whether softmax or softmax with entropy still satisfy the (weak) gradient domination type of assumptions for the stochastic PG updates based on the exact PG analysis of Mei et al. (2020).

**Sample complexity for the average regret convergence.** We refer to the sample complexity for average regret as an analysis that guarantees that $J^* - \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[J(\theta_t)\right] \leq \epsilon$. Zhang et al. (2021b) show that with sample complexity $\widetilde{\mathcal{O}}(\epsilon^{-6})$, PG methods can converge to the average regret optimum by using as little as a single sampled trajectory per iteration (i.e., mini-batch size $m = 1$) for softmax with log barrier regularization. However, their setting does not use "vanilla" PG but a modified version with re-projection meant to guarantee a sufficient level of policy randomization. Liu et al. (2020) obtain faster sample complexity $\widetilde{\mathcal{O}}(\epsilon^{-4})$ by assuming in addition a Fisher-non-degenerate parameterization, i.e. the Fisher information matrix is strictly lower bounded (Assumption 4.19), and the compatible function approximation assumption (Assumption 4.20). Notice that the softmax with log barrier regularization does not satisfy all these assumptions and they require large batch sizes per iteration.

*Our Contributions.* We recover the sample complexity for the average regret convergence $\widetilde{\mathcal{O}}(\epsilon^{-6})$ of Zhang et al. (2021b) in the softmax with log barrier regularization with the vanilla PG setting . Compared to their results, we show that the extra phased learning step is unnecessary and the step size can be constant instead of using a decreasing step size. We also provide a wider range of parameter choices for the batch size and the step size with the same sample complexity. For the Fisher-non-degenerate parametrized policy, we also recover the sample complexity for the average regret convergence $\widetilde{\mathcal{O}}(\epsilon^{-4})$ of Liu et al. (2020) in Corollary C.17. Compared to their results, we improve upon them by using weaker assumption E-LS, allowing much wider range of choices for the batch size $m \in \left[1; \frac{2\nu}{\epsilon^2}\right]$ and the corresponded constant step size $\eta$ to achieve the same optimal sample complexity $\widetilde{\mathcal{O}}\left(\epsilon^{-4}\right)$.

### C.1.3 Better analysis of the problem dependent constants

Throughout the chapter, we also provided tighter bounds on the smoothness constants, Lipschitzness constants, and the variance of the gradient estimators under Assumption (E-LS). Notice that the smoothness and Lipschitz constants we consider here are properties of the expected return $J(\cdot)$ in (4.2) or the regularized expected return $L_\lambda(\cdot)$ in (4.31). They depend only on the assumptions and are independent to the specific PG algorithm. For this reason, below we compare our bounds with work that studies variants of PG other than vanilla PG, where the bounds on the smoothness and Lipschitz constants are also needed. On the other hand, for the variance of the gradient estimators, we only consider the vanilla gradient estimators REINFORCE (4.4) and GPOMDP (4.6) with batch size $m$. A resume of the improved problem dependent constants – smoothness and Lipschitzness constants, is provided in Table C.1.

**Table C.1** – E-LS constants $G, F$ (Assumption 4.8), smoothness constant $L$ and Lipschitzness constant $\Gamma$ for Gaussian and (regularized) Softmax tabular policies, where $\varphi$ is an upper bound on the euclidean norm of the feature function for the Gaussian policy, $R_{\max}$ is the maximum absolute-valued reward, $\gamma$ is the discount factor, $\sigma$ is the standard deviation of the Gaussian policy.

| | **Gaussian*** | **Softmax** | **Softmax with log barrier** |
|---|---|---|---|
| $G^2$ | $\frac{\varphi^2}{\sigma^2}$ | $1 - \frac{1}{\lvert\mathcal{A}\rvert}$ | ✗** |
| $F$ | $\frac{\varphi^2}{\sigma^2}$ | $1$ | ✗ |
| $L$ | $\frac{2\mathcal{R}_{\max}\varphi^2}{(1-\gamma)^2\sigma^2}$ | $\frac{\mathcal{R}_{\max}}{(1-\gamma)^2}\left(2 - \frac{1}{\lvert\mathcal{A}\rvert}\right)$ | $\frac{\mathcal{R}_{\max}}{(1-\gamma)^2}\left(2 - \frac{1}{\lvert\mathcal{A}\rvert}\right) + \frac{\lambda}{\lvert\mathcal{S}\rvert}$ |
| $\Gamma$ | $\frac{\mathcal{R}_{\max}\varphi}{(1-\gamma)^{3/2}\sigma}$ | $\frac{\mathcal{R}_{\max}}{(1-\gamma)^{3/2}}\sqrt{1 - \frac{1}{\lvert\mathcal{A}\rvert}}$ | $\sqrt{2\left(1 - \frac{1}{\lvert\mathcal{A}\rvert}\right)\left(\frac{\mathcal{R}_{\max}^2}{(1-\gamma)^3} + \frac{\lambda^2}{\lvert\mathcal{S}\rvert}\right)}$ |

*The (E-LS) constants $G^2$ and $F$ are provided in Lemma 23 in Papini et al. (2022).

**When there is a "✗", it means this is not applicable directly in such setting.

**Smoothness constant.** The smoothness constant $L = \frac{\mathcal{R}_{\max}}{(1-\gamma)^2}(G^2 + F)$ in (4.22) provided in Lemma 4.11 is tighter as compared to Papini et al. (2022, Lemma 6) under Assumption (E-LS), and is also tighter as compared to Xu et al. (2020b, Proposition 4.2 (2)) and Liu et al. (2020, Lemma B.1) under more restrictive assumptions (LS). Compared to existing bounds, our result shows that when $\gamma$ is close to 1, the smoothness constant (4.22) depends on $(1-\gamma)^{-2}$ instead of $(1-\gamma)^{-3}$ as derived in Papini et al. (2022), Xu et al. (2020b) and Liu et al. (2020). Consequently, the smoothness constant for softmax derived in Lemma C.10 and C.12 are also tighter than the one derived in Lemma 7 in Mei et al. (2020) and Lemma D.2 in Agarwal et al. (2021), which both have the dependency of $(1-\gamma)^{-3}$. Finally, compared to the smoothness constant in Shen et al. (2019) and Xu et al. (2020a), our result is independent to the horizon $H$ .

Recent works, such as Proposition 1 in Huang et al. (2020) and equation (17) in Yuan et al. (2020), have the dependency of $(1-\gamma)^{-2}$ for the smoothness constant under assumptions (LS). However, this is due to a recurring mistake in a crucial step in bounding the Hessian.[1]

**Lipschitzness constant.** The improved Lipschitzness constant under Assumption (E-LS) is provided in Lemma C.9 (iii) in Section C.4.5. Compared to the existing bounds, our result shows that when $\gamma$ is close to 1, the Lipschitzness constant $\Gamma$ depends on $(1-\gamma)^{-3/2}$ instead of $(1-\gamma)^{-2}$ derived in the proof of Lemma 6 in Papini et al. (2022) under the same Assumption (E-LS).

**Upper bound of the variance of the gradient estimators.** As for the result in Lemma 4.9, our bounds (4.21) on the variance of the gradient estimators REINFORCE and GPOMDP

[1] In a previous version of the proof in Sect. C, Xu et al. (2020b) rely on the identity $\nabla_\theta^2 J(\theta) = \mathbb{E}_\tau\left[\nabla_\theta g(\tau \mid \theta)\right]$, which is incorrect since the operators $\nabla_\theta$ and $\mathbb{E}[\cdot]$ are not commutative in this case as the density $p(\cdot \mid \theta)$ of $\mathbb{E}[\cdot]$ depends on $\theta$ as well. This error is recently fixed by Xu et al. (2020b) on https://arxiv.org/pdf/1909.08610.pdf in their original paper.

are slightly tighter than the one in Lemma 17 and 18 in Papini et al. (2022), see more details in Section C.4.1. Shen et al. (2019) and Pham et al. (2020) also showed that the variance of the vanilla gradient estimator with batch size $m = 1$ is bounded under more restrictive assumptions (LS). While their bounded variance depends on $(1-\gamma)^{-4}$ and they only consider the GPOMDP gradient estimator, ours (4.21) depends on $(1-\gamma)^{-3}$ for GPOMDP or $\frac{H}{(1-\gamma)^2}$ for REINFORCE which is tighter in both cases.

## C.2 Auxiliary Lemmas

**Lemma C.1.** *For all $\gamma \in [0, 1)$ and any strictly positive integer $H$, we have that*

$$\sum_{t=0}^{H-1}(t+1)\gamma^t \;\leq\; \sum_{t=0}^{\infty}(t+1)\gamma^t \;=\; \frac{1}{(1-\gamma)^2}.$$

*Proof.* The first part of the inequality is trivial. We now prove the second part of the inequality. Let

$$S \stackrel{\text{def}}{=} \sum_{t=0}^{\infty}(t+1)\gamma^t.$$

We have

$$\gamma S \;=\; \sum_{t=0}^{\infty}(t+1)\gamma^{t+1} \;=\; \sum_{t=1}^{\infty} t\gamma^t.$$

Subtracting of the above two equations gives

$$(1-\gamma)S \;=\; \sum_{t=0}^{\infty}(t+1)\gamma^t - \sum_{t=1}^{\infty} t\gamma^t \;=\; 1 + \sum_{t=1}^{\infty}(t+1-t)\gamma^t \;=\; \sum_{t=0}^{\infty}\gamma^t \;=\; \frac{1}{1-\gamma}.$$

Finally, the proof follows by dividing $1-\gamma$ on both hand side. □

**Lemma C.2.** *For all $\gamma \in [0, 1)$ and any strictly positive integer $H$, we have that*

$$\sum_{t=0}^{\infty}(t+1)^2\gamma^t \;\leq\; \frac{2}{(1-\gamma)^3}.$$

*Proof.* Let

$$S \stackrel{\text{def}}{=} \sum_{t=0}^{\infty}(t+1)^2\gamma^t.$$

We have

$$\gamma S \;=\; \sum_{t=0}^{\infty}(t+1)^2\gamma^{t+1} \;=\; \sum_{t=1}^{\infty}t^2\gamma^t.$$

Thus, the subtraction of the above two equations gives

$$\begin{aligned}
(1-\gamma)S &= \sum_{t=0}^{\infty}(t+1)^2\gamma^t - \sum_{t=1}^{\infty}t^2\gamma^t \\
&= 1 + \sum_{t=1}^{\infty}((t+1)^2 - t^2)\gamma^t \\
&= 1 + \sum_{t=1}^{\infty}(2t+1)\gamma^t \\
&= \sum_{t=0}^{\infty}(2t+1)\gamma^t \\
&= 2\sum_{t=0}^{\infty}(t+1)\gamma^t - \sum_{t=0}^{\infty}\gamma^t \\
&= \frac{2}{(1-\gamma)^2} - \frac{1}{1-\gamma} \\
&\leq \frac{2}{(1-\gamma)^2},
\end{aligned}$$

where the second last line is obtained by Lemma C.1. Finally, the proof follows by dividing $1-\gamma$ on both hand side. □

**Lemma C.3.** *The full policy gradient (4.3) can be re-written as (4.5) or (4.7). That is,*

$$\begin{aligned}
\nabla J(\theta) &= \mathbb{E}_\tau\left[\sum_{k=0}^{\infty}\gamma^k\mathcal{R}(s_k,a_k)\sum_{t=0}^{\infty}\nabla_\theta\log\pi_\theta(a_t\mid s_t)\right] \\
&= \mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\left(\sum_{k=0}^{t}\nabla_\theta\log\pi_\theta(a_k\mid s_k)\right)\gamma^t\mathcal{R}(s_t,a_t)\right] \\
&= \mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\nabla_\theta\log\pi_\theta(a_t\mid s_t)\sum_{t'=t}^{\infty}\gamma^{t'}\mathcal{R}(s_{t'},a_{t'})\right].
\end{aligned}$$

*Proof.* To simplify (4.3), we notice that future actions do not depend on past rewards. That is, for $0 \leq k < l$ among terms of the two sums in equation (4.3), we have

$$\begin{aligned}
&\mathbb{E}_{\tau}\left[\nabla_\theta \log \pi_\theta(a_l \mid s_l)\gamma^k \mathcal{R}(s_k, a_k)\right] \\
&= \mathbb{E}_{s_{0:l}, a_{0:l}}\left[\nabla_\theta \log \pi_\theta(a_l \mid s_l)\gamma^k \mathcal{R}(s_k, a_k)\right] \\
&= \mathbb{E}_{s_{0:l}, a_{0:(l-1)}}\left[\gamma^k \mathcal{R}(s_k, a_k)\mathbb{E}_{a_l}\left[\nabla_\theta \log \pi_\theta(a_l \mid s_l) \,\middle|\, s_{0:l}, a_{0:(l-1)}\right]\right] \\
&= \mathbb{E}_{s_{0:l}, a_{0:(l-1)}}\left[\gamma^k \mathcal{R}(s_k, a_k)\int \pi_\theta(a_l \mid s_l)\nabla_\theta \log \pi_\theta(a_l \mid s_l)\, da_l\right] \\
&= \mathbb{E}_{s_{0:l}, a_{0:(l-1)}}\left[\gamma^k \mathcal{R}(s_k, a_k)\int \nabla_\theta \pi_\theta(a_l \mid s_l)\, da_l\right] \\
&= \mathbb{E}_{s_{0:l}, a_{0:(l-1)}}\left[\gamma^k \mathcal{R}(s_k, a_k)\nabla_\theta \underbrace{\int \pi_\theta(a_l \mid s_l)\, da_l}_{=1}\right] = 0.
\end{aligned}$$

Plugging the above property into (4.3) yields the lemma's claim. □

**Lemma C.4.** *Under Assumption 4.8, for all non negative integer $t$ and any state-action pair $(s_t, a_t) \in \mathcal{S} \times \mathcal{A}$ at time $t$ of a trajectory $\tau \sim p(\cdot \mid \theta)$ sampled under the parametrized policy $\pi_\theta$, we have that*

$$\mathbb{E}_{\tau \sim p(\cdot|\theta)}\left[\|\nabla_\theta \log \pi_\theta(a_t \mid s_t)\|^2\right] \leq G^2, \tag{C.1}$$

$$\mathbb{E}_{\tau \sim p(\cdot|\theta)}\left[\left\|\nabla^2_\theta \log \pi_\theta(a_t \mid s_t)\right\|\right] \leq F. \tag{C.2}$$

*Proof.* For $t > 0$ and $(s_t, a_t) \in \mathcal{S} \times \mathcal{A}$, we have

$$\mathbb{E}_{\tau}\left[\|\nabla_\theta \log \pi_\theta(a_t \mid s_t)\|^2\right] = \mathbb{E}_{s_t}\left[\mathbb{E}_{a_t \sim \pi_\theta(\cdot|s_t)}\left[\|\nabla_\theta \log \pi_\theta(a_t \mid s_t)\|^2 \,|s_t\right]\right] \overset{(4.18)}{\leq} G^2,$$

where the first equality is obtained by the Markov property.

Similarly, we have

$$\mathbb{E}_{\tau}\left[\left\|\nabla^2_\theta \log \pi_\theta(a_t \mid s_t)\right\|\right] = \mathbb{E}_{s_t}\left[\mathbb{E}_{a_t \sim \pi_\theta(\cdot|s_t)}\left[\left\|\nabla^2_\theta \log \pi_\theta(a_t \mid s_t)\right\| \,|s_t\right]\right] \overset{(4.19)}{\leq} F.$$

□

**Lemma C.5.** *For all non negative integers $0 \leq h < h'$, and any state-action pairs $(s_h, a_h), (s_{h'}, a_{h'}) \in \mathcal{S} \times \mathcal{A}$ at time $h$ and $h'$ respectively of the same trajectory $\tau \sim p(\cdot \mid \theta)$ sampled under the parametrized policy $\pi_\theta$, we have*

$$\mathbb{E}_\tau \left[ \left( \nabla_\theta \log \pi_\theta(a_h \mid s_h) \right)^\top \nabla_\theta \log \pi_\theta(a_{h'} \mid s_{h'}) \right] = 0. \tag{C.3}$$

*Proof.* For $0 \leq h < h'$, we have

$$\begin{aligned}
&\mathbb{E}_\tau \left[ \left( \nabla_\theta \log \pi_\theta(a_h \mid s_h) \right)^\top \nabla_\theta \log \pi_\theta(a_{h'} \mid s_{h'}) \right] \\
&= \mathbb{E}_{a_h, s_h, s_{h'}} \left[ \mathbb{E}_{a_{h'}} \left[ \left( \nabla_\theta \log \pi_\theta(a_h \mid s_h) \right)^\top \nabla_\theta \log \pi_\theta(a_{h'} \mid s_{h'}) \middle| s_h, a_h, s_{h'} \right] \right] \\
&= \mathbb{E}_{a_h, s_h, s_{h'}} \left[ \left( \nabla_\theta \log \pi_\theta(a_h \mid s_h) \right)^\top \mathbb{E}_{a_{h'}} \left[ \nabla_\theta \log \pi_\theta(a_{h'} \mid s_{h'}) \middle| s_h, a_h, s_{h'} \right] \right] \\
&= \mathbb{E}_{a_h, s_h, s_{h'}} \left[ \left( \nabla_\theta \log \pi_\theta(a_h \mid s_h) \right)^\top \int_{a_{h'}} \pi_\theta(a_{h'} \mid s_{h'}) \nabla_\theta \log \pi_\theta(a_{h'} \mid s_{h'}) da_{h'} \right] \\
&= \mathbb{E}_{a_h, s_h, s_{h'}} \left[ \left( \nabla_\theta \log \pi_\theta(a_h \mid s_h) \right)^\top \int_{a_{h'}} \nabla_\theta \pi_\theta(a_{h'} \mid s_{h'}) da_{h'} \right] \\
&= \mathbb{E}_{a_h, s_h, s_{h'}} \left[ \left( \nabla_\theta \log \pi_\theta(a_h \mid s_h) \right)^\top \nabla_\theta \underbrace{\int_{a_{h'}} \pi_\theta(a_{h'} \mid s_{h'}) da_{h'}}_{=1} \right] = 0,
\end{aligned}$$

where the first and second equality is obtained by the Markov property. □

**Lemma C.6.** *For all non negative integers $0 \leq t$, and any state-action pairs $(s_h, a_h) \in \mathcal{S} \times \mathcal{A}$ at time $0 \leq h \leq t$ of the same trajectory $\tau \sim p(\cdot \mid \theta)$ sampled under the parametrized policy $\pi_\theta$, we have*

$$\mathbb{E}_\tau \left[ \left\| \sum_{h=0}^{t} \nabla_\theta \log \pi_\theta(a_h \mid s_h) \right\|^2 \right] = \sum_{h=0}^{t} \mathbb{E}_\tau \left[ \left\| \log \pi_\theta(a_h \mid s_h) \right\|^2 \right]. \tag{C.4}$$

*Proof.* For $0 \leq t$, we have

$$\begin{aligned}
&\mathbb{E}_\tau \left[ \left\| \sum_{h=0}^{t} \nabla_\theta \log \pi_\theta(a_h \mid s_h) \right\|^2 \right] \\
= &\sum_{h=0}^{t} \mathbb{E}_\tau \left[ \left\| \log \pi_\theta(a_h \mid s_h) \right\|^2 \right]
\end{aligned}$$

$$+ 2 \sum_{h=0}^{t-1} \sum_{h'=h+1}^{t} \mathbb{E}_\tau \left[ \left( \nabla_\theta \log \pi_\theta(a_h \mid \theta_h) \right)^\top \nabla_\theta \log \pi_\theta(a_{h'} \mid \theta_{h'}) \right]$$

$$\overset{(C.3)}{=} \sum_{h=0}^{t} \mathbb{E}_\tau \left[ \left\| \log \pi_\theta(a_h \mid s_h) \right\|^2 \right].$$

□

## C.3 Proof of Section 4.3

### C.3.1 Proof of Theorem 4.4

*Proof.* We start with $L$-smoothness of $J$ from Assumption 4.1, which implies

$$\begin{aligned} J(\theta_{t+1}) &\geq J(\theta_t) + \langle \nabla J(\theta_t), \theta_{t+1} - \theta_t \rangle - \frac{L}{2} \left\| \theta_{t+1} - \theta_t \right\|^2 \\ &= J(\theta_t) + \eta \left\langle \nabla J(\theta_t), \widehat{\nabla}_m J(\theta_t) \right\rangle - \frac{L\eta^2}{2} \left\| \widehat{\nabla}_m J(\theta_t) \right\|^2. \end{aligned}$$

Taking expectations conditioned on $\theta_t$, we get

$$\begin{aligned} \mathbb{E}_t \left[ J(\theta_{t+1}) \right] &\geq J(\theta_t) + \eta \left\langle \nabla J(\theta_t), \nabla J_H(\theta_t) \right\rangle - \frac{L\eta^2}{2} \mathbb{E}_t \left[ \left\| \widehat{\nabla}_m J(\theta_t) \right\|^2 \right] \\ &\overset{(ABC)}{\geq} J(\theta_t) + \eta \left\langle \nabla J_H(\theta_t) + \left( \nabla J(\theta_t) - \nabla J_H(\theta_t) \right), \nabla J_H(\theta_t) \right\rangle \\ &\quad - \frac{L\eta^2}{2} \left( 2A(J^* - J(\theta_t)) + B \left\| \nabla J_H(\theta_t) \right\|^2 + C \right) \\ &= J(\theta_t) + \eta \left( 1 - \frac{LB\eta}{2} \right) \left\| \nabla J_H(\theta_t) \right\|^2 - L\eta^2 A(J^* - J(\theta_t)) \\ &\quad - \frac{LC\eta^2}{2} + \eta \left\langle \nabla J_H(\theta_t), \nabla J(\theta_t) - \nabla J_H(\theta_t) \right\rangle \\ &\overset{(4.12)}{\geq} J(\theta_t) + \eta \left( 1 - \frac{LB\eta}{2} \right) \left\| \nabla J_H(\theta_t) \right\|^2 - L\eta^2 A(J^* - J(\theta_t)) \\ &\quad - \frac{LC\eta^2}{2} - \eta D \gamma^H. \end{aligned}$$

Subtracting $J^*$ from both sides gives

$$\begin{aligned} -\left( J^* - \mathbb{E}_t \left[ J(\theta_{t+1}) \right] \right) \geq -(1 + L\eta^2 A)(J^* - J(\theta_t)) + \eta \left( 1 - \frac{LB\eta}{2} \right) \left\| \nabla J_H(\theta_t) \right\|^2 \\ - \frac{LC\eta^2}{2} - \eta D \gamma^H. \end{aligned}$$

Taking the total expectation and rearranging, we get

$$\begin{aligned}&\mathbb{E}\left[J^* - J(\theta_{t+1})\right] + \eta\left(1 - \frac{LB\eta}{2}\right)\mathbb{E}\left[\|\nabla J_H(\theta_t)\|^2\right]\\ \leq\ & (1 + L\eta^2 A)\mathbb{E}\left[J^* - J(\theta_t)\right] + \frac{LC\eta^2}{2} + \eta D\gamma^H.\end{aligned}$$

Letting $\delta_t \overset{\text{def}}{=} \mathbb{E}\left[J^* - J(\theta_t)\right]$ and $r_t \overset{\text{def}}{=} \mathbb{E}\left[\|\nabla J_H(\theta_t)\|^2\right]$, we can rewrite the last inequality as

$$\eta\left(1 - \frac{LB\eta}{2}\right) r_t \leq (1 + L\eta^2 A)\delta_t - \delta_{t+1} + \frac{LC\eta^2}{2} + \eta D\gamma^H. \tag{C.5}$$

We now introduce a sequence of weights $w_{-1}, w_0, w_1, \cdots, w_{T-1}$ based on a technique developed by Stich (2019). Let $w_{-1} > 0$. Define $w_t \overset{\text{def}}{=} \frac{w_{t-1}}{1+L\eta^2 A}$ for all $t \geq 0$. Notice that if $A = 0$, we have $w_t = w_{t-1} = \cdots = w_{-1}$. Multiplying (C.5) by $w_t/\eta$,

$$\begin{aligned}\left(1 - \frac{LB\eta}{2}\right) w_t r_t &\leq \frac{w_t(1 + L\eta^2 A)}{\eta}\delta_t - \frac{w_t}{\eta}\delta_{t+1} + \frac{LC\eta}{2}w_t + D\gamma^H w_t\\ &= \frac{w_{t-1}}{\eta}\delta_t - \frac{w_t}{\eta}\delta_{t+1} + \left(\frac{LC\eta}{2} + D\gamma^H\right) w_t.\end{aligned} \tag{C.6}$$

Summing up both sides as $t = 0, 1, \cdots, T-1$ and using telescopic sum, we have,

$$\begin{aligned}\left(1 - \frac{LB\eta}{2}\right)\sum_{t=0}^{T-1} w_t r_t &\leq \frac{w_{-1}}{\eta}\delta_0 - \frac{w_{T-1}}{\eta}\delta_T + \left(\frac{LC\eta}{2} + D\gamma^H\right)\sum_{t=0}^{T-1} w_t\\ &\leq \frac{w_{-1}}{\eta}\delta_0 + \left(\frac{LC\eta}{2} + D\gamma^H\right)\sum_{t=0}^{T-1} w_t.\end{aligned} \tag{C.7}$$

Let $W_T \overset{\text{def}}{=} \sum_{t=0}^{T-1} w_t$. Dividing both sides by $W_T$, we have,

$$\left(1 - \frac{LB\eta}{2}\right)\min_{0\leq t\leq T-1} r_t \ \leq\ \frac{1}{W_T}\cdot\left(1 - \frac{LB\eta}{2}\right)\sum_{t=0}^{T-1} w_t r_t \ \leq\ \frac{w_{-1}}{W_T}\frac{\delta_0}{\eta} + \frac{LC\eta}{2} + D\gamma^H. \tag{C.8}$$

Note that,

$$W_T \ =\ \sum_{t=0}^{T-1} w_t \geq \sum_{t=0}^{T-1}\min_{0\leq i\leq T-1} w_i \ =\ T w_{T-1} \ =\ \frac{T w_{-1}}{(1 + L\eta^2 A)^T}. \tag{C.9}$$

Using this in (C.8),

$$\left(1 - \frac{LB\eta}{2}\right)\min_{0\leq t\leq T-1} r_t \leq \frac{(1 + L\eta^2 A)^T}{\eta T}\delta_0 + \frac{LC\eta}{2} + D\gamma^H. \tag{C.10}$$

However, we have

$$\begin{aligned}\mathbb{E}\left[\|\nabla J(\theta_t)\|^2\right] &= \mathbb{E}\left[\|\nabla J(\theta_t) - \nabla J_H(\theta_t) + \nabla J_H(\theta_t)\|^2\right] \\ &= \mathbb{E}\left[\|\nabla J_H(\theta_t)\|^2\right] + 2\mathbb{E}\left[\langle \nabla J_H(\theta_t), \nabla J(\theta_t) - \nabla J_H(\theta_t)\rangle\right] \\ &\quad + \mathbb{E}\left[\|\nabla J(\theta_t) - \nabla J_H(\theta_t)\|^2\right] \\ &\overset{(4.12)+(4.13)}{\leq} \mathbb{E}\left[\|\nabla J_H(\theta_t)\|^2\right] + 2D\gamma^H + D'^2\gamma^{2H}. \end{aligned} \tag{C.11}$$

Substituting $r_t$ in (C.10) by $\mathbb{E}\left[\|\nabla J(\theta_t)\|^2\right]$ and using (C.11), we get

$$\begin{aligned}\left(1 - \frac{LB\eta}{2}\right) \min_{0\leq t\leq T-1} \mathbb{E}\left[\|\nabla J(\theta_t)\|^2\right] \;\leq\; & \frac{(1+L\eta^2 A)^T}{\eta T}\delta_0 + \frac{LC\eta}{2} + D\gamma^H \\ & + \left(1 - \frac{LB\eta}{2}\right)\left(2D\gamma^H + D'^2\gamma^{2H}\right).\end{aligned}$$

Our choice of step size guarantees that no matter $B > 0$ or $B = 0$, we have $1 - \frac{LB\eta}{2} > 0$. Dividing both sides by $1 - \frac{LB\eta}{2}$ and rearranging yields the theorem's claim.

If $A = 0$, we know that $\{w_t\}_{t\geq -1}$ is a constant sequence. In this case, $W_T = Tw_{-1}$. Dividing both sides of (C.7) by $W_T$, we have,

$$\left(1 - \frac{LB\eta}{2}\right)\frac{1}{T}\sum_{t=0}^{T-1} r_t \;\leq\; \frac{\delta_0}{\eta T} + \frac{LC\eta}{2} + D\gamma^H. \tag{C.12}$$

Similarly, substituting $r_t$ in (C.12) by $\mathbb{E}\left[\|\nabla J(\theta_t)\|^2\right]$ and using (C.11), we get

$$\begin{aligned}\left(1 - \frac{LB\eta}{2}\right)\mathbb{E}\left[\|\nabla J(\theta_U)\|^2\right] &= \left(1 - \frac{LB\eta}{2}\right)\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla J(\theta_t)\|^2\right] \\ &\leq \frac{\delta_0}{\eta T} + \frac{LC\eta}{2} + D\gamma^H + \left(1 - \frac{LB\eta}{2}\right)\left(2D\gamma^H + D'^2\gamma^{2H}\right).\end{aligned}$$

Dividing both sides by $1 - \frac{LB\eta}{2}$ and rearranging yields the theorem's claim. □

### C.3.2 Proof of Corollary 4.5

*Proof.* Given $\epsilon > 0$, from Corollary 1 in Khaled and Richtárik (2023), we know that if $\eta = \min\left\{\frac{1}{\sqrt{LAT}}, \frac{1}{LB}, \frac{\epsilon}{2LC}\right\}$ and the number of iterations $T$ satisfies

$$T \;\geq\; \frac{12\delta_0 L}{\epsilon^2}\max\left\{B, \frac{12\delta_0 A}{\epsilon^2}, \frac{2C}{\epsilon^2}\right\},$$

we have

$$\frac{2\delta_0(1+L\eta^2 A)^T}{\eta T(2-LB\eta)} + \frac{LC\eta}{2-LB\eta} \;\leq\; \epsilon^2.$$

It remains to show

$$\left(\frac{2D(3-LB\eta)}{2-LB\eta} + D'^2\gamma^H\right)\gamma^H \leq \epsilon^2.$$

Besides, our choice of the step size $\eta \leq \frac{1}{LB}$ implies that $\frac{1}{2-LB\eta} \leq 1$, thus

$$\left(\frac{2D(3-LB\eta)}{2-LB\eta} + D'^2\gamma^H\right)\gamma^H \leq \left(6D + D'^2\gamma^H\right)\gamma^H.$$

Finally, it suffices to choose $H$ such that

$$\gamma^H \;\leq\; \epsilon^2 \quad\Longleftrightarrow\quad H \;\geq\; \frac{2\log\epsilon^{-1}}{\log\gamma^{-1}} \;=\; \mathcal{O}(\log\epsilon^{-1}),$$

to guarantee that $\min_{0\leq t\leq T-1}\mathbb{E}\left[\|\nabla J(\theta_t)\|^2\right] = \mathcal{O}(\epsilon^{-2})$, which concludes the proof. □

**Remark.** When $\gamma$ is close to 1, the horizon has the following property.

$$H \;=\; \frac{2\log\epsilon^{-1}}{\log\gamma^{-1}} \;=\; \mathcal{O}\left(\frac{\log\epsilon^{-1}}{1-\gamma}\right).$$

**Remark.** When $A = 0$, by following the same analysis of Corollary 4.5 applied to (4.15) in Theorem 4.4, choosing the parameters proposed in Corollary 4.5 guarantees that $\mathbb{E}\left[\|\nabla J(\theta_U)\|^2\right] = \mathcal{O}(\epsilon^2)$.

### C.3.3 Average regret convergence under the relaxed weak gradient domination assumption

When the relaxed weak gradient domination assumption (4.17) is available, it is straightforward to obtain the average regret to the global optimum convergence under the setting of Corollary 4.5.

**Corollary C.7.** *Suppose that Assumption 4.1, 4.2, 4.3 and 4.6 hold with $A = 0$. Given $\epsilon > 0$, let $\eta = \min\left\{\frac{1}{LB}, \frac{\epsilon}{2LC}\right\}$ and the horizon $H = \mathcal{O}(\log\epsilon^{-1})$. If the number of iterations $T$ satisfies*

$$T \;\geq\; \frac{12\delta_0 L}{\epsilon^2}\max\left\{B, \frac{2C}{\epsilon^2}\right\}, \tag{C.13}$$

*then $J^* - \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[J(\theta_t)\right] = \mathcal{O}(\epsilon) + \mathcal{O}(\epsilon')$.*

*Proof.* From the remark of the proof analysis of Corollary 4.5 with $A = 0$, we know that

$$\mathbb{E}\left[\|\nabla J(\theta_U)\|^2\right] = \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla J(\theta_t)\|^2\right] = \mathcal{O}(\epsilon^2).$$

From Assumption 4.2, we get

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla J_H(\theta_t)\|^2\right] = \mathcal{O}(\epsilon^2). \tag{C.14}$$

Besides, from (4.17), we obtain that

$$(\epsilon')^2 + \|\nabla J_H(\theta)\|^2 \geq \frac{(\epsilon' + \|\nabla J_H(\theta)\|)^2}{2} \geq 2\mu(J^* - J(\theta))^2. \tag{C.15}$$

Thus, by (C.14) and (C.15), we have

$$\begin{aligned}
(\epsilon')^2 + \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla J_H(\theta_t)\|^2\right] &\overset{(C.14)}{=} (\epsilon')^2 + \mathcal{O}(\epsilon^2) \\
&\overset{(C.15)}{\geq} \frac{2\mu}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[(J^* - J(\theta_t))^2\right] \\
&\geq 2\mu\mathbb{E}\left[\left(J^* - \frac{1}{T}\sum_{t=0}^{T-1}J(\theta_t)\right)^2\right] \\
&\geq 2\mu\left(J^* - \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[J(\theta_t)\right]\right)^2,
\end{aligned}$$

where the last two inequalities are obtained by applying Jensen inequality twice. By using $(a+b)^2 \geq a^2 + b^2$ with $a, b \geq 0$, we conclude that $J^* - \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[J(\theta_t)\right] = \mathcal{O}(\epsilon) + \mathcal{O}(\epsilon')$. □

### C.3.4 Global optimum convergence under the relaxed weak gradient domination assumption

In this section, we present the new global optimum convergence theory under the relaxed weak gradient domination assumption (4.17).

**Theorem C.8.** *Suppose that Assumption 4.1, 4.2, 4.3 and 4.6 hold. Given $\epsilon > 0$, define $\delta$ s.t. if $\epsilon' = 0$, set $\delta = \epsilon$, if $\epsilon' > 0$, set $\delta = \epsilon'$. Suppose that PG defined in (4.10) is run for $T > 0$ iterations with step size $(\eta_t)_t$ chosen as*

$$\eta_t = \begin{cases} \frac{1}{b} & \text{if } T \leq \frac{b}{\mu\delta} \text{ or } t \leq t_0 \\ \frac{2}{2b + \mu\delta(t - t_0)} & \text{if } T \geq \frac{b}{\mu\delta} \text{ and } t > t_0 \end{cases} \tag{C.16}$$

*with* $t_0 = \left\lceil \frac{T}{2} \right\rceil$ *and* $b = \max\{\frac{2AL}{\mu\delta}, 2BL, \mu\delta\}$. *If* $J^* - \mathbb{E}\left[J(\theta_t)\right] \geq \delta$ *for all* $t \in \{0, 1, \cdots, T-1\}$, *then*

$$J^* - \mathbb{E}\left[J(\theta_T)\right] \leq 16 \exp\left(-\frac{\mu\delta(T-1)}{2b}\right)\left(J^* - J(\theta_0)\right) + \frac{12LC}{\mu^2\delta^2 T} + \frac{26D\gamma^H}{\mu\delta} + \frac{12(\epsilon')^2(2b - LB)}{\mu^2\delta^2 T} + \frac{2\epsilon'}{\mu}, \tag{C.17}$$

*otherwise, we have*

$$\min_{t\in\{0,1,\cdots,T-1\}} J^* - \mathbb{E}\left[J(\theta_t)\right] \leq \delta.$$

**Remark.** Similar to the exact full gradient update in Thm. 4.4, notice that for the exact full gradient update, we have Asm. 4.2 and 4.3 hold with $A = C = D = 0$ and $B = 1$. Thus under the smoothness and the weak gradient domination assumption (i.e., $\epsilon' = 0$), we have

$$J^* - \mathbb{E}\left[J(\theta_T)\right] \leq 16 \exp\left(-\frac{\mu\epsilon(T-1)}{2b}\right)\left(J^* - J(\theta_0)\right).$$

With $T = \frac{1}{\epsilon}\log\left(\frac{1}{\epsilon}\right)$, we have $J^* - \mathbb{E}\left[J(\theta_T)\right] \leq \epsilon$. Thus we establish $\widetilde{\mathcal{O}}(\epsilon^{-1})$ convergence rate for the number of iterations to the global optimal. We recover the same rate for the softmax tabular policy in Theorem 4 in Mei et al. (2020) where the smoothness assumption holds and the weak gradient domination condition (4.17) holds on the path of the iterates in the exact case.

*Proof.* From (4.17), we obtain that

$$\begin{aligned} &(\epsilon')^2 + \|\nabla J_H(\theta)\|^2 \geq \frac{\left(\epsilon' + \|\nabla J_H(\theta)\|\right)^2}{2} \geq 2\mu(J^* - J(\theta))^2 \\ \implies &\|\nabla J_H(\theta)\|^2 \geq 2\mu(J^* - J(\theta))^2 - (\epsilon')^2. \end{aligned} \tag{C.18}$$

Let $t \in \{0, 1, \cdots, T-1\}$. Using the $L$-smoothness of $J$ from Assumption 4.1,

$$\begin{aligned} J^* - J(\theta_{t+1}) &\leq J^* - J(\theta_t) - \langle \nabla J(\theta_t), \theta_{t+1} - \theta_t \rangle + \frac{L}{2}\|\theta_{t+1} - \theta_t\|^2 \\ &= J^* - J(\theta_t) - \eta_t \left\langle \nabla J(\theta_t), \widehat{\nabla}_m J(\theta_t) \right\rangle + \frac{L\eta_t^2}{2}\left\|\widehat{\nabla}_m J(\theta_t)\right\|^2. \end{aligned}$$

Taking expectation conditioned on $\theta_t$ and using Assumption 4.3 and 4.6,

$$\mathbb{E}_t\left[J^* - J(\theta_{t+1})\right]$$

$$
\begin{aligned}
&\leq J^* - J(\theta_t) - \eta_t \left\langle \nabla J(\theta_t), \nabla J_H(\theta_t) \right\rangle + \frac{L\eta_t^2}{2} \mathbb{E}_t \left[ \left\| \widehat{\nabla}_m J(\theta_t) \right\|^2 \right] \\
&\overset{\text{(ABC)}}{\leq} J^* - J(\theta_t) - \eta_t \left\langle \nabla J_H(\theta_t) + (\nabla J(\theta_t) - \nabla J_H(\theta_t)), \nabla J_H(\theta_t) \right\rangle + \\
&\quad + \frac{L\eta_t^2}{2} \left( 2A(J^* - J(\theta_t)) + B \left\| \nabla J_H(\theta_t) \right\|^2 + C \right) \\
&= (1 + L\eta_t^2 A)(J^* - J(\theta_t)) - \eta_t \left( 1 - \frac{LB\eta_t}{2} \right) \left\| \nabla J_H(\theta_t) \right\|^2 + \frac{L\eta_t^2 C}{2} \\
&\quad - \eta_t \left\langle \nabla J(\theta_t) - \nabla J_H(\theta_t), \nabla J_H(\theta_t) \right\rangle \\
&\overset{\text{(C.18)}}{\leq} \left( 1 + L\eta_t^2 A \right) (J^* - J(\theta_t)) - \mu\eta_t \left( 2 - LB\eta_t \right) (J^* - J(\theta_t))^2 + \eta_t \left( 1 - \frac{LB\eta_t}{2} \right) (\epsilon')^2 \\
&\quad + \frac{L\eta_t^2 C}{2} - \eta_t \left\langle \nabla J(\theta_t) - \nabla J_H(\theta_t), \nabla J_H(\theta_t) \right\rangle \\
&\overset{\text{(4.12)}}{\leq} \left( 1 + L\eta_t^2 A \right) (J^* - J(\theta_t)) - \mu\eta_t \left( 2 - LB\eta_t \right) (J^* - J(\theta_t))^2 + \eta_t \left( 1 - \frac{LB\eta_t}{2} \right) (\epsilon')^2 \\
&\quad + \frac{L\eta_t^2 C}{2} + \eta_t D\gamma^H \\
&\leq \left( 1 + L\eta_t^2 A \right) (J^* - J(\theta_t)) - \frac{3\mu}{2} \eta_t (J^* - J(\theta_t))^2 + \eta_t \left( 1 - \frac{LB\eta_t}{2} \right) (\epsilon')^2 \\
&\quad + \frac{L\eta_t^2 C}{2} + \eta_t D\gamma^H, \qquad \text{(C.19)}
\end{aligned}
$$

where the last line is obtained by the choice of the step size $\eta_t \leq \frac{1}{b}$ with $b \geq 2LB$.

Taking total expectation and letting $r_t \overset{\text{def}}{=} \mathbb{E}\left[ J^* - J(\theta_t) \right]$ on (C.19), we have

$$
r_{t+1} \leq r_t + LA\eta_t^2 r_t - \frac{3\mu}{2} \eta_t r_t^2 + \eta_t \left( 1 - \frac{LB\eta_t}{2} \right) (\epsilon')^2 + \frac{LC}{2} \eta_t^2 + \eta_t D\gamma^H. \qquad \text{(C.20)}
$$

If there exists $t \in \{0, 1, \cdots, T-1\}$ such that $r_t < \delta$, then we are done. Alternatively if $r_t \geq \delta$ for all $t \in \{0, 1, \cdots, T-1\}$, from (C.20), we have

$$
\begin{aligned}
r_{t+1} &\leq r_t + LA\eta_t^2 r_t - \frac{3\mu\delta}{2} \eta_t r_t + \eta_t \left( 1 - \frac{LB\eta_t}{2} \right) (\epsilon')^2 + \frac{LC}{2} \eta_t^2 + \eta_t D\gamma^H \\
&\leq (1 - \mu\delta\eta_t) r_t + \eta_t \left( 1 - \frac{LB\eta_t}{2} \right) (\epsilon')^2 + \frac{LC}{2} \eta_t^2 + \eta_t D\gamma^H, \qquad \text{(C.21)}
\end{aligned}
$$

where the last line is obtained by the choice of the step size $\eta_t \leq \frac{1}{b}$ with $b \geq \frac{2LA}{\mu\delta}$. Here $1 - \mu\delta\eta_t \geq 0$ as $\eta_t \leq \frac{1}{b}$ with $b \geq \mu\delta$. We notice that (C.21) is similar to (C.99). The rest of the proof is similar to the one of Theorem C.22.

If $T \leq \frac{b}{\mu\delta}$, $\eta_t = \frac{1}{b}$. From (C.21), we have

$$
r_T \leq \left( 1 - \frac{\mu\delta}{b} \right) r_{T-1} + \frac{LC}{2b^2} + \frac{D\gamma^H}{b} + \frac{2b - LB}{2b^2} (\epsilon')^2
$$

$$\overset{(C.21)}{\leq} \left(1-\frac{\mu\delta}{b}\right)^T r_0 + \left(\frac{LC}{2b^2}+\frac{D\gamma^H}{b}+\frac{2b-LB}{2b^2}(\epsilon')^2\right)\sum_{i=0}^{T-1}\left(1-\frac{\mu\delta}{b}\right)^i$$

$$\leq \exp\left(-\frac{\mu\delta T}{b}\right) r_0 + \frac{LC}{2\mu\delta b}+\frac{D\gamma^H}{\mu\delta}+\frac{2b-LB}{2\mu\delta b}(\epsilon')^2 \tag{C.22}$$

$$\overset{T\leq\frac{b}{\mu\delta}}{\leq} \exp\left(-\frac{\mu\delta T}{b}\right) r_0 + \frac{LC}{2\mu^2\delta^2 T}+\frac{D\gamma^H}{\mu\delta}+\frac{2b-LB}{2\mu^2\delta^2 T}(\epsilon')^2. \tag{C.23}$$

If $T \geq \frac{b}{\mu\delta}$, as $\eta_t = \frac{1}{b}$ when $t \leq t_0$, from (C.22), we have

$$\begin{aligned} r_{t_0} &\leq \exp\left(-\frac{\mu\delta t_0}{b}\right) r_0 + \frac{LC}{2\mu\delta b}+\frac{D\gamma^H}{\mu\delta}+\frac{2b-LB}{2\mu\delta b}(\epsilon')^2 \\ &\leq \exp\left(-\frac{\mu\delta(T-1)}{2b}\right) r_0 + \frac{LC}{2\mu\delta b}+\frac{D\gamma^H}{\mu\delta}+\frac{2b-LB}{2\mu\delta b}(\epsilon')^2, \end{aligned} \tag{C.24}$$

where the last line is obtained by $t_0 = \left\lceil \frac{T}{2} \right\rceil \geq \frac{T-1}{2}$.

For $t > t_0$,

$$\eta_t = \frac{2}{\mu\delta\left(\frac{2b}{\mu\delta}+t-t_0\right)}.$$

From (C.21), we have

$$\begin{aligned} r_t \leq{}& \frac{\frac{2b}{\mu\delta}+t-t_0-2}{\frac{2b}{\mu\delta}+t-t_0} r_{t-1} + \frac{2LC}{\mu^2\delta^2\left(\frac{2b}{\mu\delta}+t-t_0\right)^2} + \frac{2D\gamma^H}{\mu\delta\left(\frac{2b}{\mu\delta}+t-t_0\right)} \\ &+ \frac{2(\epsilon')^2}{\mu\delta\left(\frac{2b}{\mu\delta}+t-t_0\right)}\left(1-\frac{LB}{\mu\delta\left(\frac{2b}{\mu\delta}+t-t_0\right)}\right). \end{aligned}$$

Multiplying both sides by $\left(\frac{2b}{\mu\delta}+t-t_0\right)^2$, we have

$$\begin{aligned} \left(\frac{2b}{\mu\delta}+t-t_0\right)^2 r_t \leq{}& \left(\frac{2b}{\mu\delta}+t-t_0\right)\left(\frac{2b}{\mu\delta}+t-t_0-2\right) r_{t-1} + \frac{2LC}{\mu^2\delta^2} + \frac{2D\gamma^H}{\mu\delta}\left(\frac{2b}{\mu\delta}+t-t_0\right) \\ &+ \frac{2(\epsilon')^2}{\mu\delta}\left(\frac{2b-LB}{\mu\delta}+t-t_0\right) \\ \leq{}& \left(\frac{2b}{\mu\delta}+t-t_0-1\right)^2 r_{t-1} + \frac{2LC}{\mu^2\delta^2} + \frac{2D\gamma^H}{\mu\delta}\left(\frac{2b}{\mu\delta}+t-t_0\right) \\ &+ \frac{2(\epsilon')^2}{\mu\delta}\left(\frac{2b-LB}{\mu\delta}+t-t_0\right). \end{aligned}$$

Let $w_t \overset{\text{def}}{=} \left(\frac{2b}{\mu\epsilon}+t-t_0\right)^2$. We have

$$w_t r_t \leq w_{t-1} r_{t-1} + \frac{2LC}{\mu^2\delta^2} + \frac{2D\gamma^H}{\mu\delta}\left(\frac{2b}{\mu\delta}+t-t_0\right) + \frac{2(\epsilon')^2}{\mu\delta}\left(\frac{2b-LB}{\mu\delta}+t-t_0\right).$$

Summing up for $t = t_0 + 1, \cdots, T$ and telescoping, we get,

$$\begin{aligned} w_T r_T &\le w_{t_0} r_{t_0} + \frac{2LC(T-t_0)}{\mu^2\delta^2} + \frac{2D\gamma^H}{\mu\delta}\sum_{t=t_0+1}^{T}\left(\frac{2b}{\mu\delta} + t - t_0\right) \\ &\quad + \frac{2(\epsilon')^2}{\mu\delta}\sum_{t=t_0+1}^{T}\left(\frac{2b-LB}{\mu\delta} + t - t_0\right) \\ &= \frac{4b^2}{\mu^2\delta^2} r_{t_0} + \frac{2LC(T-t_0)}{\mu^2\delta^2} + \frac{4bD(T-t_0)\gamma^H}{\mu^2\delta^2} + \frac{D\gamma^H}{\mu\delta}(T-t_0)(T-t_0+1) \\ &\quad + \frac{2(\epsilon')^2(2b-LB)(T-t_0)}{\mu^2\delta^2} + \frac{(\epsilon')^2}{\mu\delta}(T-t_0)(T-t_0+1). \end{aligned}$$

Dividing both sides by $w_T$ and using that since

$$w_T \;=\; \left(\frac{2b}{\mu\delta} + T - t_0\right)^2 \;\ge\; (T-t_0)^2,$$

we have

$$\begin{aligned} r_T &\le \frac{4b^2}{\mu^2\delta^2 w_T} r_{t_0} + \frac{2LC(T-t_0)}{\mu^2\delta^2 w_T} + \frac{4bD(T-t_0)\gamma^H}{\mu^2\delta^2 w_T} + \frac{D\gamma^H}{\mu\delta w_T}(T-t_0)(T-t_0+1) \\ &\quad + \frac{2(\epsilon')^2(2b-LB)(T-t_0)}{\mu^2\delta^2 w_T} + \frac{(\epsilon')^2}{\mu\delta w_T}(T-t_0)(T-t_0+1) \\ &\le \frac{4b^2}{\mu^2\delta^2(T-t_0)^2} r_{t_0} + \frac{2LC}{\mu^2\delta^2(T-t_0)} + \frac{4bD\gamma^H}{\mu^2\delta^2(T-t_0)} + \frac{2D\gamma^H}{\mu\delta} + \frac{2(\epsilon')^2(2b-LB)}{\mu^2\delta^2(T-t_0)} + \frac{2(\epsilon')^2}{\mu\delta}. \end{aligned}$$

By the definition of $t_0$, we have $T - t_0 \ge \frac{T}{2}$. Plugging this estimate and notice that $\frac{(\epsilon')^2}{\delta} = \epsilon'$ by the definition of $\delta$, we have

$$\begin{aligned} r_T &\le \frac{16b^2}{\mu^2\delta^2T^2} r_{t_0} + \frac{4LC + 8bD\gamma^H}{\mu^2\delta^2 T} + \frac{2D\gamma^H}{\mu\delta} + \frac{4(\epsilon')^2(2b-LB)}{\mu^2\delta^2T} + \frac{2\epsilon'}{\mu} \\ &\overset{T\ge\frac{b}{\mu\delta}}{\le} \frac{16b^2}{\mu^2\delta^2T^2} r_{t_0} + \frac{4LC}{\mu^2\delta^2T} + \frac{10D\gamma^H}{\mu\delta} + \frac{4(\epsilon')^2(2b-LB)}{\mu^2\delta^2T} + \frac{2\epsilon'}{\mu} \\ &\overset{(C.24)}{\le} \frac{16b^2}{\mu^2\delta^2T^2}\left(\exp\left(-\frac{\mu\delta(T-1)}{2b}\right) r_0 + \frac{LC}{2\mu\delta b} + \frac{D\gamma^H}{\mu\delta} + \frac{(\epsilon')^2(2b-LB)}{2\mu\delta b}\right) \\ &\quad + \frac{4LC}{\mu^2\delta^2T} + \frac{10D\gamma^H}{\mu\delta} + \frac{4(\epsilon')^2(2b-LB)}{\mu^2\delta^2T} + \frac{2\epsilon'}{\mu} \\ &\overset{T\ge\frac{b}{\mu\delta}}{\le} 16\exp\left(-\frac{\mu\delta(T-1)}{2b}\right) r_0 + \frac{8LC}{\mu^2\delta^2T} + \frac{16D\gamma^H}{\mu\delta} + \frac{8(\epsilon')^2(2b-LB)}{\mu^2\delta^2T} \\ &\quad + \frac{4LC}{\mu^2\delta^2T} + \frac{10D\gamma^H}{\mu\delta} + \frac{4(\epsilon')^2(2b-LB)}{\mu^2\delta^2T} + \frac{2\epsilon'}{\mu} \\ &= 16\exp\left(-\frac{\mu\delta(T-1)}{2b}\right) r_0 + \frac{12LC}{\mu^2\delta^2T} + \frac{26D\gamma^H}{\mu\delta} + \frac{12(\epsilon')^2(2b-LB)}{\mu^2\delta^2T} + \frac{2\epsilon'}{\mu}. \end{aligned}$$

(C.25)

It remains to take the maximum of the two bounds (C.23) and (C.25) with

$$b = \max\{\frac{2AL}{\mu\delta}, 2BL, \mu\delta\}.$$

□

### C.3.5 Proof of Corollary 4.7

*Proof.* From Theorem C.8, when $H = \mathcal{O}(\log \epsilon^{-1})$, the dominant terms in (C.17) are $\frac{12LC}{\mu^2\delta^2 T}$ and $\frac{2\epsilon'}{\mu}$. To guarantee that

$$\min_{t\in\{0,1,\cdots,T\}} J^* - \mathbb{E}\left[J(\theta_t)\right] \leq \mathcal{O}(\epsilon) + \mathcal{O}(\epsilon'),$$

it suffices to choose $T = \mathcal{O}(\delta^{-2}\epsilon^{-1})$ such that $\frac{12LC}{\mu^2\delta^2 T} = \mathcal{O}(\epsilon)$. Thus, by the definition of $\delta$, when $\epsilon' = 0$, we have $T = \mathcal{O}(\epsilon^{-3})$; when $\epsilon' > 0$, we have $T = \mathcal{O}((\epsilon')^{-2}\epsilon^{-1})$. Otherwise, from Theorem C.8, notice that $\delta \leq \epsilon + \epsilon'$, we have $\min_{t\in\{0,1,\cdots,T-1\}} J^* - \mathbb{E}\left[J(\theta_t)\right] \leq \mathcal{O}(\epsilon) + \mathcal{O}(\epsilon')$, which concludes the proof. □

## C.4 Proof of Section 4.4.1

### C.4.1 Proof of Lemma 4.9

Note that a similar result to Lemma 4.9 is given as Lemma 17 and 18 in (Papini et al., 2022). More precisely, Lemma 17 and 18 in (Papini et al., 2022) provide an upper bound of the variance of the PG estimator similar to the following result

$$\mathbb{V}\text{ar}\left[\widehat{\nabla}_m J(\theta)\right] \leq \frac{\nu}{m}.$$

We derive a slightly tighter bound

$$\mathbb{V}\text{ar}\left[\widehat{\nabla}_m J(\theta)\right] \leq \frac{\nu - \|\nabla J_H(\theta)\|}{m}.$$

This tighter bound is crucial for our work since it results in a tighter bound on $\mathbb{E}\left[\left\|\widehat{\nabla}_m J(\theta)\right\|^2\right]$ which still fits the format of (ABC). Here is the proof for Lemma 4.9.

*Proof.* Let $g(\tau \mid \theta)$ be a stochastic gradient estimator of one single sampled trajectory $\tau$. Thus $\widehat{\nabla}_m J(\theta) = \frac{1}{m}\sum_{i=1}^m g(\tau_i \mid \theta)$. Both $\widehat{\nabla}_m J(\theta)$ and $g(\tau \mid \theta)$ are unbiased estimators of $J_H(\theta)$. We have

$$\begin{aligned}
\mathbb{E}\left[\left\|\widehat{\nabla}_m J(\theta)\right\|^2\right] &= \mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=0}^{m-1} g(\tau_i \mid \theta)\right\|^2\right] \\
&= \mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=0}^{m-1} g(\tau_i \mid \theta) - \nabla J_H(\theta) + \nabla J_H(\theta)\right\|^2\right] \\
&= \|\nabla J_H(\theta)\|^2 + \mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=0}^{m-1} \left(g(\tau_i \mid \theta) - \nabla J_H(\theta)\right)\right\|^2\right] \\
&= \|\nabla J_H(\theta)\|^2 + \frac{1}{m^2}\sum_{i=0}^{m-1} \mathbb{E}\left[\|g(\tau_i \mid \theta) - \nabla J_H(\theta)\|^2\right] \\
&= \|\nabla J_H(\theta)\|^2 + \frac{1}{m}\mathbb{E}\left[\|g(\tau_1 \mid \theta) - \nabla J_H(\theta)\|^2\right] \\
&= \|\nabla J_H(\theta)\|^2 + \frac{\mathbb{E}\left[\|g(\tau_1 \mid \theta)\|^2 - \|\nabla J_H(\theta)\|^2\right]}{m},
\end{aligned} \tag{C.26}$$

where the third, the fourth and the fifth lines are all obtained by using $\nabla J_H(\theta) = \mathbb{E}\left[g(\tau_i \mid \theta)\right]$. It remains to show $\mathbb{E}_\tau\left[\|g(\tau \mid \theta)\|^2\right]$ is bounded under Assumption 4.8.

If $\widehat{\nabla}_m J(\theta)$ is a REINFORCE gradient estimator, then

$$\begin{aligned}
\mathbb{E}_\tau\left[\|g(\tau \mid \theta)\|^2\right] &\overset{(4.4)}{=} \mathbb{E}_\tau\left[\left\|\sum_{t'=0}^{H-1} \gamma^{t'} \mathcal{R}(s_{t'}, a_{t'}) \cdot \sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t)\right\|^2\right] \\
&\leq \frac{\mathcal{R}_{\max}^2}{(1-\gamma)^2}\mathbb{E}_\tau\left[\left\|\sum_{t=0}^{H-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t)\right\|^2\right] \\
&\overset{(C.4)}{=} \frac{\mathcal{R}_{\max}^2}{(1-\gamma)^2}\sum_{t=0}^{H-1} \mathbb{E}_\tau\left[\|\nabla_\theta \log \pi_\theta(a_t \mid s_t)\|^2\right] \\
&\overset{(C.1)}{\leq} \frac{HG^2\mathcal{R}_{\max}^2}{(1-\gamma)^2},
\end{aligned} \tag{C.27}$$

where the second line is obtained by using $|\mathcal{R}(s_{t'}, a_{t'})| \leq \mathcal{R}_{\max}$.

Finally, the ABC assumption holds with

$$\mathbb{E}\left[\left\|\widehat{\nabla}_m J(\theta)\right\|^2\right] \overset{(C.26)+(C.27)}{\leq} \left(1 - \frac{1}{m}\right)\|\nabla J_H(\theta)\|^2 + \frac{HG^2\mathcal{R}_{\max}^2}{m(1-\gamma)^2}.$$

If $\widehat{\nabla}_m J(\theta)$ is a GPOMDP gradient estimator, then

$$
\begin{aligned}
\mathbb{E}_\tau\left[\|g(\tau \mid \theta)\|^2\right] &\overset{(4.6)}{=} \mathbb{E}_\tau\left[\left\|\sum_{t=0}^{H-1}\gamma^{t/2}\mathcal{R}(s_t,a_t)\gamma^{t/2}\left(\sum_{k=0}^{t}\nabla_\theta \log \pi_\theta(a_k \mid s_k)\right)\right\|^2\right] \\
&\leq \mathbb{E}_\tau\left[\left(\sum_{t=0}^{H-1}\gamma^t\mathcal{R}(s_t,a_t)^2\right)\left(\sum_{k=0}^{H-1}\gamma^k\left\|\sum_{k'=0}^{k}\nabla_\theta \log \pi_\theta(a_{k'} \mid s_{k'})\right\|^2\right)\right] \\
&\leq \frac{\mathcal{R}_{\max}^2}{1-\gamma}\cdot\sum_{k=0}^{H-1}\gamma^k\mathbb{E}_\tau\left[\left\|\sum_{k'=0}^{k}\nabla_\theta \log \pi_\theta(a_{k'} \mid s_{k'})\right\|^2\right] \\
&\overset{(C.4)}{=} \frac{\mathcal{R}_{\max}^2}{1-\gamma}\cdot\sum_{k=0}^{H-1}\gamma^k\sum_{k'=0}^{k}\mathbb{E}_\tau\left[\|\nabla_\theta \log \pi_\theta(a_{k'} \mid s_{k'})\|^2\right] \\
&\overset{(C.1)}{\leq} \frac{G^2\mathcal{R}_{\max}^2}{1-\gamma}\cdot\sum_{k=0}^{H-1}\gamma^k(k+1) \\
&\leq \frac{G^2\mathcal{R}_{\max}^2}{(1-\gamma)^3},
\end{aligned}
\tag{C.28}
$$

where the second line is from the Cauchy-Schwarz inequality, the third line is obtained by using $|\mathcal{R}(s_t,a_t)| \leq \mathcal{R}_{\max}$ and the last line is obtained by Lemma C.1.

The above together with (C.26) imply that ABC assumption holds with

$$
\mathbb{E}\left[\left\|\widehat{\nabla}_m J(\theta)\right\|^2\right] \overset{(C.26)+(C.28)}{\leq} \left(1-\frac{1}{m}\right)\|\nabla J_H(\theta)\|^2 + \frac{G^2\mathcal{R}_{\max}^2}{m(1-\gamma)^3}.
$$

□

### C.4.2 Proof of Corollary 4.10

*Proof.* It is trivial that Assumption (LS) implies (E-LS). Now we show that (E-LS) is strictly weaker than (LS).

Consider a scalar-action, fixed-variance, Gaussian policy:

$$
\pi_\theta(a \mid s) \;=\; \mathcal{N}\left(a \mid \theta^\top\phi(s), \sigma^2\right) \;=\; \frac{1}{\sigma\sqrt{2\pi}}\exp\left\{-\frac{1}{2}\left(\frac{a-\theta^\top\phi(s)}{\sigma}\right)^2\right\}, \tag{C.29}
$$

where $\theta \in \mathbb{R}^d$, $\sigma > 0$ is the standard deviation, and $\phi : \mathcal{S} \to \mathcal{R}^d$ is a mapping from the state space to the feature space.

From Lemma 23 in Papini et al. (2022), the Gaussian policy (C.29) under the condition that the state feature vectors are bounded satisfies (E-LS). That is, under the condition that there exists $\varphi \geq 0$ such that $\sup_{s\in\mathcal{S}}\|\phi(s)\| \leq \varphi$. One does not require that the actions are

bounded for the Gaussian policy. This is not the case in Xu et al. (2020b) in Section D under assumptions (LS).

Besides, from Lemma 4.9, we know that Assumption (E-LS) implies (ABC). This concludes the claim of the corollary. □

### C.4.3 Proof of Lemma 4.11

*Proof.* We know that

$$
\begin{aligned}
\nabla^2 J(\theta) &\overset{(4.5)}{=} \nabla_\theta \mathbb{E}_\tau \left[ \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \left( \sum_{k=0}^{t} \nabla_\theta \log \pi_\theta(a_k \mid s_k) \right) \right] \\
&= \nabla_\theta \int p(\tau \mid \theta) \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \left( \sum_{k=0}^{t} \nabla_\theta \log \pi_\theta(a_k \mid s_k) \right) d\tau \\
&= \int \nabla_\theta p(\tau \mid \theta) \left( \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \left( \sum_{k=0}^{t} \nabla_\theta \log \pi_\theta(a_k \mid s_k) \right) \right)^\top d\tau \\
&\quad + \int p(\tau \mid \theta) \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \left( \sum_{k=0}^{t} \nabla_\theta^2 \log \pi_\theta(a_k \mid s_k) \right) d\tau \\
&= \int p(\tau \mid \theta) \nabla_\theta \log p(\tau \mid \theta) \left( \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \left( \sum_{k=0}^{t} \nabla_\theta \log \pi_\theta(a_k \mid s_k) \right) \right)^\top d\tau \\
&\quad + \int p(\tau \mid \theta) \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \left( \sum_{k=0}^{t} \nabla_\theta^2 \log \pi_\theta(a_k \mid s_k) \right) d\tau \\
&= \mathbb{E}_\tau \left[ \nabla_\theta \log p(\tau \mid \theta) \left( \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \left( \sum_{k=0}^{t} \nabla_\theta \log \pi_\theta(a_k \mid s_k) \right) \right)^\top \right] \\
&\quad + \mathbb{E}_\tau \left[ \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \left( \sum_{k=0}^{t} \nabla_\theta^2 \log \pi_\theta(a_k \mid s_k) \right) \right] \\
&\overset{(4.1)}{=} \underbrace{\mathbb{E}_\tau \left[ \sum_{t'=0}^{\infty} \nabla_\theta \log \pi_\theta(a_{t'} \mid \theta_{t'}) \left( \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \left( \sum_{k=0}^{t} \nabla_\theta \log \pi_\theta(a_k \mid s_k) \right) \right)^\top \right]}_{①} \\
&\quad \underbrace{+ \mathbb{E}_\tau \left[ \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \left( \sum_{k=0}^{t} \nabla_\theta^2 \log \pi_\theta(a_k \mid s_k) \right) \right]}_{②}. \qquad \text{(C.30)}
\end{aligned}
$$

We now bound the above two terms separately. The second term can be bounded easily. That is,

$$\begin{aligned}
\|②\| &\leq \mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma^t\left|\mathcal{R}(s_t,a_t)\right|\left(\sum_{k=0}^{t}\left\|\nabla_\theta^2\log\pi_\theta(a_k\mid s_k)\right\|\right)\right]\\
&\leq \mathcal{R}_{\max}\sum_{t=0}^{\infty}\gamma^t\left(\sum_{k=0}^{t}\mathbb{E}_\tau\left[\left\|\nabla_\theta^2\log\pi_\theta(a_k\mid s_k)\right\|\right]\right)\\
&\overset{(C.2)}{\leq} F\mathcal{R}_{\max}\sum_{t=0}^{\infty}\gamma^t(t+1)\\
&= \frac{F\mathcal{R}_{\max}}{(1-\gamma)^2},
\end{aligned}\tag{C.31}$$

where the second line is obtained by using $|\mathcal{R}(s_t,a_t)|\leq\mathcal{R}_{\max}$ and the last line is obtained by Lemma C.1.

To bound the first term, we use the following notation $x_{0:t}\overset{\text{def}}{=}(x_0,x_1,\cdots,x_t)$ with $\{x_t\}_{t\geq 0}$ a sequence of random variables. Similar to the derivation of GPOMDP, we notice that future actions do not depend on past rewards and past actions. That is, for $0\leq t<t'$ among terms of the two sums in ①, we have

$$\begin{aligned}
&\mathbb{E}_\tau\left[\nabla_\theta\log\pi_\theta(a_{t'}\mid s_{t'})\cdot\gamma^t\mathcal{R}(s_t,a_t)\left(\sum_{k=0}^{t}\nabla_\theta\log\pi_\theta(a_k\mid s_k)\right)^\top\right]\\
&=\mathbb{E}_{s_{0:t'},a_{0:t'}}\left[\nabla_\theta\log\pi_\theta(a_{t'}\mid s_{t'})\cdot\gamma^t\mathcal{R}(s_t,a_t)\left(\sum_{k=0}^{t}\nabla_\theta\log\pi_\theta(a_k\mid s_k)\right)^\top\right]\\
&=\mathbb{E}_{s_{0:t'},a_{0:(t'-1)}}\left[\mathbb{E}_{a_{t'}}\left[\nabla_\theta\log\pi_\theta(a_{t'}\mid s_{t'})\cdot\gamma^t\mathcal{R}(s_t,a_t)\left(\sum_{k=0}^{t}\nabla_\theta\log\pi_\theta(a_k\mid s_k)\right)^\top\,\middle|\, s_{0:t'},a_{0:(t'-1)}\right]\right]\\
&=\mathbb{E}_{s_{0:t'},a_{0:(t'-1)}}\left[\mathbb{E}_{a_{t'}}\left[\nabla_\theta\log\pi_\theta(a_{t'}\mid s_{t'})\,\middle|\, s_{t'}\right]\cdot\gamma^t\mathcal{R}(s_t,a_t)\left(\sum_{k=0}^{t}\nabla_\theta\log\pi_\theta(a_k\mid s_k)\right)^\top\right]\\
&=\mathbb{E}_{s_{0:t'},a_{0:(t'-1)}}\left[\int\pi_\theta(a_{t'}\mid s_{t'})\nabla_\theta\log\pi_\theta(a_{t'}\mid s_{t'})da_{t'}\cdot\gamma^t\mathcal{R}(s_t,a_t)\left(\sum_{k=0}^{t}\nabla_\theta\log\pi_\theta(a_k\mid s_k)\right)^\top\right]\\
&=\mathbb{E}_{s_{0:t'},a_{0:(t'-1)}}\left[\int\nabla_\theta\pi_\theta(a_{t'}\mid s_{t'})da_{t'}\cdot\gamma^t\mathcal{R}(s_t,a_t)\left(\sum_{k=0}^{t}\nabla_\theta\log\pi_\theta(a_k\mid s_k)\right)^\top\right]\\
&=\mathbb{E}_{s_{0:t'},a_{0:(t'-1)}}\left[\nabla_\theta\underbrace{\int\pi_\theta(a_{t'}\mid s_{t'})da_{t'}}_{=1}\cdot\gamma^t\mathcal{R}(s_t,a_t)\left(\sum_{k=0}^{t}\nabla_\theta\log\pi_\theta(a_k\mid s_k)\right)^\top\right]\\
&=0,
\end{aligned}\tag{C.32}$$

where the third equality is obtained by the Markov property. Thus, ① can be simplified. We have

$$\begin{aligned} ① &\overset{(C.32)}{=} \mathbb{E}_\tau \left[ \sum_{t'=0}^{t} \nabla_\theta \log \pi_\theta(a_{t'} \mid \theta_{t'}) \left( \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \left( \sum_{k=0}^{t} \nabla_\theta \log \pi_\theta(a_k \mid s_k) \right) \right)^\top \right] \\ &= \mathbb{E}_\tau \left[ \sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \left( \sum_{t'=0}^{t} \nabla_\theta \log \pi_\theta(a_{t'} \mid \theta_{t'}) \right) \left( \sum_{k=0}^{t} \nabla_\theta \log \pi_\theta(a_k \mid s_k) \right)^\top \right]. \end{aligned} \tag{C.33}$$

Now we can bound ① easily. That is,

$$\begin{aligned} \|①\| &\overset{(C.33)}{\leq} \mathbb{E}_\tau \left[ \sum_{t=0}^{\infty} \gamma^t \left| \mathcal{R}(s_t, a_t) \right| \left\| \sum_{t'=0}^{t} \nabla_\theta \log \pi_\theta(a_{t'} \mid \theta_{t'}) \right\|^2 \right] \\ &\leq \mathcal{R}_{\max} \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_\tau \left[ \left\| \sum_{t'=0}^{t} \nabla_\theta \log \pi_\theta(a_{t'} \mid \theta_{t'}) \right\|^2 \right] \\ &\overset{(C.4)}{=} \mathcal{R}_{\max} \sum_{t=0}^{\infty} \gamma^t \sum_{t'=0}^{t} \mathbb{E}_\tau \left[ \left\| \nabla_\theta \log \pi_\theta(a_{t'} \mid \theta_{t'}) \right\|^2 \right] \\ &\overset{(C.1)}{\leq} G^2 \mathcal{R}_{\max} \sum_{t=0}^{\infty} \gamma^t (t+1) \\ &= \frac{G^2 \mathcal{R}_{\max}}{(1-\gamma)^2}, \end{aligned} \tag{C.34}$$

where the second line is obtained by using $|\mathcal{R}(s_t, a_t)| \leq \mathcal{R}_{\max}$ and the last line is obtained by Lemma C.1.

Finally,

$$\left\| \nabla^2 J(\theta) \right\| \overset{(C.30)+(C.34)+(C.31)}{\leq} \frac{\mathcal{R}_{\max}}{(1-\gamma)^2} (G^2 + F).$$

□

### C.4.4 Proof of Lemma 4.12

*Proof.* From (4.5), we have

$$\begin{aligned} & \left\| \nabla J(\theta) - \nabla J_H(\theta) \right\|^2 \\ = & \left\| \mathbb{E}_\tau \left[ \sum_{t=H}^{\infty} \gamma^t \mathcal{R}(s_t, a_t) \left( \sum_{k=0}^{t} \nabla_\theta \log \pi_\theta(a_k \mid s_k) \right) \right] \right\|^2 \end{aligned}$$

$$
\begin{aligned}
&\leq \mathbb{E}_\tau\left[\left\|\sum_{t=H}^{\infty}\gamma^{t/2}\mathcal{R}(s_t,a_t)\gamma^{t/2}\left(\sum_{k=0}^{t}\nabla_\theta\log\pi_\theta(a_k\mid s_k)\right)\right\|^2\right]\\
&\leq \mathbb{E}_\tau\left[\left(\sum_{t=H}^{\infty}\gamma^t\mathcal{R}(s_t,a_t)^2\right)\left(\sum_{k=H}^{\infty}\gamma^k\left\|\sum_{k'=0}^{k}\nabla_\theta\log\pi_\theta(a_{k'}\mid s_{k'})\right\|^2\right)\right]\\
&\leq \frac{\mathcal{R}_{\max}^2\gamma^H}{1-\gamma}\mathbb{E}_\tau\left[\sum_{k=H}^{\infty}\gamma^k\left\|\sum_{k'=0}^{k}\nabla_\theta\log\pi_\theta(a_{k'}\mid s_{k'})\right\|^2\right]\\
&\overset{(C.4)}{=} \frac{\mathcal{R}_{\max}^2\gamma^H}{1-\gamma}\sum_{k=H}^{\infty}\gamma^k\sum_{k'=0}^{k}\mathbb{E}_\tau\left[\|\nabla_\theta\log\pi_\theta(a_{k'}\mid s_{k'})\|^2\right]\\
&\overset{(C.1)}{\leq} \frac{G^2\mathcal{R}_{\max}^2\gamma^H}{1-\gamma}\sum_{k=H}^{\infty}\gamma^k(k+1)\\
&= \frac{G^2\mathcal{R}_{\max}^2\gamma^{2H}}{1-\gamma}\sum_{k=0}^{\infty}\gamma^k(k+1+H)\\
&= \left(\frac{1}{1-\gamma}+H\right)\frac{G^2\mathcal{R}_{\max}^2\gamma^{2H}}{(1-\gamma)^2},
\end{aligned}
\tag{C.35}
$$

where the second and third lines are obtained by Jensen and Cauchy-Schwarz inequality respectively, the fourth line is obtained by using $|\mathcal{R}(s_t,a_t)|\leq\mathcal{R}_{\max}$ and the last line is obtained by Lemma C.1.

Thus

$$
D' \overset{(C.35)}{=} \frac{G\mathcal{R}_{\max}}{1-\gamma}\sqrt{\frac{1}{1-\gamma}+H}.
$$

Next, by inequality of Cauchy-Swartz we have

$$
\begin{aligned}
|\langle\nabla J_H(\theta),\nabla J_H(\theta)-\nabla J(\theta)\rangle| &\leq \|\nabla J_H(\theta)\|\,\|\nabla J_H(\theta)-\nabla J(\theta)\|\\
&\overset{(4.13)}{\leq} \|\nabla J_H(\theta)\|\cdot D'\gamma^H\\
&\leq \frac{D'G\mathcal{R}_{\max}}{(1-\gamma)^{3/2}}\gamma^H,
\end{aligned}
\tag{C.36}
$$

where the last line is obtained by Lemma C.9 (iii). Thus

$$
D \overset{(C.36)}{=} \frac{D'G\mathcal{R}_{\max}}{(1-\gamma)^{3/2}}.
$$

□

### C.4.5 Lipschitz continuity of $J(\cdot)$

In this section, we show that $J(\cdot)$ is Lipschitz-continuous under Assumption 4.8.

**Lemma C.9.** *If Assumption 4.8 holds, for any $m$ trajectories $\tau_i$ and $\theta \in \mathbb{R}^d$, we have*

*(i)* $\widehat{\nabla}_m J(\theta)$ *is $L_g$-Lipschitz continuous if conditions (LS) hold;*

*(ii) The norm of the gradient estimator squared in expectation is bounded, i.e.* $\mathbb{E}\left[\left\|\widehat{\nabla}_m J(\theta)\right\|^2\right] \leq \Gamma_g^2$.

*(iii)* $J(\cdot)$ *is $\Gamma$-Lipschitz, namely* $\|\nabla J(\theta)\| \leq \Gamma$ *with* $\Gamma = \frac{G\mathcal{R}_{\max}}{(1-\gamma)^{3/2}}$. *Similarly, we have* $\|\nabla J_H(\theta)\| \leq \Gamma$ *for the exact policy gradient of the truncated function $J_H(\cdot)$ for any horizon $H$.*

*Furthermore, if $\widehat{\nabla}_m J(\theta)$ is a REINFORCE gradient estimator, then* $L_g = \frac{HF\mathcal{R}_{\max}}{1-\gamma}$ *and* $\Gamma_g = \frac{\sqrt{H}G\mathcal{R}_{\max}}{1-\gamma}$; *if $\widehat{\nabla}_m J(\theta)$ is a GPOMDP gradient estimator, then* $L_g = \frac{F\mathcal{R}_{\max}}{(1-\gamma)^2}$ *and* $\Gamma_g = \Gamma$.

**Remark.** The Lipschitzness constant proposed in Lemma C.9 (iii) is novel. See Appendix C.1.3 for more details.

The results in Lemma C.9 (ii) match the special case of Lemma 4.9 when the mini-batch size $m = 1$. It also implies Assumption (ABC) but with a looser upper bound, which is independent to the batch size $m$. We include a proof for completeness of the properties of a general vanilla policy gradient estimator. Notice that the bound of $\mathbb{E}\left[\left\|\widehat{\nabla}_m J(\theta)\right\|^2\right]$ with GPOMDP gradient estimator is a factor of $1-\gamma$ tighter as compared to Proposition 4.2 (3) in Xu et al. (2020b) and equation (17) in Yuan et al. (2020) under more restrictive assumptions (LS).

The result with GPOMDP gradient estimator in Lemma C.9 (i) was already proposed in Proposition 4.2 in Xu et al. (2020b), but not with REINFORCE gradient estimator. We include a proof for both gradient estimators for the completeness.

*Proof.* To prove (i), let $\widehat{\nabla}_m J(\theta)$ be a REINFORCE gradient estimator. From (4.4), we have

$$\begin{aligned}
\left\|\nabla\left(\widehat{\nabla}_m J(\theta)\right)\right\| &= \left\|\frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{H-1}\left(\sum_{t'=0}^{H-1}\gamma^{t'}\mathcal{R}(s_{t'}^i, a_{t'}^i)\right)\nabla_\theta^2 \log \pi_\theta(a_t^i \mid s_t^i)\right\| \\
&\leq \frac{1}{m}\sum_{i=1}^{m}\left(\sum_{t'=0}^{H-1}\gamma^{t'}\left|\mathcal{R}(s_{t'}^i, a_{t'}^i)\right|\right)\sum_{t=0}^{H-1}\left\|\nabla_\theta^2 \log \pi_\theta(a_t^i \mid s_t^i)\right\| \\
&\leq \frac{\mathcal{R}_{\max}}{1-\gamma}\cdot\frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{H-1}\left\|\nabla_\theta^2 \log \pi_\theta(a_t^i \mid s_t^i)\right\|
\end{aligned}$$

$$\overset{\text{(LS)}}{\leq} \quad \frac{HF\mathcal{R}_{\max}}{1-\gamma},$$

where the third line is obtained by using $|\mathcal{R}(s^i_{t'}, a^i_{t'})| \leq \mathcal{R}_{\max}$. In this case, $L_g = \frac{HF\mathcal{R}_{\max}}{1-\gamma}$.

Let $\widehat{\nabla}_m J(\theta)$ be a GPOMDP gradient estimator. From (4.6), we have

$$\begin{aligned}
\left\|\nabla\left(\widehat{\nabla}_m J(\theta)\right)\right\| &= \left\|\frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{H-1}\gamma^t\mathcal{R}(s^i_t,a^i_t)\left(\sum_{k=0}^{t}\nabla^2_\theta\log\pi_\theta(a^i_k\mid s^i_k)\right)\right\| \\
&\leq \frac{1}{m}\sum_{i=1}^{m}\sum_{t=0}^{H-1}\gamma^t\left|\mathcal{R}(s^i_t,a^i_t)\right|\left(\sum_{k=0}^{t}\left\|\nabla^2_\theta\log\pi_\theta(a^i_k\mid s^i_k)\right\|\right) \\
&\leq \frac{\mathcal{R}_{\max}}{m}\sum_{i=1}^{m}\sum_{t=0}^{H-1}\gamma^t\left(\sum_{k=0}^{t}\left\|\nabla^2_\theta\log\pi_\theta(a^i_k\mid s^i_k)\right\|\right) \\
&\overset{\text{(LS)}}{\leq} F\mathcal{R}_{\max}\sum_{t=0}^{H-1}\gamma^t(t+1) \\
&\overset{\text{Lemma C.1}}{\leq} \frac{F\mathcal{R}_{\max}}{(1-\gamma)^2},
\end{aligned}$$

where similarly, the third line is obtained by using $|\mathcal{R}(s^i_t, a^i_t)| \leq \mathcal{R}_{\max}$. In this case, $L_g = \frac{F\mathcal{R}_{\max}}{(1-\gamma)^2}$.

To prove (ii), let $g(\tau \mid \theta)$ be a stochastic gradient estimator of one single sampled trajectory $\tau$. Thus $\widehat{\nabla}_m J(\theta) = \frac{1}{m}\sum_{i=1}^{m} g(\tau_i \mid \theta)$. Both $\widehat{\nabla}_m J(\theta)$ and $g(\tau \mid \theta)$ are unbiased estimators of $J_H(\theta)$. We have

$$\mathbb{E}\left[\left\|\widehat{\nabla}_m J(\theta)\right\|^2\right] \leq \mathbb{E}_\tau\left[\|g(\tau\mid\theta)\|^2\right].$$

If $\widehat{\nabla}_m J(\theta)$ is a REINFORCE gradient estimator, from (C.27), we have $\Gamma_g = \frac{\sqrt{H}G\mathcal{R}_{\max}}{1-\gamma}$. If $\widehat{\nabla}_m J(\theta)$ is a GPOMDP gradient estimator, from (C.28), we have $\Gamma_g = \frac{G\mathcal{R}_{\max}}{(1-\gamma)^{3/2}}$.

To prove (iii), we have

$$\begin{aligned}
\|\nabla J(\theta)\|^2 &\overset{(4.5)}{=} \left\|\mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma^t\mathcal{R}(s_t,a_t)\left(\sum_{k=0}^{t}\nabla_\theta\log\pi_\theta(a_k\mid s_k)\right)\right]\right\|^2 \\
&\leq \mathbb{E}_\tau\left[\left\|\sum_{t=0}^{\infty}\gamma^{t/2}\mathcal{R}(s_t,a_t)\gamma^{t/2}\left(\sum_{k=0}^{t}\nabla_\theta\log\pi_\theta(a_k\mid s_k)\right)\right\|^2\right] \\
&\leq \mathbb{E}_\tau\left[\left(\sum_{t=0}^{\infty}\gamma^t\mathcal{R}(s_t,a_t)^2\right)\left(\sum_{k=0}^{\infty}\gamma^k\left\|\sum_{k'=0}^{k}\nabla_\theta\log\pi_\theta(a_{k'}\mid s_{k'})\right\|^2\right)\right] \\
&\leq \frac{\mathcal{R}^2_{\max}}{1-\gamma}\mathbb{E}_\tau\left[\sum_{k=0}^{\infty}\gamma^k\left\|\sum_{k'=0}^{k}\nabla_\theta\log\pi_\theta(a_{k'}\mid s_{k'})\right\|^2\right]
\end{aligned}$$

$$\begin{aligned}
&\overset{(C.4)}{=} \frac{\mathcal{R}_{\max}^2}{1-\gamma}\sum_{k=0}^{\infty}\gamma^k\sum_{k'=0}^{k}\mathbb{E}_\tau\left[\|\nabla_\theta \log \pi_\theta(a_{k'} \mid s_{k'})\|^2\right] \\
&\overset{(C.1)}{\leq} \frac{G^2\mathcal{R}_{\max}^2}{1-\gamma}\sum_{k=0}^{\infty}\gamma^k(k+1) \\
&= \frac{G^2\mathcal{R}_{\max}^2}{(1-\gamma)^3},
\end{aligned}$$

where the second and third lines are obtained by Jensen and Cauchy-Schwarz inequality respectively, the fourth line is obtained by using $|\mathcal{R}(s_t, a_t)| \leq \mathcal{R}_{\max}$ and the last line is obtained by Lemma C.1.

Thus,

$$\|\nabla J(\theta)\| \leq \Gamma \quad \text{with} \quad \Gamma = \frac{G\mathcal{R}_{\max}}{(1-\gamma)^{3/2}}.$$

Similarly, we also have

$$\|\nabla J_H(\theta)\| \leq \Gamma \quad \text{with} \quad \Gamma = \frac{G\mathcal{R}_{\max}}{(1-\gamma)^{3/2}}$$

for the exact policy gradient of the truncated function $J(\cdot)$ for any horizon $H$. □

### C.4.6 Proof of Corollary 4.13

*Proof.* From Lemma 4.11, we know that $J$ is $L$-smooth. Consider policy gradient with a mini-batch sampling of size $m$. From Lemma 4.9, we have Assumption 4.3 holds with $A = 0$, $B = 1 - \frac{1}{m}$ and $C = \nu/m$. Assumption 4.2 is verified as well by Lemma 4.12 with appropriate $D$ and $D'$. By Theorem 4.4, plugging $A = 0$, $B = 1 - \frac{1}{m}$ and $C = \nu/m$ in (4.15) yields the corollary's claim with step size $\eta \in \left(0, \frac{2}{L(1-\frac{1}{m})}\right)$. □

### C.4.7 Proof of Corollary 4.14

*Proof.* Consider vanilla policy gradient with step size $\eta \in \left(0, \frac{1}{L(1-\frac{1}{m})}\right)$ and a mini-batch sampling of size $m$. We have

$$\begin{aligned}
\mathbb{E}\left[\|\nabla J(\theta_U)\|^2\right] \overset{(4.26)}{\leq}\ & \frac{2\delta_0}{\eta T\left(2 - L\eta\left(1-\frac{1}{m}\right)\right)} + \frac{L\nu\eta}{m\left(2 - L\eta\left(1-\frac{1}{m}\right)\right)} \\
& + \left(\frac{2D\left(3 - L\eta\left(1-\frac{1}{m}\right)\right)}{2 - L\eta\left(1-\frac{1}{m}\right)} + D'^2\gamma^H\right)\gamma^H
\end{aligned}$$

$$\leq \quad \frac{2\delta_0}{\eta T} + \frac{L\nu\eta}{m} + \left(6D + D'^2\gamma^H\right)\gamma^H,$$

where the second inequality is obtained by $\frac{1}{2-L\eta\left(1-\frac{1}{m}\right)} \leq 1$ with $\eta \in \left(0, \frac{1}{L\left(1-\frac{1}{m}\right)}\right)$.

To get $\mathbb{E}\left[\|\nabla J(\theta_U)\|^2\right] = \mathcal{O}(\epsilon^2)$, it suffices to have

$$\mathcal{O}(\epsilon^2) \geq \frac{2\delta_0}{\eta T} + \frac{L\nu\eta}{m} \tag{C.37}$$

and

$$\mathcal{O}(\epsilon^2) \geq \left(6D + D'^2\gamma^H\right)\gamma^H \tag{C.38}$$

respectively. To make the right hand side of (C.38) smaller than $\epsilon^2$, we need $H\gamma^H = \mathcal{O}(\epsilon^2)$. Thus, we require

$$H = \mathcal{O}\left(\log\left(\frac{1}{\epsilon}\right) / \log\left(\frac{1}{\gamma}\right)\right).$$

To make the right hand side of (C.37) smaller than $\epsilon^2$, we require

$$\frac{L\nu\eta}{m} \leq \frac{\epsilon^2}{2} \quad \Longleftrightarrow \quad \eta \leq \frac{\epsilon^2 m}{2L\nu}. \tag{C.39}$$

Similarly, for the first term of the right hand side of (C.37), we require

$$\frac{2\delta_0}{\eta T} \leq \frac{\epsilon^2}{2} \quad \Longleftrightarrow \quad \frac{4\delta_0}{\epsilon^2 T} \leq \eta. \tag{C.40}$$

Combining the above two inequalities gives

$$\frac{4\delta_0}{\epsilon^2 T} \leq \eta \leq \frac{\epsilon^2 m}{2L\nu}. \tag{C.41}$$

This implies

$$Tm \geq \frac{8\delta_0 L\nu}{\epsilon^4}. \tag{C.42}$$

The condition on the step size $\eta \in \left(0, \frac{1}{L\left(1-\frac{1}{m}\right)}\right)$ requires that the mini-batch size satisfies

$$\frac{\epsilon^2 m}{2L\nu} \leq \frac{1}{L\left(1-\frac{1}{m}\right)} \Longrightarrow m \leq \frac{2\nu}{\epsilon^2}.$$

To conclude, it suffices to choose the step size $\eta = \frac{4\delta_0}{\epsilon^2 T} = \frac{\epsilon^2 m}{2L\nu}$, a mini-batch size $m$ between 1 and $\frac{2\nu}{\epsilon^2}$, the number of iterations $T = \frac{8\delta_0 L\nu}{m\epsilon^4}$ and the fixed Horizon $H = \mathcal{O}\left(\log\left(\frac{1}{\epsilon}\right) / \log\left(\frac{1}{\gamma}\right)\right)$ so that

the inequalities (C.38), (C.39), (C.40), (C.41) and (C.42) hold, which guarantee $\mathbb{E}\left[\|\nabla J(\theta_U)\|^2\right] = \mathcal{O}(\epsilon^2)$.

Thus, the total sample complexity is

$$Tm \times H \;=\; \frac{8\delta_0 L\nu \log\left(\frac{1}{\epsilon}\right)}{\log\left(\frac{1}{\gamma}\right)\epsilon^4} \;=\; \widetilde{\mathcal{O}}(\epsilon^{-4}).$$

More precisely, from Lemma 4.11, $L = \frac{\mathcal{R}_{\max}}{(1-\gamma)^2}(G^2+F)$. When using REINFORCE gradient estimator (4.4), from Lemma 4.9, $\nu = \frac{HG^2\mathcal{R}_{\max}^2}{(1-\gamma)^2}$. Thus, when $\gamma$ is close to 1, the sample complexity is

$$\begin{aligned}\frac{8\delta_0 H^2 G^2 \mathcal{R}_{\max}^3 (G^2+F)}{(1-\gamma)^4\epsilon^4} &= \frac{8\delta_0 G^2 \mathcal{R}_{\max}^3 (G^2+F)\left(\log\left(\frac{1}{\epsilon}\right)\right)^2}{\left(\log\left(\frac{1}{\gamma}\right)\right)^2 (1-\gamma)^4\epsilon^4} \\ &= \mathcal{O}\left(\left(\log\left(\frac{1}{\epsilon}\right)\right)^2 (1-\gamma)^{-6}\epsilon^{-4}\right). \end{aligned} \tag{C.43}$$

In this case, we can choose the mini-batch size $m \in \left[1; \frac{2\nu}{\epsilon^2}\right]$, i.e. from 1 to $\mathcal{O}\left(H(1-\gamma)^{-2}\epsilon^{-2}\right)$ and the constant step size $\eta = \frac{\epsilon^2 m}{2L\nu}$ varies from $\mathcal{O}\left((1-\gamma)^2\right)$ to $\mathcal{O}\left(H^{-1}(1-\gamma)^4\epsilon^2\right)$ accordingly.

When using GPOMDP gradient estimator (4.6), from Lemma 4.9, $\nu = \frac{G^2\mathcal{R}_{\max}^2}{(1-\gamma)^3}$. Thus, when $\gamma$ is close to 1, the sample complexity is

$$\frac{8\delta_0 H G^2 \mathcal{R}_{\max}^3 (G^2+F)}{(1-\gamma)^5\epsilon^4} \;=\; \frac{8\delta_0 G^2 \mathcal{R}_{\max}^3 (G^2+F)\log\left(\frac{1}{\epsilon}\right)}{\log\left(\frac{1}{\gamma}\right)(1-\gamma)^5\epsilon^4} \;=\; \mathcal{O}\left(\log\left(\frac{1}{\epsilon}\right)(1-\gamma)^{-6}\epsilon^{-4}\right). \tag{C.44}$$

In this case, we can choose the mini-batch size $m \in \left[1; \frac{2\nu}{\epsilon^2}\right]$, i.e. from 1 to $\mathcal{O}\left((1-\gamma)^{-3}\epsilon^{-2}\right)$ and the constant step size $\eta = \frac{\epsilon^2 m}{2L\nu}$ varies from $\mathcal{O}\left((1-\gamma)^2\right)$ to $\mathcal{O}\left((1-\gamma)^5\epsilon^2\right)$ accordingly. □

**Remark.** Comparing (C.44) to (C.43), we have that the sample complexity of GPOMDP is a factor of $\log(1/\epsilon)$ smaller than that of REINFORCE.

## C.5 Proof of Section 4.4.2

In this section, $\theta \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ and denote $\theta_s \equiv (\theta_{s,a})_{a\in\mathcal{A}} \in \mathbb{R}^{|\mathcal{A}|}$. We also use the following notations

$$\pi_{s,a}(\theta) \overset{\text{def}}{=} \pi_\theta(a \mid s) \qquad \text{and} \qquad \pi_s(\theta) \overset{\text{def}}{=} \pi_\theta(\cdot \mid s) \in \Delta(\mathcal{A}) \in \mathbb{R}^{|\mathcal{A}|}.$$

### C.5.1 Preliminaries for the softmax tabular policy

Recall the softmax tabular policy given by

$$\pi_{s,a}(\theta) \overset{\text{def}}{=} \frac{\exp(\theta_{s,a})}{\sum_{a' \in \mathcal{A}} \exp(\theta_{s,a'})}. \tag{C.45}$$

From (C.45), for any $(s, a, a') \in \mathcal{S} \times \mathcal{A} \times \mathcal{A}$ with $a' \neq a$, we have immediately the following partial derivatives for the softmax tabular policy

$$\frac{\partial \pi_{s,a}(\theta)}{\partial \theta_{s,a}} = \pi_{s,a}(\theta)(1 - \pi_{s,a}(\theta)), \tag{C.46}$$

$$\frac{\partial \pi_{s,a}(\theta)}{\partial \theta_{s,a'}} = -\pi_{s,a}(\theta)\pi_{s,a'}(\theta). \tag{C.47}$$

Notice that for $s' \in \mathcal{S}$ with $s' \neq s$, we have $\frac{\partial \pi_{s,a}(\theta)}{\partial \theta_{s',a}} = 0$. From (C.46) and (C.47), we obtain respectively the gradient of $\pi_{s,a}(\theta)$ and the Jacobian of $\pi_s(\theta)$ w.r.t. $\theta_s$

$$\frac{\partial \pi_{s,a}(\theta)}{\partial \theta_s} = \left( \frac{\partial \pi_s(\theta)}{\partial \theta_{s,a}} \right)^\top = \pi_{s,a}(\theta)(\mathbf{1}_a - \pi_s(\theta)), \tag{C.48}$$

$$\frac{\partial \pi_s(\theta)}{\partial \theta_s} = \mathbf{Diag}\left(\pi_s(\theta)\right) - \pi_s(\theta)\pi_s(\theta)^\top \overset{\text{def}}{=} \mathbf{H}(\pi_s(\theta)), \tag{C.49}$$

where $\mathbf{1}_a \in \mathbb{R}^{|\mathcal{A}|}$ is a vector with zero entries except one non-zero entry 1 corresponding to the action $a$. Now from (C.48) and (C.49), we obtain respectively the gradient and the Hessian of $\log \pi_{s,a}(\theta)$ w.r.t. $\theta_s$ given by

$$\frac{\partial \log \pi_{s,a}(\theta)}{\partial \theta_s} = \mathbf{1}_a - \pi_s(\theta), \tag{C.50}$$

$$\frac{\partial^2 \log \pi_{s,a}(\theta)}{\partial \theta_s^2} = -\mathbf{H}(\pi_s(\theta)). \tag{C.51}$$

### C.5.2 Stationary point convergence of the softmax tabular policy

First we provide the proof of Lemma 4.15.

*Proof.* For any state $s \in \mathcal{S}$ and any $\theta \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, from (C.50), we have

$$\begin{aligned}
\mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ \|\nabla_\theta \log \pi_\theta(a \mid s)\|^2 \right] &= \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[ 1 + \|\pi_s(\theta)\|^2 - 2\pi_{s,a}(\theta) \right] \\
&= 1 + \|\pi_s(\theta)\|^2 - 2 \sum_{a \in \mathcal{A}} \pi_{s,a}(\theta)^2 \\
&= 1 - \|\pi_s(\theta)\|^2
\end{aligned}$$

$$\leq 1 - \frac{1}{|\mathcal{A}|}, \tag{C.52}$$

where the last line is obtained by using Cauchy-Schwarz inequality in the following

$$\|\pi_s(\theta)\|^2 \;=\; \sum_{a\in\mathcal{A}} \pi_{s,a}(\theta)^2 \;\geq\; \frac{1}{|\mathcal{A}|}\left(\sum_{a\in\mathcal{A}} \pi_{s,a}(\theta)\right)^2 \;=\; \frac{1}{|\mathcal{A}|}.$$

Thus we have $G^2 = 1 - \frac{1}{|\mathcal{A}|}$.

Besides, from Lemma 22 in Mei et al. (2020), we have $\|\mathbf{H}(\pi_s(\theta))\| \leq 1$. Thus from (C.51), we have $\left\|\nabla^2_\theta \log \pi_\theta(a \mid s)\right\| \leq 1$. Taking expectation over action, we have

$$\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}\left[\left\|\nabla^2_\theta \log \pi_\theta(a \mid s)\right\|\right] \;\leq\; 1.$$

Thus we have $F = 1$. □

**Remark.** Without expectation, for any $(s, a) \in \mathcal{S} \times \mathcal{A}$, (C.52) becomes

$$\|\nabla_\theta \log \pi_\theta(a \mid s)\|^2 \;=\; 1 + \|\pi_s(\theta)\|^2 - 2\pi_{s,a}(\theta) \;\leq\; 2, \tag{C.53}$$

where the inequality is obtained by

$$\|\pi_s(\theta)\|^2 \;=\; \sum_{a\in\mathcal{A}} \pi_{s,a}(\theta)^2 \;\leq\; \sum_{a\in\mathcal{A}} \pi_{s,a}(\theta) \;=\; 1 \tag{C.54}$$

with $\pi_{s,a}(\theta) \in [0, 1]$. This means, the softmax tabular policy satisfies (LS) condition with a bigger constant $G^2 = 2$ instead of $1 - \frac{1}{|\mathcal{A}|}$ and $F = 1$.

Lemma 4.15 immediately implies that $J(\cdot)$ with the softmax tabular policy is smooth and Lipschitz as following.

**Lemma C.10.** *$J(\cdot)$ with the softmax tabular policy is $\frac{\mathcal{R}_{\max}}{(1-\gamma)^2}\left(2 - \frac{1}{|\mathcal{A}|}\right)$-smooth and $\frac{\mathcal{R}_{\max}}{(1-\gamma)^{3/2}}\sqrt{1 - \frac{1}{|\mathcal{A}|}}$-Lipschitz.*

*Proof.* From Lemma 4.15, we know that Assumption 4.8 is satisfied with $G^2 = 1 - \frac{1}{|\mathcal{A}|}$ and $F = 1$. Thus, $J(\cdot)$ with the softmax tabular policy is smooth and Lipschitz.

Indeed, from Lemma 4.11, we obtain the smoothness constant $\frac{\mathcal{R}_{\max}}{(1-\gamma)^2}\left(2 - \frac{1}{|\mathcal{A}|}\right)$ for $J(\cdot)$; and from Lemma C.9 (iii), we obtain the Lipschitzness constant $\frac{\mathcal{R}_{\max}}{(1-\gamma)^{3/2}}\sqrt{1 - \frac{1}{|\mathcal{A}|}}$ for $J(\cdot)$. □

Now we can provide the formal statement of Corollary 4.16.

**Corollary C.11** (Formal). *For any accuracy level $\epsilon$, if we choose the mini-batch size $m$ such that $1 \leq m \leq \frac{2\nu}{\epsilon^2}$, the step size $\eta = \frac{\epsilon^2 m}{2L\nu}$ with $L = \frac{\mathcal{R}_{\max}}{(1-\gamma)^2}\left(2 - \frac{1}{|\mathcal{A}|}\right)$ and*

$$\nu = \begin{cases} \frac{H\left(1-\frac{1}{|\mathcal{A}|}\right)\mathcal{R}_{\max}^2}{(1-\gamma)^2} & \text{for REINFORCE} \\ \frac{\left(1-\frac{1}{|\mathcal{A}|}\right)\mathcal{R}_{\max}^2}{(1-\gamma)^3} & \text{for GPOMDP} \end{cases},$$

*the number of iterations $T$ such that*

$$Tm \geq \begin{cases} \frac{8\delta_0\mathcal{R}_{\max}^3\left(1-\frac{1}{|\mathcal{A}|}\right)\left(2-\frac{1}{|\mathcal{A}|}\right)}{(1-\gamma)^4\epsilon^4} \cdot H & \text{for REINFORCE} \\ \frac{8\delta_0\mathcal{R}_{\max}^3\left(1-\frac{1}{|\mathcal{A}|}\right)\left(2-\frac{1}{|\mathcal{A}|}\right)}{(1-\gamma)^5\epsilon^4} & \text{for GPOMDP} \end{cases}, \tag{C.55}$$

*and the horizon $H = \mathcal{O}\left((1-\gamma)^{-1}\log(1/\epsilon)\right)$, then $\mathbb{E}\left[\|\nabla J(\theta_U)\|^2\right] = \mathcal{O}(\epsilon^2)$.*

*Proof.* From Lemma C.10, we know that $L = \frac{\mathcal{R}_{\max}}{(1-\gamma^2)}\left(2 - \frac{1}{|\mathcal{A}|}\right)$.

From Lemma 4.9 and 4.15, we know that

$$\nu = \begin{cases} \frac{H\left(1-\frac{1}{|\mathcal{A}|}\right)\mathcal{R}_{\max}^2}{(1-\gamma)^2} & \text{for REINFORCE} \\ \frac{\left(1-\frac{1}{|\mathcal{A}|}\right)\mathcal{R}_{\max}^2}{(1-\gamma)^3} & \text{for GPOMDP} \end{cases}.$$

Plugging in $L$ and $\nu$ in Corollary 4.14 yields the corollary's claim. □

### C.5.3 Stationary point convergence of the softmax tabular policy with log barrier regularization

First we provide the proof of Lemma 4.17.

*Proof.* Let $g(\tau \mid \theta)$ be a stochastic gradient estimator of one single sampled trajectory $\tau$. Thus $\widehat{\nabla}_m J(\theta) = \frac{1}{m}\sum_{i=1}^m g(\tau_i \mid \theta)$. Both $\widehat{\nabla}_m J(\theta)$ and $g(\tau \mid \theta)$ are unbiased estimators of $J_H(\theta)$.

From (4.32), we have the following gradient estimator

$$\widehat{\nabla}_m L_\lambda(\theta) = \widehat{\nabla}_m J(\theta) + \frac{\lambda}{|\mathcal{A}||\mathcal{S}|}\sum_{s,a}\nabla_\theta \log \pi_{s,a}(\theta). \tag{C.56}$$

For a state $s \in \mathcal{S}$, from (C.50), we have

$$\begin{aligned}\frac{\lambda}{|\mathcal{A}||\mathcal{S}|}\sum_{a\in\mathcal{A}}\frac{\partial \log \pi_{s,a}(\theta)}{\partial \theta_s} &= \frac{\lambda}{|\mathcal{A}||\mathcal{S}|}\sum_{a\in\mathcal{A}}(\mathbf{1}_a - \pi_s(\theta)) \\ &= \frac{\lambda \mathbf{1}_{|\mathcal{A}|}}{|\mathcal{A}||\mathcal{S}|} - \frac{\lambda \pi_s(\theta)}{|\mathcal{S}|} \\ &= \frac{\lambda}{|\mathcal{S}|}\left(\frac{\mathbf{1}_{|\mathcal{A}|}}{|\mathcal{A}|} - \pi_s(\theta)\right), \end{aligned} \tag{C.57}$$

where $\mathbf{1}_{|\mathcal{A}|} \in \mathbb{R}^{|\mathcal{A}|}$ is a vector of all ones. Thus we have

$$\widehat{\nabla}_m L_\lambda(\theta) \overset{(C.56)+(C.57)}{=} \widehat{\nabla}_m J(\theta) + \frac{\lambda}{|\mathcal{S}|}\left(\frac{\mathbf{1}}{|\mathcal{A}|} - \Big[\pi_s(\theta)\Big]_{s\in\mathcal{S}}\right), \tag{C.58}$$

where $\mathbf{1} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ and

$$\Big[\pi_s(\theta)\Big]_{s\in\mathcal{S}} = \Big[\pi_{s_1}(\theta)\,;\,\cdots\,;\,\pi_{s_{|\mathcal{S}|}}(\theta)\Big] \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$$

is the stacking[2] of the vectors $\pi_{s_i}(\theta)$.

Next, taking expectation on the trajectories, we have

$$\begin{aligned}
& \mathbb{E}\left[\left\|\widehat{\nabla}_m L_\lambda(\theta)\right\|^2\right] \\
\overset{(C.58)}{=}\ & \mathbb{E}\left[\left\|\widehat{\nabla}_m J(\theta) + \frac{\lambda}{|\mathcal{S}|}\left(\frac{\mathbf{1}}{|\mathcal{A}|} - \Big[\pi_s(\theta)\Big]_{s\in\mathcal{S}}\right)\right\|^2\right] \\
=\ & \mathbb{E}\left[\left\|\nabla J_H(\theta) + \frac{\lambda}{|\mathcal{S}|}\left(\frac{\mathbf{1}}{|\mathcal{A}|} - \Big[\pi_s(\theta)\Big]_{s\in\mathcal{S}}\right) + \widehat{\nabla}_m J(\theta) - \nabla J_H(\theta)\right\|^2\right] \\
=\ & \|\nabla L_{\lambda,H}(\theta)\|^2 + \mathbb{E}\left[\left\|\widehat{\nabla}_m J(\theta) - \nabla J_H(\theta)\right\|^2\right] \\
\overset{(C.26)}{=}\ & \|\nabla L_{\lambda,H}(\theta)\|^2 + \frac{\mathbb{E}\left[\|g(\tau_1 \mid \theta) - \nabla J_H(\theta)\|^2\right]}{m} \\
=\ & \|\nabla L_{\lambda,H}(\theta)\|^2 \\
& + \frac{\mathbb{E}\left[\left\|g(\tau_1 \mid \theta) + \frac{\lambda}{|\mathcal{S}|}\left(\frac{\mathbf{1}}{|\mathcal{A}|} - \Big[\pi_s(\theta)\Big]_{s\in\mathcal{S}}\right) - \nabla J_H(\theta) - \frac{\lambda}{|\mathcal{S}|}\left(\frac{\mathbf{1}}{|\mathcal{A}|} - \Big[\pi_s(\theta)\Big]_{s\in\mathcal{S}}\right)\right\|^2\right]}{m} \\
=\ & \left(1 - \frac{1}{m}\right)\|\nabla L_{\lambda,H}(\theta)\|^2 + \frac{\mathbb{E}\left[\left\|g(\tau_1 \mid \theta) + \frac{\lambda}{|\mathcal{S}|}\left(\frac{\mathbf{1}}{|\mathcal{A}|} - \Big[\pi_s(\theta)\Big]_{s\in\mathcal{S}}\right)\right\|^2\right]}{m}
\end{aligned}$$

[2]Here vectors are columns by default, and given $x_1, \cdots, x_{|\mathcal{S}|} \in \mathbb{R}^{|\mathcal{A}|}$ we note $[x_1\,;\,\ldots\,;\,x_{|\mathcal{S}|}] \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ the (column) vector stacking the $x_i$'s on top of each other.

$$\leq \quad \left(1-\frac{1}{m}\right)\|\nabla L_{\lambda,H}(\theta)\|^2 + \frac{2\mathbb{E}\left[\|g(\tau_1 \mid \theta)\|^2\right] + 2\left\|\frac{\lambda}{|\mathcal{S}|}\left(\frac{\mathbf{1}}{|\mathcal{A}|} - \left[\pi_s(\theta)\right]_{s\in\mathcal{S}}\right)\right\|^2}{m}. \tag{C.59}$$

In particular, we have

$$\left\|\frac{\lambda}{|\mathcal{S}|}\left(\frac{\mathbf{1}}{|\mathcal{A}|} - \left[\pi_s(\theta)\right]_{s\in\mathcal{S}}\right)\right\|^2 \leq \frac{\lambda^2}{|\mathcal{S}|^2}\left(\frac{|\mathcal{S}||\mathcal{A}|}{|\mathcal{A}|^2} - 2\frac{|\mathcal{S}|}{|\mathcal{A}|} + |\mathcal{S}|\right) = \frac{\lambda^2}{|\mathcal{S}|}\left(1-\frac{1}{|\mathcal{A}|}\right), \tag{C.60}$$

where the inequality is obtained by using $\|\pi_s(\theta)\|^2 \leq 1$ in (C.54).

As for $\mathbb{E}\left[\|g(\tau_1 \mid \theta)\|^2\right]$, if $\widehat{\nabla}_m J(\theta)$ is a REINFORCE gradient estimator, from (C.27), we have

$$\mathbb{E}\left[\|g(\tau_1 \mid \theta)\|^2\right] \leq \frac{HG^2\mathcal{R}_{\max}^2}{(1-\gamma)^2} = \frac{H\mathcal{R}_{\max}^2\left(1-\frac{1}{|\mathcal{A}|}\right)}{(1-\gamma)^2}, \tag{C.61}$$

where the equality is obtained by Lemma 4.15 with $G^2 = \left(1-\frac{1}{|\mathcal{A}|}\right)$.

Combining (C.59), (C.60) and (C.61), we have that the REINFORCE gradient estimator $\widehat{\nabla}_m L_\lambda(\theta)$ satisfies (ABC) assumption with

$$\mathbb{E}\left[\left\|\widehat{\nabla}_m L_\lambda(\theta)\right\|^2\right] \leq \left(1-\frac{1}{m}\right)\|\nabla L_{\lambda,H}(\theta)\|^2 + \frac{2}{m}\left(1-\frac{1}{|\mathcal{A}|}\right)\left(\frac{H\mathcal{R}_{\max}^2}{(1-\gamma)^2} + \frac{\lambda^2}{|\mathcal{S}|}\right).$$

If $\widehat{\nabla}_m J(\theta)$ is a GPOMDP gradient estimator, from (C.28), we have

$$\mathbb{E}\left[\|g(\tau_1 \mid \theta)\|^2\right] \leq \frac{G^2\mathcal{R}_{\max}^2}{(1-\gamma)^3} = \frac{\mathcal{R}_{\max}^2\left(1-\frac{1}{|\mathcal{A}|}\right)}{(1-\gamma)^3}. \tag{C.62}$$

Combining (C.59), (C.60) and (C.62), we have that the GPOMDP gradient estimator $\widehat{\nabla}_m L_\lambda(\theta)$ satisfies (ABC) assumption with

$$\mathbb{E}\left[\left\|\widehat{\nabla}_m L_\lambda(\theta)\right\|^2\right] \leq \left(1-\frac{1}{m}\right)\|\nabla L_{\lambda,H}(\theta)\|^2 + \frac{2}{m}\left(1-\frac{1}{|\mathcal{A}|}\right)\left(\frac{\mathcal{R}_{\max}^2}{(1-\gamma)^3} + \frac{\lambda^2}{|\mathcal{S}|}\right).$$

Thus $\widehat{\nabla}_m L_\lambda(\theta)$ satisfies the (ABC) assumption for both REINFORCE and GPOMDP gradient estimators, which concludes the proof. □

We also verify that $L_\lambda(\cdot)$ is smooth and Lipschitz in the following lemma.

**Lemma C.12.** *$L_\lambda(\cdot)$ is $\left(\frac{\mathcal{R}_{\max}}{(1-\gamma)^2}\left(2-\frac{1}{|\mathcal{A}|}\right)+\frac{\lambda}{|\mathcal{S}|}\right)$-smooth and $\sqrt{2\left(1-\frac{1}{|\mathcal{A}|}\right)\left(\frac{\mathcal{R}_{\max}^2}{(1-\gamma)^3}+\frac{\lambda^2}{|\mathcal{S}|}\right)}$-Lipschitz.*

*Proof.* For the smoothness constant, first, from Lemma C.10, we know that $J(\cdot)$ is $\frac{\mathcal{R}_{\max}}{(1-\gamma)^2}\left(2-\frac{1}{|\mathcal{A}|}\right)$-smooth.

It remains to show the regularizer $R(\theta) \stackrel{\text{def}}{=} \frac{\lambda}{|\mathcal{A}||\mathcal{S}|}\sum_{s,a}\log\pi_\theta(a\mid s)$ is $\frac{\lambda}{|\mathcal{S}|}$-smooth. From (C.58), we have

$$\nabla R(\theta) \;=\; \frac{\lambda}{|\mathcal{S}|}\left(\frac{\mathbf{1}}{|\mathcal{A}|}-\left[\pi_s(\theta)\right]_{s\in\mathcal{S}}\right).$$

From (C.49), we have

$$\left\|\frac{\partial^2 R(\theta)}{\partial\theta_s^2}\right\| \;=\; \left\|-\frac{\lambda}{|\mathcal{S}|}\mathbf{H}(\pi_s(\theta))\right\| \;\le\; \frac{\lambda}{|\mathcal{S}|},$$

where the inequality is obtained by using $\|\mathbf{H}(\pi_s(\theta))\|\le 1$ from Lemma 22 in Mei et al. (2020).

Since $\frac{\partial^2 R(\theta)}{\partial\theta_s\partial\theta_{s'}} = 0$ for $s\neq s'$, we have that $\|\nabla^2 R(\theta)\|\le\frac{\lambda}{|\mathcal{S}|}$, which yields the smoothness constant of $L_\lambda(\cdot)$.

For the Lipschitzness constant, from (C.58), we know that

$$\begin{aligned}
\|\nabla L_\lambda(\theta)\|^2 \quad &= \quad \left\|\nabla J(\theta)+\frac{\lambda}{|\mathcal{S}|}\left(\frac{\mathbf{1}}{|\mathcal{A}|}-\left[\pi_s(\theta)\right]_{s\in\mathcal{S}}\right)\right\|^2 \\
&\le \quad 2\left\|\nabla J(\theta)\right\|^2+2\left\|\frac{\lambda}{|\mathcal{S}|}\left(\frac{\mathbf{1}}{|\mathcal{A}|}-\left[\pi_s(\theta)\right]_{s\in\mathcal{S}}\right)\right\|^2 \\
\overset{\text{Lemma C.10}}{\le} \quad & 2\left(1-\frac{1}{|\mathcal{A}|}\right)\frac{\mathcal{R}_{\max}^2}{(1-\gamma)^3}+2\left\|\frac{\lambda}{|\mathcal{S}|}\left(\frac{\mathbf{1}}{|\mathcal{A}|}-\left[\pi_s(\theta)\right]_{s\in\mathcal{S}}\right)\right\|^2 \\
\overset{(C.60)}{\le} \quad & 2\left(1-\frac{1}{|\mathcal{A}|}\right)\frac{\mathcal{R}_{\max}^2}{(1-\gamma)^3}+\frac{2\lambda^2}{|\mathcal{S}|}\left(1-\frac{1}{|\mathcal{A}|}\right) \\
= \quad & 2\left(1-\frac{1}{|\mathcal{A}|}\right)\left(\frac{\mathcal{R}_{\max}^2}{(1-\gamma)^3}+\frac{\lambda^2}{|\mathcal{S}|}\right). \qquad \text{(C.63)}
\end{aligned}$$

Thus,

$$\|\nabla L_\lambda(\theta)\| \;\le\; \sqrt{2\left(1-\frac{1}{|\mathcal{A}|}\right)\left(\frac{\mathcal{R}_{\max}^2}{(1-\gamma)^3}+\frac{\lambda^2}{|\mathcal{S}|}\right)}.$$

□

**The truncated gradient assumption in the case of $L_{\lambda,H}(\cdot)$.** As $L_\lambda(\theta)$ and $L_{\lambda,H}(\theta)$ use the same regularizer, the bias due to the truncation does not affect the regularization. Besides, from Lemma 4.15, we have that Assumption (E-LS) holds. Thus, from Lemma 4.12, Assumption 4.2 holds for $L_\lambda(\theta)$ and $L_{\lambda,H}(\theta)$ with the same constant $D$ and $D'$ in Lemma 4.12 and the constant

$G$ in Lemma 4.15. That is,

$$|\langle \nabla L_{\lambda,H}(\theta), L_{\lambda,H}(\theta) - L_\lambda(\theta)\rangle| \leq D\gamma^H, \tag{C.64}$$

$$\|\nabla L_{\lambda,H}(\theta) - L_\lambda(\theta)\| \leq D'\gamma^H, \tag{C.65}$$

with

$$D = \frac{D'\mathcal{R}_{\max}}{(1-\gamma)^{3/2}}\sqrt{1-\frac{1}{|\mathcal{A}|}}, \tag{C.66}$$

$$D' = \frac{\mathcal{R}_{\max}}{1-\gamma}\sqrt{\left(\frac{1}{1-\gamma}+H\right)\left(1-\frac{1}{|\mathcal{A}|}\right)}. \tag{C.67}$$

Similar to Corollary C.11, now we can provide the FOSP convergence of $L_\lambda(\theta)$.

**Corollary C.13.** *Consider the vanilla PG (either REINFORCE or GPOMDP) applied in $L_\lambda(\cdot)$. Let $\delta_0 \overset{def}{=} L_\lambda^* - L_\lambda(\theta_0)$ with $L_\lambda^* \overset{def}{=} \max_{\theta\in\mathbb{R}^d} L_\lambda(\theta)$. For any accuracy level $\epsilon$, if we choose the mini-batch size $m$ such that $1 \leq m \leq \frac{2\nu}{\epsilon^2}$, the step size $\eta = \frac{\epsilon^2 m}{2L\nu}$ with $L = \frac{\mathcal{R}_{\max}}{(1-\gamma)^2}\left(2-\frac{1}{|\mathcal{A}|}\right)+\frac{\lambda}{|\mathcal{S}|}$ and*

$$\nu = \begin{cases} 2\left(1-\frac{1}{|\mathcal{A}|}\right)\left(\frac{H\mathcal{R}_{\max}^2}{(1-\gamma)^2}+\frac{\lambda^2}{|\mathcal{S}|}\right) & \text{when using REINFORCE} \\ 2\left(1-\frac{1}{|\mathcal{A}|}\right)\left(\frac{\mathcal{R}_{\max}^2}{(1-\gamma)^3}+\frac{\lambda^2}{|\mathcal{S}|}\right) & \text{when using GPOMDP} \end{cases}, \tag{C.68}$$

*the number of iterations $T$ such that*

$$Tm \geq \frac{8\delta_0 L\nu}{\epsilon^4} = \mathcal{O}((1-\gamma)^{-5}\epsilon^{-4}), \tag{C.69}$$

*and the horizon $H = \mathcal{O}\left((1-\gamma)^{-1}\log(1/\epsilon)\right)$, then $\mathbb{E}\left[\|\nabla L_\lambda(\theta_U)\|^2\right] = \mathcal{O}(\epsilon^2)$.*

*Proof.* From Lemma C.12, we know that $L = \frac{\mathcal{R}_{\max}}{(1-\gamma)^2}\left(2-\frac{1}{|\mathcal{A}|}\right)+\frac{\lambda}{|\mathcal{S}|}$.

From Lemma 4.17, we know that

$$\nu = \begin{cases} 2\left(1-\frac{1}{|\mathcal{A}|}\right)\left(\frac{H\mathcal{R}_{\max}^2}{(1-\gamma)^2}+\frac{\lambda^2}{|\mathcal{S}|}\right) & \text{when using REINFORCE} \\ 2\left(1-\frac{1}{|\mathcal{A}|}\right)\left(\frac{\mathcal{R}_{\max}^2}{(1-\gamma)^3}+\frac{\lambda^2}{|\mathcal{S}|}\right) & \text{when using GPOMDP} \end{cases}.$$

Plugging in $L$ and $\nu$ in Corollary 4.14 yields the corollary's claim. □

### C.5.4 Sample complexity of high probability global optimum convergence for the softmax tabular policy with log barrier regularization

In this section, we provide the sample complexity to reach a global optimum convergence of the expected return $J(\cdot)$ in high probability for the softmax tabular policy with log barrier regularization.

Before the results, we introduce the stationary distribution

$$d_{\rho,s}(\pi^*) \overset{\text{def}}{=} \mathbb{E}_{s_0\sim\rho(\cdot),\tau\sim p(\cdot|\pi^*)}\left[(1-\gamma)\sum_{t=0}^{\infty}\gamma^t\mathbb{P}\left(s_t=s\right)\right],$$

where $\pi^*$ is the optimal policy. We refer to $\left\|\frac{d_\rho(\pi^*)}{\rho}\right\|_\infty \overset{\text{def}}{=} \max_{s\in\mathcal{S}}\frac{d_{\rho,s}(\pi^*)}{\rho(s)}$ as the distribution mismatch coefficient of $\pi$ under $\rho$ (Agarwal et al., 2021)[3]. We assume that the initial state distribution $\rho$ satisfies $\min_s \rho(s) > 0$. This assumption was adapted by Agarwal et al. (2021) to ensure that the distribution mismatch coefficient is finite.

**Corollary C.14.** *For any accuracy level $\epsilon > 0$, any probability accuracy level $\delta \in (0,1)$ and any starting state distribution $\rho$, consider the vanilla PG (either REINFORCE or GPOMDP) applied to $L_\lambda(\cdot)$. If we chose the horizon $H = \mathcal{O}\left((1-\gamma)^{-1}\log\left(1/\epsilon_{opt}\right)\log\left(1/\delta\right)\right)$, the batch size $1 \leq m \leq \frac{2\nu}{\delta\epsilon_{opt}^2}$ and the number of iterations $T$ such that $Tm \geq \frac{8(L_\lambda^* - L_\lambda(\theta_0))L\nu}{\delta^2\epsilon_{opt}^4}$, the regularization parameter $\lambda = \frac{(1-\gamma)\epsilon}{2\left\|\frac{d_\rho(\pi^*)}{\rho}\right\|_\infty}$ and*

$$\epsilon_{opt} = \frac{\lambda}{2|\mathcal{S}||\mathcal{A}|} = \frac{(1-\gamma)\epsilon}{4|\mathcal{S}||\mathcal{A}|\left\|\frac{d_\rho(\theta^*)}{\rho}\right\|_\infty} \tag{C.70}$$

*with $L, \nu$ in the setting of Corollary C.13, then we have an upper bound of the sample complexity*

$$Tm \times H = \mathcal{O}\left(\frac{|\mathcal{S}|^4|\mathcal{A}|^4\left\|\frac{d_\rho(\theta^*)}{\rho}\right\|_\infty^4}{\delta^2\epsilon^4(1-\gamma)^{10}}\cdot\log\left(1/\epsilon\right)\log\left(1/\delta\right)\right) \tag{C.71}$$

*guarantees that $J^* - J(\theta_T) \leq \epsilon$ with probability at least $1-\delta$.*

The above high probability global optimum sample complexity holds with a wide range of parameters (e.g. batch size $m$ and step size $\eta$) thanks to Corollary C.13.

[3]For simplicity, we assume that the sampling for the initial state distribution is the same as the initial state distribution appeared in the expected return $J(\cdot)$. There is no difference, compared to our results, to impose a different initial state distribution $\mu \neq \rho$ for the stochastic vanilla PG. In this case, the distribution mismatch coefficient will be $\left\|\frac{d_\rho(\pi^*)}{\mu}\right\|_\infty$.

We need the following result to link the stationary point convergence of $L_\lambda(\cdot)$ to the suboptimality gap convergence $J^* - J(\cdot)$ when the norm of the gradient of a stationary point and the regularization parameter $\lambda$ are sufficiently small.

**Proposition C.15** (Theorem 5.2 in Agarwal et al. (2021)). *Suppose $\theta$ is such that $\|\nabla L_\lambda(\theta)\| \leq \frac{\lambda}{2|\mathcal{S}||\mathcal{A}|}$, then for every initial distribution $\rho$, we have*

$$J^* - J(\theta) \;\leq\; \frac{2\lambda}{1-\gamma}\left\|\frac{d_\rho(\theta^*)}{\rho}\right\|_\infty . \tag{C.72}$$

By leveraging Proposition C.15, we now derive the proof for Corollary C.14.

*Proof.* From Corollary C.13 we have that $\mathbb{E}\left[\|\nabla L_\lambda(\theta_U)\|^2\right] \leq \delta\epsilon_{opt}^2$,

Thus, there exists $t_0 \in \{0, \cdots, T-1\}$ s.t. $\mathbb{E}\left[\|\nabla L_\lambda(\theta_{t_o})\|^2\right] \leq \mathbb{E}\left[\|\nabla L_\lambda(\theta_U)\|^2\right] \leq \delta\epsilon_{opt}^2$.

From Proposition C.15, we know that if $\|\nabla L_\lambda(\theta_{t_o})\| \leq \epsilon_{opt}$, we have

$$J^* - J(\theta_{t_0}) \;\leq\; \frac{2\lambda}{1-\gamma}\left\|\frac{d_\rho(\theta^*)}{\rho}\right\|_\infty \;=\; \epsilon.$$

Thus, we have

$$\mathbb{P}(J^* - J(\theta_{t_0}) \leq \epsilon) \;\geq\; \mathbb{P}\left(\|\nabla L_\lambda(\theta_{t_o})\| \leq \epsilon_{opt}\right). \tag{C.73}$$

Consequently, we have

$$\begin{aligned}
\mathbb{P}(J^* - J(\theta_{t_0}) \geq \epsilon) \;&=\; 1 - \mathbb{P}(J^* - J(\theta_{t_0}) \leq \epsilon) \\
&\overset{(C.73)}{\leq}\; 1 - \mathbb{P}\left(\|\nabla L_\lambda(\theta_{t_o})\| \leq \epsilon_{opt}\right) \\
&=\; \mathbb{P}\left(\|\nabla L_\lambda(\theta_{t_o})\| \geq \epsilon_{opt}\right) \\
&=\; \mathbb{P}\left(\|\nabla L_\lambda(\theta_{t_o})\|^2 \geq \epsilon_{opt}^2\right) \\
&\leq\; \frac{\mathbb{E}\left[\|\nabla L_\lambda(\theta_{t_o})\|^2\right]}{\epsilon_{opt}^2} \qquad \text{(by Markov's inequality)} \\
&\leq\; \delta.
\end{aligned} \tag{C.74}$$

Since $t_0 m \leq Tm$, we conclude that the upper bound of the sample complexity is

$$Tm \times H \;\geq\; \frac{8(J^* - J(\theta_0))L\nu}{\delta^2\epsilon_{opt}^4} \times H \;=\; \mathcal{O}\left(\frac{|\mathcal{S}|^4|\mathcal{A}|^4\left\|\frac{d_\rho(\theta^*)}{\rho}\right\|_\infty^4}{\delta^2\epsilon^4(1-\gamma)^{10}} \cdot \log(1/\epsilon)\log(1/\delta)\right).$$

□

**Remark.** Following the proof of Corollary C.14, we can also deduce the iteration complexity of the exact full gradient updates for the global optimum convergence.

Indeed, from Lemma C.12, $L_\lambda(\cdot)$ is smooth. From Theorem 4.4, we know that with the number of iterations

$$T \geq \frac{12\delta_0 L}{\epsilon_{opt}^2} = \mathcal{O}\left(\frac{\delta_0}{(1-\gamma)^4\epsilon^2}\right), \tag{C.75}$$

we have $\min_{0\leq t\leq T-1} \|\nabla L_\lambda(\theta_t)\|^2 \leq \epsilon_{opt}^2$ for the exact full gradient updates.

From Proposition C.15, we have $\min_{0\leq t\leq T-1} J^* - J(\theta_t) \leq \epsilon$.

Compared to the iteration complexity in Corollary 5.1 in Agarwal et al. (2021), ours (C.75) is improved by a factor of $1-\gamma$ thanks to an improved analysis of the smoothness constant in Lemma C.12.

### C.5.5 Sample complexity of the average regret convergence for softmax tabular policy with log barrier regularization

By leveraging Proposition C.15, we now derive the proof for Corollary 4.18.

*Proof.* We define the following set of "bad" iterates based on a technique developed by Zhang et al. (2021b)

$$I^+ \stackrel{\text{def}}{=} \left\{ t \in \{0, \cdots, T-1\} \;\middle|\; \|\nabla L_\lambda(\theta_t)\| \geq \frac{\lambda}{2|\mathcal{S}||\mathcal{A}|} \right\} \tag{C.76}$$

with

$$\lambda = \frac{(1-\gamma)\epsilon}{2\left\|\frac{d_\rho(\theta^*)}{\mu}\right\|_\infty}. \tag{C.77}$$

We have

$$\begin{aligned} J^* - \frac{1}{T}\sum_{t=0}^{T-1} J(\theta_t) &= \frac{1}{T}\sum_{t\in I^+} J^* - J(\theta_t) + \frac{1}{T}\sum_{t\notin I^+} J^* - J(\theta_t) \\ &\leq \frac{|I^+|}{T}\cdot\frac{2\mathcal{R}_{\max}}{1-\gamma} + \frac{1}{T}\sum_{t\notin I^+} J^* - J(\theta_t) \\ &\leq \frac{|I^+|}{T}\cdot\frac{2\mathcal{R}_{\max}}{1-\gamma} + \frac{T-|I^+|}{T}\cdot\frac{2\lambda}{1-\gamma}\left\|\frac{d_\rho(\theta^*)}{\rho}\right\|_\infty \end{aligned}$$

$$\begin{aligned} &\leq \frac{|I^+|}{T} \cdot \frac{2\mathcal{R}_{\max}}{1-\gamma} + \frac{2\lambda}{1-\gamma} \left\| \frac{d_\rho(\theta^*)}{\rho} \right\|_\infty \\ &\overset{(C.77)}{=} \frac{|I^+|}{T} \cdot \frac{2\mathcal{R}_{\max}}{1-\gamma} + \epsilon. \end{aligned} \tag{C.78}$$

where the second line is obtained as $|J(\cdot)| \leq \frac{\mathcal{R}_{\max}}{1-\gamma}$ and the third line is obtained by Proposition C.15.

It remains to bound $|I^+|$. In fact,

$$\begin{aligned} \sum_{t=0}^{T-1} \|\nabla L_\lambda(\theta_t)\|^2 &\geq \sum_{t \in I^+} \|\nabla L_\lambda(\theta_t)\|^2 \\ &\geq \frac{|I^+|\lambda^2}{4|\mathcal{S}|^2|\mathcal{A}|^2}. \end{aligned}$$

Thus, we have

$$\begin{aligned} \frac{|I^+|}{T} &\leq \frac{4|\mathcal{S}|^2|\mathcal{A}|^2}{\lambda^2} \cdot \frac{1}{T} \sum_{t=0}^{T-1} \|\nabla L_\lambda(\theta_t)\|^2 \\ &\overset{(C.77)}{=} \frac{16 \left\| \frac{d_\rho(\theta^*)}{\mu} \right\|_\infty^2 |\mathcal{S}|^2|\mathcal{A}|^2}{(1-\gamma)^2\epsilon^2} \cdot \frac{1}{T} \sum_{t=0}^{T-1} \|\nabla L_\lambda(\theta_t)\|^2 . \end{aligned} \tag{C.79}$$

Thus, we have

$$J^* - \frac{1}{T} \sum_{t=0}^{T-1} J(\theta_t) \overset{(C.78)+(C.79)}{\leq} \frac{32\mathcal{R}_{\max} \left\| \frac{d_\rho(\theta^*)}{\rho} \right\|_\infty^2 |\mathcal{S}|^2|\mathcal{A}|^2}{(1-\gamma)^3\epsilon^2} \cdot \frac{1}{T} \sum_{t=0}^{T-1} \|\nabla L_\lambda(\theta_t)\|^2 + \epsilon. \tag{C.80}$$

Taking expectation over the iterations on both side, we have

$$J^* - \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[J(\theta_t)\right] \overset{(C.78)+(C.79)}{\leq} \frac{32\mathcal{R}_{\max} \left\| \frac{d_\rho(\theta^*)}{\rho} \right\|_\infty^2 |\mathcal{S}|^2|\mathcal{A}|^2}{(1-\gamma)^3\epsilon^2} \cdot \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla L_\lambda(\theta_t)\|^2\right] + \epsilon. \tag{C.81}$$

It suffices to have $\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla L_\lambda(\theta_t)\|^2\right] \leq (1-\gamma)^3\epsilon^3$ to guarantee that $J^* - \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[J(\theta_t)\right] \leq \mathcal{O}(\epsilon)$.

From Corollary 4.14, consider the batch size $m$ such that $1 \leq m \leq \frac{2\nu}{(1-\gamma)^3\epsilon^3} = \mathcal{O}\left(\frac{1}{(1-\gamma)^6\epsilon^3}\right)$, the step size $\mathcal{O}(\epsilon^3) \leq \eta = \frac{(1-\gamma)^3\epsilon^3 m}{2L\nu} \leq \mathcal{O}(1)$ with $L, \nu$ in the setting of Corollary C.13 . If the horizon $H = \mathcal{O}\left(\frac{\log(1/\epsilon)}{1-\gamma}\right)$ and the number of iterations $T$ is such that

$$Tm \times H \geq \frac{8(J^* - J(\theta_0))L\nu}{(1-\gamma)^6\epsilon^6} \times H = \tilde{\mathcal{O}}\left(\frac{1}{(1-\gamma)^{12}\epsilon^6}\right),$$

we have $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla L_\lambda(\theta_t)\|^2\right] \leq (1-\gamma)^3\epsilon^3$, which conclude the proof. □

## C.6 Proof of Section 4.4.3

First, we give the definition of the advantage function $A^{\pi_\theta}$ induced by the policy $\pi_\theta$ appeared in the transferred compatible function approximation error in Assumption 4.20. To do this, given a policy $\pi$, we define the state-action value function $Q^\pi : \mathcal{S}\times\mathcal{A}\to\mathbb{R}$ as

$$Q^\pi(s,a) \stackrel{\text{def}}{=} \mathbb{E}_{a_t\sim\pi(\cdot|s_t),s_{t+1}\sim\mathcal{P}(\cdot|s_t,a_t)}\left[\sum_{t=0}^{\infty}\gamma^t\mathcal{R}(s_t,a_t)\,\middle|\, s_0=s,a_0=a\right].$$

From this, the state-value function $V^\pi : \mathcal{S}\to\mathbb{R}$ and the advantage function $A^\pi : \mathcal{S}\times\mathcal{A}\to\mathbb{R}$, under the policy $\pi$, can be defined as

$$V^\pi(s) \stackrel{\text{def}}{=} \mathbb{E}_{a\sim\pi(\cdot|s)}\left[Q^\pi(s,a)\right],$$
$$A^\pi(s,a) \stackrel{\text{def}}{=} Q^\pi(s,a) - V^\pi(s).$$

Before presenting the sample complexity of the average regret convergence and the proof of Corollary 4.21 for Fisher-non-degenerate parametrized policy, we need the following result to show that Fisher-non-degenerate parametrized policy satisfies the relaxed weak gradient domination assumption.

**Proposition C.16** (Lemma 4.7 in Ding et al. (2022)). *If the policy $\pi_\theta$ satisfies Assumption 4.8, 4.19 and 4.20, then*

$$\frac{\mu_F\sqrt{\epsilon_{bias}}}{(1-\gamma)G} + \|\nabla J_H(\theta)\| \geq \frac{\mu_F}{G}(J^* - J(\theta)). \tag{C.82}$$

**Remark.** Here we use the weaker assumption (E-LS) instead of (LS) compared to the original Lemma 4.7 in Ding et al. (2022). The relaxed weak gradient domination property still holds. The proof essentially follows the same arguments and thus is omitted here.

### C.6.1 Sample complexity of the average regret convergence for Fisher-non-degenerate policy

Consequently, it is straightforward to obtain the average regret to the global optimum convergence under the setting of Corollary 4.14 for Fisher-non-degenerate parametrized policy.

**Corollary C.17.** *Assume that the policy $\pi_\theta$ satisfies Asm. 4.8, 4.19 and 4.20. For a given $\epsilon > 0$, by choosing the mini-batch size $m$ such that $1 \leq m \leq \frac{2\nu}{\epsilon^2}$, the step size $\eta = \frac{\epsilon^2 m}{2L\nu}$, the number of iterations $T$ such that*

$$Tm \geq \frac{8\delta_0 L\nu}{\epsilon^4} = \begin{cases} \mathcal{O}\left(\frac{H}{(1-\gamma)^4\epsilon^4}\right) & \text{for REINFORCE} \\ \mathcal{O}\left(\frac{1}{(1-\gamma)^5\epsilon^4}\right) & \text{for GPOMDP} \end{cases} \tag{C.83}$$

*and the horizon $H = \mathcal{O}\left((1-\gamma)^{-1}\log(1/\epsilon)\right)$, then $J^* - \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[J(\theta_t)\right] = \mathcal{O}(\epsilon) + \mathcal{O}(\sqrt{\epsilon_{bias}})$.*

**Remark.** The sample complexity $\widetilde{\mathcal{O}}(\epsilon^{-4})$ of the average regret is also shown in Theorem 4.6 in Liu et al. (2020). However, Liu et al. (2020) use the more restrictive assumption (LS) and require large batch size $m = \mathcal{O}(\epsilon^{-2})$. We improve upon them by using weaker assumption E-LS, allowing much wider range of choices for the batch size $m \in \left[1; \frac{2\nu}{\epsilon^2}\right]$ and the constant step size $\eta$ to achieve the same optimal sample complexity $\widetilde{\mathcal{O}}\left(\epsilon^{-4}\right)$.

*Proof.* From Corollary 4.14, we know that $\mathbb{E}\left[\|\nabla J(\theta_U)\|^2\right] = \mathcal{O}(\epsilon^2)$. However, from Proposition C.16, we know that Assumption 4.6 is satisfied. Thus, by doing a similar analysis as in Corollary C.7, we conclude the proof. □

### C.6.2 Proof of Corollary 4.21

Now we provide the proof of Corollary 4.21.

*Proof.* From Proposition C.16, we have that Assumption 4.6 holds. Also because of Assumption (E-LS), we have Lemmas 4.9, 4.11 and 4.12 hold. Finally, by Corollary 4.7, this directly concludes the proof. □

## C.7 FOSP convergence analysis for the softmax with entropy regularization.

In this section, we study stochastic gradient ascent on the softmax tabular policy with entropy regularization, which is

$$\tilde{J}(\theta) \overset{\text{def}}{=} J(\theta) + \mathbb{H}(\theta) \tag{C.84}$$

where $\mathbb{H}(\theta)$ is the "discounted entropy" defined as

$$\mathbb{H}(\theta) \overset{\text{def}}{=} \mathbb{E}_{\tau\sim p(\cdot|\theta)}\left[\sum_{t=0}^{\infty} -\gamma^t \lambda \log \pi_{s_t,a_t}(\theta)\right].$$

Using the same technique to derive the full gradient of the expected return (4.3), we have

$$\begin{aligned}
\nabla\tilde{J}(\theta) &= \nabla J(\theta) - \lambda\mathbb{E}_\tau\left[\nabla \log p(\tau \mid \theta)\sum_{t=0}^{\infty}\gamma^t \log \pi_{s_t,a_t}(\theta)\right] \\
&\quad -\lambda\mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma^t\nabla_\theta \log \pi_{s_t,a_t}(\theta)\right] \\
&\overset{(4.1)}{=} \nabla J(\theta) - \lambda\mathbb{E}_\tau\left[\sum_{k=0}^{\infty}\nabla_\theta \log \pi_{s_k,a_k}(\theta)\sum_{t=0}^{\infty}\gamma^t \log \pi_{s_t,a_t}(\theta)\right] \\
&\quad -\lambda\mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma^t\nabla_\theta \log \pi_{s_t,a_t}(\theta)\right] \\
&= \nabla J(\theta) - \lambda\mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma^t \log \pi_{s_t,a_t}(\theta)\left(\sum_{k=0}^{t}\nabla_\theta \log \pi_{s_k,a_k}(\theta)\right)\right] \\
&\quad -\lambda\mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma^t\nabla_\theta \log \pi_{s_t,a_t}(\theta)\right] \\
&\overset{(4.5)}{=} \mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma^t\left(\left(\mathcal{R}(s_t,a_t) - \lambda \log \pi_{s_t,a_t}(\theta)\right)\left(\sum_{k=0}^{t}\nabla_\theta \log \pi_{s_k,a_k}(\theta)\right)\right.\right. \\
&\quad \left.\left. -\lambda\nabla_\theta \log \pi_{s_t,a_t}(\theta)\right)\right],
\end{aligned} \tag{C.85}$$

where the third line is obtained by using the fact that for any $0 \le t < k$, we have

$$\mathbb{E}_\tau\left[\log \pi_{s_t,a_t}(\theta)\nabla_\theta \log \pi(s_k,a_k)(\theta)\right] = 0. \tag{C.86}$$

Equation (C.86) is derived by following the same proof technique of Lemma C.5.

Thus, the stochastic gradient estimator of $\nabla\tilde{J}(\theta)$ with mini-batch size $m$ is

$$\widehat{\nabla}_m\tilde{J}(\theta) \overset{\text{def}}{=} \widehat{\nabla}_m J(\theta) - \frac{\lambda}{m}\sum_{i=1}^{m}\sum_{t=0}^{H-1}\gamma^t\left(\log \pi_{s_t^i,a_t^i}(\theta)\left(\sum_{k=0}^{t}\nabla_\theta \log \pi_{s_k^i,a_k^i}(\theta)\right) + \nabla_\theta \log \pi_{s_t^i,a_t^i}(\theta)\right). \tag{C.87}$$

Notice that $\widehat{\nabla}_m\tilde{J}(\cdot)$ is the unbiased gradient estimator of the truncated function

$$\tilde{J}_H(\theta) \overset{\text{def}}{=} \mathbb{E}_\tau\left[\sum_{t=0}^{H-1}\gamma^t\left(\mathcal{R}(s_t,a_t) - \lambda \log \pi_{s_t,a_t}(\theta)\right)\right]. \tag{C.88}$$

We show that $\widehat{\nabla}_m \tilde{J}(\cdot)$ satisfies the (ABC) assumption as following.

**Lemma C.18.** *The stochastic gradient estimator* (C.87) *satisfies Assumption* (ABC) *with*

$$\mathbb{E}\left[\left\|\widehat{\nabla}_m \tilde{J}(\theta)\right\|^2\right] \leq \left(1-\frac{1}{m}\right)\left\|\nabla \tilde{J}(\theta)\right\|^2 + \frac{2\left(1-\frac{1}{|\mathcal{A}|}\right)\mathcal{R}_{\max}^2}{m(1-\gamma)^3} + \frac{2\lambda^2}{m(1-\gamma^2)}\left(1-\frac{1}{|\mathcal{A}|}\right) + \frac{8H|\mathcal{A}|\lambda^2}{m(1-\gamma)^3}.$$

*Proof.* Let $g(\tau \mid \theta)$ be a stochastic gradient estimator of one single sampled trajectory $\tau$ of $\nabla J_H(\theta)$. Thus $\widehat{\nabla}_m J(\theta) = \frac{1}{m}\sum_{i=1}^m g(\tau_i \mid \theta)$. Both $\widehat{\nabla}_m J(\theta)$ and $g(\tau \mid \theta)$ are unbiased estimators of $J_H(\theta)$.

Similarly, let $\tilde{g}(\tau \mid \theta)$ be a stochastic gradient estimator of one single sampled trajectory $\tau$ of $\nabla \tilde{J}_H(\theta)$. Thus $\widehat{\nabla}_m \tilde{J}(\theta) = \frac{1}{m}\sum_{i=1}^m \tilde{g}(\tau_i \mid \theta)$, and $\widehat{\nabla}_m \tilde{J}(\theta)$ and $\tilde{g}(\tau \mid \theta)$ are unbiased estimators of $\tilde{J}_H(\theta)$.

Similar to (C.26), from (C.87) we have

$$\begin{aligned}
\mathbb{E}\left[\left\|\widehat{\nabla}_m \tilde{J}(\theta)\right\|^2\right] &= \mathbb{E}\left[\left\|\widehat{\nabla}_m \tilde{J}(\theta) + \nabla \tilde{J}_H(\theta) - \nabla \tilde{J}_H(\theta)\right\|^2\right] \\
&= \left\|\nabla \tilde{J}_H(\theta)\right\|^2 + \mathbb{E}\left[\left\|\widehat{\nabla}_m \tilde{J}(\theta) - \nabla \tilde{J}_H(\theta)\right\|^2\right] \\
&= \left\|\nabla \tilde{J}_H(\theta)\right\|^2 + \mathbb{E}\left[\left\|\frac{1}{m}\sum_{i=1}^m (\tilde{g}(\tau_i \mid \theta) - \nabla \tilde{J}_H(\theta))\right\|^2\right] \\
&= \left\|\nabla \tilde{J}_H(\theta)\right\|^2 + \frac{1}{m}\mathbb{E}\left[\left\|\tilde{g}(\tau_1 \mid \theta) - \nabla \tilde{J}_H(\theta)\right\|^2\right] \\
&= \left(1-\frac{1}{m}\right)\left\|\nabla \tilde{J}(\theta)\right\|^2 + \frac{1}{m}\mathbb{E}\left[\|\tilde{g}(\tau_1 \mid \theta)\|^2\right]. \qquad \text{(C.89)}
\end{aligned}$$

It remains to show $\mathbb{E}_\tau\left[\|\tilde{g}(\tau \mid \theta)\|^2\right]$ is bounded. From (C.87) we have

$$\begin{aligned}
\mathbb{E}\left[\|\tilde{g}(\tau \mid \theta)\|^2\right] &= \mathbb{E}_\tau\left[\left\|g(\tau \mid \theta) - \lambda\sum_{t=0}^{H-1}\gamma^t \log \pi_{s_t,a_t}(\theta)\left(\sum_{k=0}^{t}\nabla_\theta \log \pi_{s_k,a_k}(\theta)\right)\right.\right. \\
&\qquad \left.\left. - \lambda\sum_{t=0}^{H-1}\gamma^t \nabla_\theta \log \pi_{s_t,a_t}(\theta)\right\|^2\right] \\
&\leq 2\mathbb{E}\left[\|g(\tau \mid \theta)\|^2\right] + 2\lambda^2\mathbb{E}\left[\left\|\sum_{t=0}^{H-1}\gamma^t \log \pi_{s_t,a_t}(\theta)\left(\sum_{k=0}^{t}\nabla_\theta \log \pi_{s_k,a_k}(\theta)\right)\right\|^2\right]
\end{aligned}$$

$$
\begin{aligned}
&+ 2\lambda^2 \mathbb{E}\left[\left\|\sum_{t=0}^{H-1}\gamma^t \nabla_\theta \log \pi_{s_t,a_t}(\theta)\right\|^2\right] \\
&\leq \frac{2\left(1-\frac{1}{|\mathcal{A}|}\right)\mathcal{R}_{\max}^2}{(1-\gamma)^3} + 2\lambda^2 \underbrace{\mathbb{E}\left[\left\|\sum_{t=0}^{H-1}\gamma^t \log \pi_{s_t,a_t}(\theta)\left(\sum_{k=0}^{t}\nabla_\theta \log \pi_{s_k,a_k}(\theta)\right)\right\|^2\right]}_{①} \\
&+ 2\lambda^2 \underbrace{\mathbb{E}\left[\left\|\sum_{t=0}^{H-1}\gamma^t \nabla_\theta \log \pi_{s_t,a_t}(\theta)\right\|^2\right]}_{②}, \qquad \text{(C.90)}
\end{aligned}
$$

where the last inequality is obtained by Lemma 4.9 with GPOMDP estimator and the constant $G^2 = 1 - \frac{1}{|\mathcal{A}|}$ provided from Lemma 4.15.

Now we will bound ① and ② separately.

From Lemma C.6, we know that

$$
\begin{aligned}
② \quad &= \quad \sum_{t=0}^{H-1}\gamma^{2t}\mathbb{E}\left[\|\nabla_\theta \log \pi_{s_t,a_t}(\theta)\|^2\right] \\
&\overset{\text{Lemma 4.15}}{\leq} \quad \left(1-\frac{1}{|\mathcal{A}|}\right)\sum_{t=0}^{H-1}\gamma^{2t} \\
&\leq \quad \frac{1}{1-\gamma^2}\left(1-\frac{1}{|\mathcal{A}|}\right). \qquad \text{(C.91)}
\end{aligned}
$$

As for ①, we have

$$
\begin{aligned}
① \quad &\leq \quad H\sum_{t=0}^{H-1}\gamma^{2t}\mathbb{E}\left[(\log \pi_{s_t,a_t}(\theta))^2\left\|\sum_{k=0}^{t}\nabla_\theta \log \pi_{s_k,a_k}(\theta)\right\|^2\right] \\
&\leq \quad H\sum_{t=0}^{H-1}\gamma^{2t}\mathbb{E}\left[(\log \pi_{s_t,a_t}(\theta))^2\left\|\sum_{k=0}^{t}\nabla_\theta \log \pi_{s_k,a_k}(\theta)\right\|^2\right] \\
&\leq \quad H\sum_{t=0}^{H-1}\gamma^{2t}\mathbb{E}\left[(\log \pi_{s_t,a_t}(\theta))^2(t+1)\sum_{k=0}^{t}\|\nabla_\theta \log \pi_{s_k,a_k}(\theta)\|^2\right] \\
&\overset{\text{(C.53)}}{\leq} \quad 2H\sum_{t=0}^{H-1}\gamma^{2t}(t+1)^2\mathbb{E}\left[(\log \pi_{s_t,a_t}(\theta))^2\right] \\
&\leq \quad 2H|\mathcal{A}|\sum_{t=0}^{H-1}\gamma^{2t}(t+1)^2 \qquad \text{(C.92)} \\
&\leq \quad \frac{4H|\mathcal{A}|}{(1-\gamma)^3}, \qquad \text{(C.93)}
\end{aligned}
$$

where (C.92) is obtained by using

$$\mathbb{E}\left[\left(\log \pi_{s_t,a_t}(\theta)\right)^2\right] \;=\; \mathbb{E}_{s_t}\left[\sum_{a\in\mathcal{A}} \pi_{s_t,a}(\theta)\left(\log \pi_{s_t,a}(\theta)\right)^2\right] \;\leq\; |\mathcal{A}|,$$

and the last line is obtained by $\gamma^{2t} \leq \gamma^t$ and Lemma C.2.

Combining (C.89), (C.90), (C.91) and (C.93) yields the claim of the lemma. □

By adopting Lemma 14 in Mei et al. (2020), we show that $\tilde{J}(\cdot)$ is smooth as following.

**Lemma C.19.** *$\tilde{J}(\cdot)$ is $\left(\frac{\mathcal{R}_{\max}}{(1-\gamma)^2}\left(2-\frac{1}{|\mathcal{A}|}\right)+\frac{\lambda(4+8\log|\mathcal{A}|)}{(1-\gamma)^3}\right)$-smooth.*

*Proof.* From (C.84), we have

$$\tilde{J}(\theta) \;=\; J(\theta) - \lambda\mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma^t \log \pi_{s_t,a_t}(\theta)\right].$$

From Lemma C.10, we know that $J(\cdot)$ is $\left(\frac{\mathcal{R}_{\max}}{(1-\gamma)^2}\left(2-\frac{1}{|\mathcal{A}|}\right)\right)$-smooth.

From Lemma 14 in Mei et al. (2020), we know that $\mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\gamma^t \log \pi_{s_t,a_t}(\theta)\right]$ is $\left(\frac{\lambda(4+8\log|\mathcal{A}|)}{(1-\gamma)^3}\right)$-smooth.

Combining the two smoothness constants yields the claim of the lemma. □

From Lemma C.18 and Lemma C.19 we can also establish a similar FOSP convergence as for Corollary 4.14.

**Corollary C.20.** *Consider the vanilla PG updates (C.87) for the softmax with entropy regularization (C.84) . For a given $\epsilon > 0$, by choosing the mini-batch size $m$ such that $1 \leq m \leq \frac{2\nu}{\epsilon^2}$, the step size $\eta = \frac{\epsilon^2 m}{2L\nu}$, the horizon $H = \mathcal{O}\left((1-\gamma)^{-1}\log(1/\epsilon)\right)$ and the number of iterations $T$ such that*

$$Tm \geq \frac{8\delta_0 L\nu}{\epsilon^4} = \mathcal{O}((1-\gamma)^{-6}\epsilon^{-4}) \tag{C.94}$$

*with*

$$L = \left(\frac{\mathcal{R}_{\max}}{(1-\gamma)^2}\left(2-\frac{1}{|\mathcal{A}|}\right)+\frac{\lambda(4+8\log|\mathcal{A}|)}{(1-\gamma)^3}\right)$$

*and*

$$\nu = \frac{2\left(1-\frac{1}{|\mathcal{A}|}\right)\mathcal{R}_{\max}^2}{(1-\gamma)^3} + \frac{2\lambda^2}{(1-\gamma^2)}\left(1-\frac{1}{|\mathcal{A}|}\right) + \frac{8H|\mathcal{A}|\lambda^2}{(1-\gamma)^3},$$

*then* $\mathbb{E}\left[\left\|\nabla \tilde{J}(\theta_U)\right\|^2\right] = \mathcal{O}(\epsilon^2)$.

**Remark.** The sample complexity $Tm \times H$ is $\mathcal{O}((1-\gamma)^{-8}\epsilon^{-4})$ instead of $\mathcal{O}((1-\gamma)^{-6}\epsilon^{-4})$ as in Corollary 4.14 due to the $(1-\gamma)^{-3}$ dependency on the smoothness constant $L$ and the $(1-\gamma)^{-4}$ dependency on the bounded variance constant $\nu$.

*Proof.* From Lemma C.19, we know that

$$L = \left(\frac{\mathcal{R}_{\max}}{(1-\gamma)^2}\left(2 - \frac{1}{|\mathcal{A}|}\right) + \frac{\lambda(4 + 8\log|\mathcal{A}|)}{(1-\gamma)^3}\right).$$

From Lemma C.18, we know that

$$\nu = \frac{2\left(1 - \frac{1}{|\mathcal{A}|}\right)\mathcal{R}_{\max}^2}{(1-\gamma)^3} + \frac{2\lambda^2}{(1-\gamma^2)}\left(1 - \frac{1}{|\mathcal{A}|}\right) + \frac{8H|\mathcal{A}|\lambda^2}{(1-\gamma)^3}.$$

Plugging in $L$ and $\nu$ in Corollary 4.14 yields the corollary's claim. □

## C.8 Global optimum convergence under the gradient domination assumption

As Fazel et al. (2018) and Mei et al. (2020) did for the exact policy gradient update, relying on the following gradient domination assumption, we establish a global optimum convergence guarantee and the sample complexity analysis for the stochastic vanilla PG.

**Assumption C.21** (Gradient domination)**.** *We say that a differentiable function $J$ satisfies the gradient domination condition if for all $\theta \in \mathbb{R}^d$, there exists $\mu > 0$ such that*

$$\frac{1}{2}\|\nabla J_H(\theta)\|^2 \;\geq\; \mu\left(J^* - J(\theta)\right). \tag{PL}$$

The gradient domination condition is also known as the Polyak-Lojasiewicz (PL) condition. The PL condition was originally discovered independently in the seminal works of B. Polyak and S. Łojasiewicz (Polyak, 1963; Łojasiewicz, 1963; Łojasiewicz, 1959). Equipped with this additional assumption, we can adapt Theorem 3 in Khaled and Richtárik (2023) and obtain the following global optimum convergence guarantee.

**Theorem C.22.** *Suppose that Assumptions 4.1, 4.2, 4.3 and C.21 hold. Suppose that PG defined in (4.10) (Alg. 4) is run for $T > 0$ iterations with step size $(\eta_t)_t$ chosen as*

$$\eta_t = \begin{cases} \frac{1}{b} & \text{if } T \leq \frac{b}{\mu} \text{ or } t \leq t_0 \\ \frac{2}{2b+\mu(t-t_0)} & \text{if } T \geq \frac{b}{\mu} \text{ and } t > t_0 \end{cases} \tag{C.95}$$

*with $t_0 = \left\lceil \frac{T}{2} \right\rceil$ and $b = \max\{2AL/\mu, 2BL, \mu\}$. Then*

$$J^* - \mathbb{E}\left[J(\theta_T)\right] \leq 16 \exp\left(-\frac{\mu(T-1)}{2\max\{\frac{2AL}{\mu}, 2BL, \mu\}}\right)(J^* - J(\theta_0)) + \frac{12LC}{\mu^2 T} + \frac{26D\gamma^H}{\mu}. \tag{C.96}$$

**Remark.** Notice that for the exact full gradient update, we have Assumption 4.2 and 4.3 hold with $A = C = D = 0$ and $B = 1$. Thus under the smoothness assumption and the (PL) condition , we establish a linear convergence rate for the number of iterations to the global optimal. We recover the linear convergence rate for the softmax with entropy regularization in Theorem 6 in Mei et al. (2020) where the smoothness assumption holds and the (PL) condition holds under the path of the iterations in the exact case.

As for the stochastic vanilla PG, the dominant term in (C.96) is $\frac{12LC}{\mu^2 T}$. This implies that the sample complexity is $T \times H = \widetilde{\mathcal{O}}(\epsilon^{-1})$ with $T = \mathcal{O}(\epsilon^{-1})$ and $H = \log \epsilon^{-1}$.

*Proof.* Using the $L$-smoothness of $J$ from Assumption 4.1,

$$\begin{aligned} J^* - J(\theta_{t+1}) &\leq J^* - J(\theta_t) - \langle \nabla J(\theta_t), \theta_{t+1} - \theta_t \rangle + \frac{L}{2}\left\|\theta_{t+1} - \theta_t\right\|^2 \\ &= J^* - J(\theta_t) - \eta_t \left\langle \nabla J(\theta_t), \widehat{\nabla}_m J(\theta_t) \right\rangle + \frac{L\eta_t^2}{2}\left\|\widehat{\nabla}_m J(\theta_t)\right\|^2. \end{aligned}$$

Taking expectation conditioned on $\theta_t$ and using Assumption 4.3 and C.21,

$$\begin{aligned} \mathbb{E}_t\left[J^* - J(\theta_{t+1})\right] &\leq J^* - J(\theta_t) - \eta_t \left\langle \nabla J(\theta_t), \nabla J_H(\theta_t) \right\rangle + \frac{L\eta_t^2}{2}\mathbb{E}_t\left[\left\|\widehat{\nabla}_m J(\theta_t)\right\|^2\right] \\ &\overset{\text{(ABC)}}{\leq} J^* - J(\theta_t) - \eta_t \left\langle \nabla J_H(\theta_t) + (\nabla J(\theta_t) - \nabla J_H(\theta_t)), \nabla J_H(\theta_t) \right\rangle + \\ &\quad + \frac{L\eta_t^2}{2}\left(2A(J^* - J(\theta_t)) + B\left\|\nabla J_H(\theta_t)\right\|^2 + C\right) \\ &= (1 + L\eta_t^2 A)(J^* - J(\theta_t)) - \eta_t\left(1 - \frac{LB\eta_t}{2}\right)\left\|\nabla J_H(\theta_t)\right\|^2 + \frac{L\eta_t^2 C}{2} \end{aligned}$$

$$
\begin{aligned}
& -\eta_t \langle \nabla J(\theta_t) - \nabla J_H(\theta_t), \nabla J_H(\theta_t) \rangle \\
\overset{\text{(PL)}}{\leq} & \left(1 - 2\eta_t\mu\left(1 - \frac{LB\eta_t}{2}\right) + L\eta_t^2 A\right)(J^* - J(\theta_t)) + \frac{L\eta_t^2 C}{2} \\
& -\eta_t \langle \nabla J(\theta_t) - \nabla J_H(\theta_t), \nabla J_H(\theta_t) \rangle \\
\leq & \left(1 - \frac{3\eta_t\mu}{2} + L\eta_t^2 A\right)(J^* - J(\theta_t)) + \frac{L\eta_t^2 C}{2} \qquad \text{(C.97)} \\
\overset{\text{(4.12)}}{\leq} & \left(1 - \frac{3\eta_t\mu}{2} + L\eta_t^2 A\right)(J^* - J(\theta_t)) + \frac{L\eta_t^2 C}{2} + \eta_t D\gamma^H \\
\leq & (1 - \eta_t\mu)(J^* - J(\theta_t)) + \frac{L\eta_t^2 C}{2} + \eta_t D\gamma^H, \qquad \text{(C.98)}
\end{aligned}
$$

where (C.97) is obtained by the inequality $1 - \frac{LB\eta_t}{2} \geq \frac{3}{4}$, and (C.98) is obtained by the inequality $L\eta_t A \leq \frac{\mu}{2}$, due to the choice of step size $\eta_t \leq \frac{1}{b}$ for all $t \geq 0$ with $b \geq 2BL, 2AL/\mu$, respectively. Here, $1 - \eta_t\mu \geq 0$ as $\eta_t \leq \frac{1}{b}$ and $b \geq \mu$.

Taking total expectation and letting $r_t \overset{\text{def}}{=} \mathbb{E}\left[J^* - J(\theta_t)\right]$ on (C.98), we have

$$
r_{t+1} \leq (1 - \eta_t\mu) r_t + \frac{L\eta_t^2 C}{2} + \eta_t D\gamma^H. \qquad \text{(C.99)}
$$

If $T \leq \frac{b}{\mu}$, we have $\eta_t = \frac{1}{b}$. Recursing the above inequality, we get

$$
\begin{aligned}
r_T \quad \leq & \left(1 - \frac{\mu}{b}\right) r_{T-1} + \frac{LC}{2b^2} + \frac{D\gamma^H}{b} \\
\overset{\text{(C.99)}}{\leq} & \left(1 - \frac{\mu}{b}\right)^T r_0 + \left(\frac{LC}{2b^2} + \frac{D\gamma^H}{b}\right) \sum_{i=0}^{T-1} \left(1 - \frac{\mu}{b}\right)^i \\
\leq & \exp\left(-\frac{\mu T}{b}\right) r_0 + \frac{LC}{2\mu b} + \frac{D\gamma^H}{\mu} \qquad \text{(C.100)} \\
\overset{T \leq \frac{b}{\mu}}{\leq} & \exp\left(-\frac{\mu T}{b}\right) r_0 + \frac{LC}{2\mu^2 T} + \frac{D\gamma^H}{\mu}. \qquad \text{(C.101)}
\end{aligned}
$$

If $T \geq \frac{b}{\mu}$, as $\eta_t = \frac{1}{b}$ when $t \leq t_0$, from (C.100), we have

$$
\begin{aligned}
r_{t_0} & \leq \exp\left(-\frac{\mu t_0}{b}\right) r_0 + \frac{LC}{2\mu b} + \frac{D\gamma^H}{\mu} \\
& \leq \exp\left(-\frac{\mu(T-1)}{2b}\right) r_0 + \frac{LC}{2\mu b} + \frac{D\gamma^H}{\mu}, \qquad \text{(C.102)}
\end{aligned}
$$

where the last line is obtained by $t_0 = \left\lceil \frac{T}{2} \right\rceil \geq \frac{T-1}{2}$.

For $t > t_0$,

$$\eta_t = \frac{2}{\mu\left(\frac{2b}{\mu} + t - t_0\right)}.$$

From (C.99), we have

$$\begin{aligned} r_t &\leq (1 - \eta_t\mu)r_{t-1} + \frac{L\eta_t^2 C}{2} + \eta_t D\gamma^H \\ &= \frac{\frac{2b}{\mu} + t - t_0 - 2}{\frac{2b}{\mu} + t - t_0} r_{t-1} + \frac{2LC}{\mu^2\left(\frac{2b}{\mu} + t - t_0\right)^2} + \frac{2D\gamma^H}{\mu\left(\frac{2b}{\mu} + t - t_0\right)}. \end{aligned}$$

Multiplying both sides by $\left(\frac{2b}{\mu} + t - t_0\right)^2$, we have

$$\begin{aligned} \left(\frac{2b}{\mu} + t - t_0\right)^2 r_t &\leq \left(\frac{2b}{\mu} + t - t_0\right)\left(\frac{2b}{\mu} + t - t_0 - 2\right) r_{t-1} + \frac{2LC}{\mu^2} + \frac{2D\gamma^H}{\mu}\left(\frac{2b}{\mu} + t - t_0\right) \\ &\leq \left(\frac{2b}{\mu} + t - t_0 - 1\right)^2 r_{t-1} + \frac{2LC}{\mu^2} + \frac{2D\gamma^H}{\mu}\left(\frac{2b}{\mu} + t - t_0\right). \end{aligned}$$

Let $w_t \overset{\text{def}}{=} \left(\frac{2b}{\mu} + t - t_0\right)^2$. Then,

$$w_t r_t \leq w_{t-1} r_{t-1} + \frac{2LC}{\mu^2} + \frac{2D\gamma^H}{\mu}\left(\frac{2b}{\mu} + t - t_0\right).$$

Summing up for $t = t_0 + 1, \cdots, T$ and telescoping, we get,

$$\begin{aligned} w_T r_T &\leq w_{t_0} r_{t_0} + \frac{2LC(T - t_0)}{\mu^2} + \frac{2D\gamma^H}{\mu} \sum_{t=t_0+1}^{T} \left(\frac{2b}{\mu} + t - t_0\right) \\ &= \frac{4b^2}{\mu^2} r_{t_0} + \frac{2LC(T - t_0)}{\mu^2} + \frac{4bD(T - t_0)\gamma^H}{\mu^2} + \frac{D\gamma^H}{\mu}(T - t_0)(T - t_0 + 1). \end{aligned}$$

Dividing both sides by $w_T$ and using that since

$$w_T \;=\; \left(\frac{2b}{\mu} + T - t_0\right)^2 \;\geq\; (T - t_0)^2,$$

we have

$$\begin{aligned} r_T &\leq \frac{4b^2}{\mu^2 w_T} r_{t_0} + \frac{2LC(T - t_0)}{\mu^2 w_T} + \frac{4bD(T - t_0)\gamma^H}{\mu^2 w_T} + \frac{D\gamma^H}{\mu w_T}(T - t_0)(T - t_0 + 1) \\ &\leq \frac{4b^2}{\mu^2(T - t_0)^2} r_{t_0} + \frac{2LC}{\mu^2(T - t_0)} + \frac{4bD\gamma^H}{\mu^2(T - t_0)} + \frac{2D\gamma^H}{\mu}. \end{aligned}$$

By the definition of $t_0$, we have $T - t_0 \geq \frac{T}{2}$. Plugging this estimate, we have

$$\begin{aligned}
r_T &\leq \frac{16b^2}{\mu^2 T^2} r_{t_0} + \frac{4LC + 8bD\gamma^H}{\mu^2 T} + \frac{2D\gamma^H}{\mu} \\
&\overset{T \geq \frac{b}{\mu}}{\leq} \frac{16b^2}{\mu^2 T^2} r_{t_0} + \frac{4LC}{\mu^2 T} + \frac{10D\gamma^H}{\mu} \\
&\overset{(C.102)}{\leq} \frac{16b^2}{\mu^2 T^2} \left( \exp\left( -\frac{\mu(T-1)}{2b} \right) r_0 + \frac{LC}{2\mu b} + \frac{D\gamma^H}{\mu} \right) + \frac{4LC}{\mu^2 T} + \frac{10D\gamma^H}{\mu} \\
&\overset{T \geq \frac{b}{\mu}}{\leq} 16 \exp\left( -\frac{\mu(T-1)}{2b} \right) r_0 + \frac{8LC}{\mu^2 T} + \frac{16D\gamma^H}{\mu} + \frac{4LC}{\mu^2 T} + \frac{10D\gamma^H}{\mu} \\
&= 16 \exp\left( -\frac{\mu(T-1)}{2b} \right) r_0 + \frac{12LC}{\mu^2 T} + \frac{26D\gamma^H}{\mu}. \qquad \text{(C.103)}
\end{aligned}$$

It remains to take the maximum of the two bounds (C.101) and (C.103) with

$$b = \max\{2AL/\mu, 2BL, \mu\}.$$

□

# Appendix D

# Complements on Chapter 5

Contents



Here we provide the related work discussion, the missing proofs from Chapter 5 and some additional noteworthy observations made in Chapter 5.

## D.1 Related work

### D.1.1 Technical Contribution and Novelty Compared to Xiao (2022)

Our technical novelty compared to Xiao (2022) is summarized as follows.

- Our linear convergence results (i.e., Theorem 5.5, 5.9 and 5.15) are not direct applications of Theorem 10 in Xiao (2022). Indeed, Xiao (2022) establishes the connection between NPG and a specific form of policy mirror descent (PMD) with the use of the weighted Bregman divergence for the tabular setting, while we show that this connection can also be established for the function approximation setting via the compatible function approximation framework (5.11). We also modify the PMD framework of Xiao (2022)

with the linear approximation of the advantage function in (5.18), inspired from the compatible function approximation framework. Thus, the approaches of deriving the PMD form update are different. Without this work of using the compatible function approximation framework to bridge NPG and PMD, it was not clear at all that the analysis of Xiao (2022) could be extended in the log-linear policy setting. So our work is the first step of showing that the proof techniques used in Xiao (2022) can be extended in function approximation regime. In fact, the extension is highly nontrivial and requires significant innovation (see details below). As for future work, one can extend our work to other function approximation setting through a similar compatible function approximation framework. See Section 5.6 for more details about the future work.

- Besides, our linear convergence results only consider the inexact NPG update. Compared to Theorem 14 in Xiao (2022), which is their corresponding result on the inexact PMD method, we improve their analysis by making much weaker assumptions on the accuracy of the estimation $Q(\pi)$. Xiao (2022) requires an $L_\infty$ supremum norm bound on the estimation error of $Q$, i.e., $\|\widehat{Q}(\pi) - Q(\pi)\|_\infty \leq \epsilon_{\text{stat}}$, whereas our convergence guarantee depends on the expected $L_2$ error of the estimate, i.e., Assumption 5.1 and 5.12. For instance, Assumption 5.1 from equation (D.26) can be written as $\mathbb{E}\left[(\phi_{s,a}^\top w^{(k)} - \phi_{s,a}^\top w_\star^{(k)})^2\right] \leq \epsilon_{\text{stat}}$, which can be interpreted as $\mathbb{E}\left[(\widehat{Q}(\pi) - Q(\pi))^2\right] \leq \epsilon_{\text{stat}}$ under the linear approximation setting. The techniques for handling $L_\infty$ and $L_2$ errors are very different. Not only our assumption is weaker, it also benefits from the sample complexity analysis that we explain next.
- Consequently, when considering the sample complexity results we derived for sample-based (Q)-NPG in Corollary 5.11 and 5.17, the difference between our work and Theorem 16 in Xiao (2022), which corresponds to their sample complexity results, is even more significant. Corollary 5.11 with Algorithm `Q-NPG-SGD` (Algorithm 16) satisfies Assumption 5.1 with a number of samples that depends only on the feature dimension $m$ of $\phi$ and does not depend on the cardinality of state space $|\mathcal{S}|$ or action space $|\mathcal{A}|$. In contrast, the assumption $\|\widehat{Q}(\pi) - Q(\pi)\|_\infty \leq \epsilon_{\text{stat}}$ with the $L_\infty$ norm in Xiao (2022, Theorem 16) causes the sample complexity to depend on $|\mathcal{S}||\mathcal{A}|$.

  Furthermore, Xiao (2022) uses a Monte-Carlo approach with multiple independent rollouts per iteration, while our sample-based (Q)-NPG uses one single rollout (Algorithm 13 and 14) combined with regression solvers; Xiao (2022) derives a high probability sample complexity result, while we derive the convergence of the optimality gap $\mathbb{E}\left[V_\rho(\pi^{(K)})\right] - V_\rho(\pi^*)$ which can guarantee that the variance of $V_\rho(\pi^{(K)})$ converges to zero. Thus, our sample-based algorithms had not been considered in Xiao (2022) and our proofs of Corollary 5.11 and 5.17 require a different approach.

  In particular, our sample complexity analysis regarding to the policy evaluation is novel. Although our sample-based algorithms had been considered previously in Agarwal

et al. (2021) and Liu et al. (2020), none of their analysis on the sample complexity was correct. Indeed, Agarwal et al. (2021) required the boundedness of the stochastic gradient estimator, which might not hold as we extensively discussed in Appendix D.4.5. We fixed this by showing that $\mathbb{E}\left[\widehat{Q}_{s,a}(\theta)^2\right]$ is bounded. See Appendix D.4.5 for all the subtleties, including a proof sketch of Corollary 5.11. Liu et al. (2020) also incorrectly used an inequality where the random variables are correlated. See the detailed explanation (Footnote 2) in Appendix D.5.4. We fixed this error with a careful conditional expectation argument. Please refer to Appendix D.5.4 for all the details, including a proof sketch of Corollary 5.17. These dimensions are where an important part of the technical work was done. Therefore, outside of the tabular setting, and considering NPG methods that make use of a regression solver, our complexity analysis is currently the only analysis that is entirely correct that we are aware of.

- Finally we not only extend the work of Xiao (2022) to NPG for log-linear policy, but also consider the Q-NPG method and establish its linear convergence analysis. This is a method that is unique to log-linear policy and again had not been considered in Xiao (2022).

### D.1.2 Finite-time analysis of the natural policy gradient

**NPG for the softmax tabular policies.** For the softmax tabular policies, Shani et al. (2020) show that the unregularized NPG has a $\mathcal{O}(1/\sqrt{k})$ convergence rate and the regularized NPG has a faster $\mathcal{O}(1/k)$ convergence rate by using a decaying step size. Agarwal et al. (2021) improve the convergence rate of the unregularized NPG to $\mathcal{O}(1/k)$ with constant step sizes. Further, Khodadadian et al. (2021a) also achieves $\mathcal{O}(1/k)$ convergence rate for the off-policy natural actor-critic (NAC), and a slower sublinear result is established by Khodadadian et al. (2022a) for the two-time-scale NAC.

By using the entropy regularization, Cen et al. (2021a) achieve a linear convergence rate for NPG. A similar linear convergence result has been obtained by rewriting the NPG update under the PMD framework with the Kullback–Leibler (KL) divergence (Lan, 2022) or with a more general convex regularizer (Zhan et al., 2021). Such approach is also applied in the averaged MDP setting to achieve linear convergence for NPG (Li et al., 2022b). However, adding regularization might induce bias for the solution. Thus, Lan (2022) considers exponentially diminishing regularization to guarantee unbiased solution. Furthermore, by considering both the KL divergence and the diminishing entropy regularization, Li et al. (2022c) establish the linear convergence rate not only for the optimality gap but also for the policy. That is, the policy will converge to the fixed high entropy optimal policy. Consequently, Li et al. (2022c) show a local super-linear convergence of both the policy and optimality gap, as discussed in Xiao (2022, Section 4.3).

Recently, Bhandari and Russo (2021), Khodadadian et al. (2021b) and Khodadadian et al. (2022b) and Xiao (2022) show that regularization is unnecessary for obtaining linear convergence, and it suffices to use appropriate step sizes for NPG. In particular, Bhandari and Russo (2021) propose to use an exact line search for the step size (Theorem 1 (a)) or to choose an adaptive step size (Theorem 1 (c)). Similar adaptive step size is proposed by Khodadadian et al. (2021b) and Khodadadian et al. (2022b). Notice that such adaptive step size requires complete knowledge about the environmental model. Instead, a sufficiently large step size might be enough. In this chapter, we extend the results of Xiao (2022) from the tabular setting to the log-linear policies, using *non-adaptive* geometrically increasing step size and obtaining a linear convergence rate for NPG without regularization.

**NPG with function approximation.** In the function approximation regime, there have been many works investigating the convergence rate of the NPG or NAC algorithms from different perspectives. Wang et al. (2020) establish the $\mathcal{O}(1/\sqrt{k})$ convergence rate for two-layer neural NAC with a projection step. The sublinear convergence results are also established by Zanette et al. (2021) and Hu et al. (2022) for the linear MDP (Jin et al., 2020). Agarwal et al. (2021) obtain the same $\mathcal{O}(1/\sqrt{k})$ convergence rate for the smooth policies with projections. This was later improved to $\mathcal{O}(1/k)$ by Liu et al. (2020) by replacing the projection step with a strong regularity condition on the Fisher information matrix, and it was also improved to $\mathcal{O}(1/k)$ by Xu et al. (2020c) with NAC under Markovian sampling. The same $\mathcal{O}(1/k)$ convergence rate is established for log-linear policies by Chen et al. (2022b) when considering the off-policy NAC.

With entropy regularization and a projection step, Cayci et al. (2021) obtain a linear convergence for log-linear policies. Same entropy regularization and a projection step are applied by Cayci et al. (2022a) for the neural NAC to improve the $\mathcal{O}(1/\sqrt{k})$ convergence rate of Wang et al. (2020) to $\mathcal{O}(1/k)$. In contrast, we show that by using a simple geometrically increasing step size, fast linear convergence can be achieved for log-linear policies without any additional regularization nor a projection step. We notice that Chen and Theja Maguluri (2022, Theorem 3.4)[1] also uses increasing step size and achieves linear convergence for log-linear policies without regularization. The main differences between our result and Theorem 3.4 in Chen and Theja Maguluri (2022) are fourfold. First, they rely on the contraction property of the generalized Bellman operator, while we consider the PMD analysis approach. So the proof techniques are completely different. Second, their parameter update results in the off-policy multi-step temporal difference learning, whereas we require to solve a linear regression problem to minimize the function approximation error. Third, their step size still depends on the iterates which is thus an adaptive step size and is proportional to the total number of iterations $K$, while ours is independent to the iterates nor to $K$. Finally, their assumption on

[1]This result appears after conference proceedings and is available on https://arxiv.org/pdf/2208.03247.pdf.

the modeling error requires an $L_\infty$ supremum norm, i.e., $\|Q_s(\theta^{(k)}) - \Phi w_\star^{(k)}\|_\infty \leq \epsilon_{\text{bias}}$ for all states $s$ of the state space, our convergence guarantee depends on the expected error (e.g., Assumption 5.2, 5.7 or 5.13) which is a much weaker assumption. After publication of our results, we are aware of the concurrent work of Alfano and Rebeschini (2022). They only analyze the Q-NPG method and achieve similar linear convergence results as our Theorem 5.5. In particular, their result in Theorem 4.7 has a better concentrability coefficient compared to our Theorem 5.5. However, their Assumption 4.6 assumes that the relative condition number upper bounds a time-varying ratio which depends on the iterates, while our Assumption 5.3 is independent to the iterates, as defined in (5.25). Furthermore, they only consider the case when the initial state distribution is the same as the target state distribution, while our analysis generalizes with any target state distribution, which is extensively discussed on the distribution mismatch coefficients in Appendix D.6.1. See Table D.1 a complete overview of NPG in the function approximation regime.

**Fast linear convergence of other policy gradient methods.** Different to the PMD analysis approach, by leveraging a gradient dominance property (Polyak, 1963; Łojasiewicz, 1963), fast linear convergence results have also been established for the PG methods under different settings, such as the linear quadratic control problems (Fazel et al., 2018) and the exact PG method with softmax tabular policy and entropy regularization (Mei et al., 2020; Yuan et al., 2022a). Such gradient domination property is widely explored by Bhandari and Russo (2019) to identify more general structural MDP settings. Linear convergence of PG can also be obtained through exact line search (Bhandari and Russo, 2021, Theorem 1 (a)) or by exploiting non-uniform smoothness (Mei et al., 2021).

Alternatively, by considering a general strongly-concave utility function of the state-action occupancy measure and by exploiting the hidden convexity of the problem, Zhang et al. (2020a) also achieve the linear convergence of a variational PG method. When the object is relaxed to a general concave utility function, Zhang et al. (2021a) still achieve the linear convergence by leveraging the hidden convexity of the problem and by adding variance reduction to the PG method.

## D.2 Standard reinforcement learning results

In this section, we prove the standard reinforcement learning results used in Chapter 5, including the policy gradient theorem (5.8), the NPG updates written through the compatible function approximation (5.12) and the NPG updates formalized as policy mirror descent ((5.17) and (5.18)). Then, we prove the performance difference lemma (Kakade and Langford,

**Table D.1** – Overview of different convergence results for NPG methods in the function approximation regime. The darker cells contain our new results. The light cells contain previously known results for NPG or Q-NPG with log-linear policies that we have a direct comparison to our new results. White cells contain existing results that do not have the same setting as ours, so that we could not make a direct comparison among them.

| Setting | Rate | Reg. | C.S. | I.S.* | Pros/cons compared to our work |
|---|---|---|---|---|---|
| Linear convergence | | | | | |
| Regularized NPG with log-linear (Cayci et al., 2021) | Linear | ✓ | ✓ | | Better concentrability coefficients $C_\nu$ |
| Off-policy NAC with log-linear (Chen and Theja Maguluri, 2022) | Linear | | | ✓ | Weaker assumptions on the approximation error with $L_2$ norm instead of $L_\infty$ norm; They use adaptive increasing stepsize, while we use non-adaptive increasing stepsize |
| Q-NPG with log-linear (Alfano and Rebeschini, 2022) | Linear | | | ✓ | Their relative condition number depends on $t$, while ours is independent to $t$ |
| Q-NPG/NPG with log-linear (this work) | Linear | | | ✓ | |
| Sublinear convergence | | | | | |
| PMD for linear MDP (Zanette et al., 2021; Hu et al., 2022) | $\mathcal{O}(\frac{1}{\sqrt{k}})$ | | ✓ | | |
| Two-layer neural NAC (Wang et al., 2020) | $\mathcal{O}(\frac{1}{\sqrt{k}})$ | | ✓ | | |
| Two-layer neural NAC (Cayci et al., 2022a) | $\mathcal{O}(\frac{1}{k})$ | ✓ | ✓ | | |
| NPG with smooth policies (Agarwal et al., 2021) | $\mathcal{O}(\frac{1}{\sqrt{k}})$ | | ✓ | | |
| NAC under Markovian sampling with smooth policies (Xu et al., 2020c) | $\mathcal{O}(\frac{1}{k})$ | | ✓ | | |
| NPG with smooth and Fisher-non-degenerate policies (Liu et al., 2020) | $\mathcal{O}(\frac{1}{k})$ | | ✓ | | |
| Q-NPG with log-linear (Agarwal et al., 2021) | $\mathcal{O}(\frac{1}{\sqrt{k}})$ | | ✓ | | They have better error floor than ours |
| Off-policy NAC with log-linear (Chen et al., 2022b) | $\mathcal{O}(\frac{1}{k})$ | | ✓ | | Weaker assumptions on the approximation error with $L_2$ norm instead of $L_\infty$ norm; They use adaptive increasing stepsize, while we use non-adaptive increasing stepsize |
| Q-NPG/NPG with log-linear (this work) | $\mathcal{O}(\frac{1}{k})$ | | ✓ | | |

* **Reg.**: regularization; **C.S.**: constant stepsize; **I.S.**: increasing stepsize.

2002), which is the first key ingredient for our PMD analysis. The three-point descent lemma (Lemma D.14) is the second key ingredient for our PMD analysis.

**Lemma D.1** (Policy gradient theorem, Theorem 1 in Sutton et al. (2000)). *The full gradient of the value function $V_\rho(\theta)$ can be re-written as (5.8). That is,*

$$\nabla_\theta V_\rho(\theta) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^\theta,\, a\sim\pi_s(\theta)}\left[Q_{s,a}(\theta)\,\nabla_\theta \log \pi_{s,a}(\theta)\right].$$

*Proof.* The gradient of the value function $V_\rho(\theta)$ can be written as follows,

$$\begin{aligned}
\nabla_\theta V_\rho(\theta) &= \nabla_\theta \sum_{s_0\in\mathcal{S},\ a_0\in\mathcal{A}} \rho(s_0)\pi_{s_0,a_0}(\theta)Q_{s_0,a_0}(\theta) \\
&= \sum_{s_0,a_0} \rho(s_0)\left(\nabla_\theta \pi_{s_0,a_0}(\theta)\right)Q_{s_0,a_0}(\theta) + \sum_{s_0,a_0} \rho(s_0)\pi_{s_0,a_0}(\theta)\nabla_\theta Q_{s_0,a_0}(\theta) \\
&= \sum_{s_0,a_0} \rho(s_0)\pi_{s_0,a_0}(\theta)\left(\nabla_\theta \log \pi_{s_0,a_0}(\theta)\right)Q_{s_0,a_0}(\theta) \\
&\quad + \sum_{s_0,a_0} \rho(s_0)\pi_{s_0,a_0}(\theta)\nabla_\theta\left(c(s_0,a_0) + \gamma\sum_{s_1}\mathcal{P}(s_1\mid s_0,a_0)V_{s_1}(\theta)\right) \\
&= \sum_{s_0,a_0} \rho(s_0)\pi_{s_0,a_0}(\theta)\left(\nabla_\theta \log \pi_{s_0,a_0}(\theta)\right)Q_{s_0,a_0}(\theta) \\
&\quad + \gamma\sum_{s_0,a_0,s_1} \rho(s_0)\pi_{s_0,a_0}(\theta)\mathcal{P}(s_1\mid s_0,a_0)\nabla_\theta V_{s_1}(\theta) \\
&= \mathbb{E}\left[Q_{s_0,a_0}(\theta)\nabla_\theta \log \pi_{s_0,a_0}(\theta)\right] + \gamma\mathbb{E}\left[\nabla_\theta V_{s_1}(\theta)\right] \\
&= \sum_{t=0}^{\infty}\gamma^t\mathbb{E}\left[Q_{s_t,a_t}(\theta)\nabla_\theta \log \pi_{s_t,a_t}(\theta)\right]
\end{aligned}$$

where the third equality is obtained through the Bellman equation, and the last step follows from recursion. The above expectation is computed over the trajectories $\{(s_t,a_t)\}_{t\geq 0}$. Notice that we can also rewrite the expectation over the state and action space $\mathcal{S}\times\mathcal{A}$. That is,

$$\begin{aligned}
\nabla_\theta V_\rho(\theta) &= \sum_{t=0}^{\infty}\gamma^t\mathbb{E}\left[Q_{s_t,a_t}(\theta)\nabla_\theta \log \pi_{s_t,a_t}(\theta)\right] \\
&= \sum_{t=0}^{\infty}\gamma^t \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \Pr(s_t=s,a_t=a)Q_{s,a}(\theta)\nabla_\theta \log \pi_{s,a}(\theta) \\
&= \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}\ \sum_{t=0}^{\infty}\Pr(s_t=s,a_t=a)Q_{s,a}(\theta)\nabla_\theta \log \pi_{s,a}(\theta) \\
&= \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \frac{1}{1-\gamma}d^\theta(s)\pi_{s,a}(\theta)Q_{s,a}(\theta)\nabla_\theta \log \pi_{s,a}(\theta),
\end{aligned}$$

where the last line is obtained by the definition of the state-action visitation distribution $\bar{d}^{\pi}_{s,a}(\rho)$ in (5.4). This completes the proof of the claim. □

**Lemma D.2** (NPG updates via compatible function approximation, Theorem 1 in Kakade (2001)). *Consider the NPG updates* (5.9)

$$\theta^{(k+1)} \;=\; \theta^{(k)} - \eta_k F_\rho(\theta^{(k)})^{\dagger}\, \nabla_\theta V_\rho(\theta^{(k)}),$$

*and the updates using the compatible function approximation* (5.12)

$$\theta^{(k+1)} \;=\; \theta^{(k)} - \eta_k w_\star^{(k)},$$

*where* $w_\star^{(k)} \in \operatorname{argmin}_{w\in\mathbb{R}^m} L_A(w, \theta^{(k)}, \bar{d}^{(k)})$. *If the parametrized policy is differentiable for all* $\theta \in \mathbb{R}^m$, *then the two updates are equivalent up to a constant scaling* $(1-\gamma)$ *of* $\eta_k$.

*Proof.* Indeed, using the policy gradient (5.8) and the fact that $\sum_{a\in\mathcal{A}} \nabla\pi_{s,a}(\theta) = 0$ for all $s \in \mathcal{S}$, as $\pi(\theta)$ is differentiable on $\theta$ and $\sum_{a\in\mathcal{A}} \pi_{s,a} = 1$, we have the policy gradient theorem (Sutton et al., 2000)

$$\nabla_\theta V_\rho(\theta) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^\theta,\, a\sim\pi_s(\theta)}\left[A_{s,a}(\theta)\,\nabla_\theta \log \pi_{s,a}(\theta)\right]. \tag{D.1}$$

Furthermore, consider the optima $w_\star^{(k)}$. By the first-order optimality condition, we have

$$\begin{aligned}
& \nabla_w L_A(w_\star^{(k)}, \theta^{(k)}, \bar{d}^{(k)}) = 0 \\
\iff\quad & \mathbb{E}_{(s,a)\sim\bar{d}^{(k)}}\left[\left((w_\star^{(k)})^\top \nabla_\theta \log \pi_{s,a}^{(k)} - A_{s,a}(\theta^{(k)})\right)\nabla_\theta \log \pi_{s,a}^{(k)}\right] = 0 \\
\iff\quad & \mathbb{E}_{(s,a)\sim\bar{d}^{(k)}}\left[\nabla_\theta \log \pi_{s,a}^{(k)}\left(\nabla_\theta \log \pi_{s,a}^{(k)}\right)^\top\right] w_\star^{(k)} = \mathbb{E}_{(s,a)\sim\bar{d}^{(k)}}\left[A_{s,a}(\theta^{(k)})\nabla_\theta \log \pi_{s,a}^{(k)}\right] \\
\overset{(5.9)+(D.1)}{\iff}\quad & F_\rho(\theta^{(k)}) w_\star^{(k)} = (1-\gamma)\nabla_\theta V_\rho(\theta^{(k)}).
\end{aligned}$$

Thus, we have

$$w_\star^{(k)} = (1-\gamma)F_\rho(\theta)^\dagger \nabla_\theta V_\rho(\theta^{(k)})$$

which yields the update (5.9) up to a constant scaling $(1-\gamma)$ of $\eta_k$. □

**Lemma D.3** (NPG updates as policy mirror descent). *The closed form solution to* (5.17) *is given by*

$$\pi_s^{(k+1)} \;=\; \pi_s^{(k)} \odot \frac{\exp\left(-\eta_k \Phi_s w^{(k)}\right)}{\sum_{a\in\mathcal{A}} \pi_{s,a}^{(k)} \exp\left(-\eta_k \phi_{s,a}^\top w^{(k)}\right)} \tag{D.2}$$

$$= \pi_s^{(k)} \odot \frac{\exp\left(-\eta_k \bar{\Phi}_s^{(k)} w^{(k)}\right)}{\sum_{a\in\mathcal{A}} \pi_{s,a}^{(k)} \exp\left(-\eta_k \left(\bar{\phi}_{s,a}(\theta^{(k)})\right)^\top w^{(k)}\right)} \tag{D.3}$$

$$= \arg\min_{p\in\Delta(\mathcal{A})} \left\{\eta_k \left\langle \bar{\Phi}_s^{(k)} w^{(k)}, p\right\rangle + D(p, \pi_s^{(k)})\right\}, \quad \forall s \in \mathcal{S}, \tag{D.4}$$

*where $\odot$ is the element-wise product between vectors, and $\bar{\Phi}_s^{(k)} \in \mathbb{R}^{|\mathcal{A}|\times m}$ is defined in (5.18), i.e.*

$$\left(\bar{\Phi}_{s,a}^{(k)}\right)^\top \overset{def}{=} \bar{\phi}_{s,a}(\theta^{(k)}) \overset{(5.13)}{=} \phi_{s,a} - \mathbb{E}_{a'\sim\pi_s^{(k)}}\left[\phi_{s,a'}\right].$$

*Such policy update coincides the approximate NPG updates (5.33) of the log-linear policy, if $\theta^{(k+1)} = \theta^{(k)} - \eta_k w^{(k)}$ with $w^{(k)} \approx \operatorname{argmin}_w L_A(w, \theta^{(k)}, \tilde{d}^{(k)})$; and coincides the approximate Q-NPG updates (5.19) of the log-linear policy, if $\theta^{(k+1)} = \theta^{(k)} - \eta_k w^{(k)}$ with $w^{(k)} \approx \operatorname{argmin}_w L_Q(w, \theta^{(k)}, \tilde{d}^{(k)})$.*

*Proof.* For shorthand, let $g = \Phi_s w^{(k)}$. Thus, (5.17) fits the format of Lemma D.11 in Appendix D.7 where $q = \pi_s^{(k)}$. Consequently, the closed form solution is given by (D.61), that is

$$\pi_s^{(k+1)} = \frac{\pi_s^{(k)} \odot e^{-\eta_k g}}{\sum_{a\in\mathcal{A}} \pi_{s,a}^{(k)} e^{-\eta_k g_a}} = \frac{\pi_s^{(k)} \odot e^{-\eta_k \Phi_s w^{(k)}}}{\sum_{a\in\mathcal{A}} \pi_{s,a}^{(k)} e^{-\eta_k \phi_{s,a}^\top w^{(k)}}}$$
$$= \pi_s^{(k)} \odot \frac{\exp\left(-\eta_k \bar{\Phi}_s(\theta^{(k)}) w^{(k)}\right)}{\sum_{a\in\mathcal{A}} \pi_{s,a}^{(k)} \exp\left(-\eta_k \left(\bar{\phi}_{s,a}(\theta^{(k)})\right)^\top w^{(k)}\right)}, \tag{D.5}$$

where the last equality is obtained as

$$\bar{\phi}_{s,a}(\theta^{(k)}) = \phi_{s,a} - \mathbb{E}_{a'\sim\pi_s^{(k)}}\left[\phi_{s,a'}\right] = \phi_{s,a} - c_s,$$

with $c_s \in \mathbb{R}$ some constant independent to $a$.

Similarly, by applying Lemma D.11 with $g = \bar{\Phi}_s^{(k)} w^{(k)}$, the closed form solution to (D.4) is (D.5).

As for the closed form updates of the policy for NPG (5.33) and Q-NPG (5.19) with the parameter updates $\theta^{(k+1)} = \theta^{(k)} - \eta_k w^{(k)}$, it is straightforward to verify that it coincides (D.2) and (D.3) given the specific structure of the log-linear policy (5.7), which concludes the proof. □

**Lemma D.4** (Performance difference lemma (Kakade and Langford, 2002)). *For any policy $\pi, \pi' \in \Delta(\mathcal{A})^{\mathcal{S}}$ and $\rho \in \Delta(\mathcal{S})$,*

$$V_\rho(\pi) - V_\rho(\pi') = \frac{1}{1-\gamma}\mathbb{E}_{(s,a)\sim\bar{d}^\pi}\left[A_{s,a}(\pi')\right] \tag{D.6}$$

$$= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^\pi}\left[\langle Q_s(\pi'), \pi_s - \pi'_s\rangle\right], \tag{D.7}$$

*where $Q_s(\pi)$ is the shorthand for $[Q_{s,a}(\pi)]_{a\in\mathcal{A}} \in \mathbb{R}^{|\mathcal{A}|}$ for any policy $\pi$.*

*Proof.* From Lemma 2 in Agarwal et al. (2021), we have

$$V_\rho(\pi) - V_\rho(\pi') = \frac{1}{1-\gamma}\mathbb{E}_{(s,a)\sim\bar{d}^\pi}\left[A_{s,a}(\pi')\right] = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^\pi}\left[\langle A_s(\pi'), \pi_s\rangle\right],$$

where $A_s(\pi)$ is the shorthand for $[A_{s,a}(\pi)]_{a\in\mathcal{A}} \in \mathbb{R}^{|\mathcal{A}|}$ for any policy $\pi$. To show (D.7), it suffices to show

$$\langle A_s(\pi'), \pi_s\rangle = \langle Q_s(\pi'), \pi_s - \pi'_s\rangle, \quad \text{for all } s \in \mathcal{S} \text{ and } \pi, \pi' \in \Delta(\mathcal{A})^{\mathcal{S}}.$$

Let $\mathbf{1}_n$ denote a vector in $\mathbb{R}^n$ with coordinates equal to 1 element-wisely. Indeed, we have

$$\begin{aligned}
\langle A_s(\pi'), \pi_s\rangle &\overset{(5.3)}{=} \left\langle Q_s(\pi') - V_s(\pi')\cdot\mathbf{1}_{|\mathcal{A}|}, \pi_s\right\rangle \\
&= \langle Q_s(\pi'), \pi_s\rangle - \left\langle V_s(\pi')\cdot\mathbf{1}_{|\mathcal{A}|}, \pi_s\right\rangle \\
&= \langle Q_s(\pi'), \pi_s\rangle - V_s(\pi') \\
&\overset{(5.1)}{=} \langle Q_s(\pi'), \pi_s - \pi'_s\rangle,
\end{aligned}$$

from which we conclude the proof. □

## D.3 Algorithms

### D.3.1 NPG and Q-NPG algorithms

Algorithm 11 combined with the sampling procedure (Algorithm 14) and the averaged SGD procedure, called `NPG-SGD` (Algorithm 15), provide the sample-based NPG methods.

Similarly, Algorithm 12 combined with the sampling procedure (Algorithm 13) and the averaged SGD procedure, called `Q-NPG-SGD` (Algorithm 16), provide the sample-based Q-NPG methods.

**Algorithm 11:** Natural policy gradient

**Input:** Initial state-action distribution $\nu$, policy $\pi^{(0)}$, discounted factor $\gamma \in [0, 1)$, step size $\eta_0 > 0$ for NPG update, step size $\alpha > 0$ for `NPG-SGD` update, number of iterations $T$ for `NPG-SGD`

**for** $k = 0$ ***to*** $K - 1$ **do**
- Compute $w^{(k)}$ of (5.33) by `NPG-SGD`, i.e., Algorithm 15 with inputs $(T, \nu, \pi^{(k)}, \gamma, \alpha)$
- Update $\theta^{(k+1)} = \theta^{(k)} - \eta_k w^{(k)}$ and $\eta_k$

**Output:** $\pi^{(K)}$

**Algorithm 12:** Q-Natural policy gradient

**Input:** Initial state-action distribution $\nu$, policy $\pi^{(0)}$, discounted factor $\gamma \in [0, 1)$, step size $\eta_0 > 0$ for Q-NPG update, step size $\alpha > 0$ for `Q-NPG-SGD` update, number of iterations $T$ for `Q-NPG-SGD`

**for** $k = 0$ ***to*** $K - 1$ **do**
- Compute $w^{(k)}$ of (5.19) by `Q-NPG-SGD`, i.e., Algorithm 16 with inputs $(T, \nu, \pi^{(k)}, \gamma, \alpha)$
- Update $\theta^{(k+1)} = \theta^{(k)} - \eta_k w^{(k)}$ and $\eta_k$

**Output:** $\pi_{\theta^{(K)}}$

### D.3.2 Sampling procedures

In practice, we cannot compute the true minimizer $w_\star^{(k)}$ of the regression problem in either (5.33) or (5.19), since computing the expectation $L_A$ or $L_Q$ requires averaging over all state-action pairs $(s, a) \sim \tilde{d}^{(k)}$ and averaging over all trajectories $(s_0, a_0, c_0, s_1, \cdots)$ to compute the values of $Q_{s,a}^{(k)}$ and $A_{s,a}^{(k)}$. So instead, we provide a sampler which is able to obtain unbiased estimates of $Q_{s,a}(\theta)$ (or $A_{s,a}(\theta)$) with $(s, a) \sim \tilde{d}^{\theta}(\nu)$ for any $\pi(\theta)$.

To solve (5.19), we sample $(s, a) \sim \tilde{d}^{(k)}$ and $\widehat{Q}_{s,a}^{(k)}$ by a standard rollout, formalized in Algorithm 13. This sampling procedure is commonly used, for example in Agarwal et al. (2021, Algorithm 1).

It is straightforward to verify that $(s_h, a_h)$ and $\widehat{Q}_{s_h,a_h}(\theta)$ obtained in Algorithm 13 are unbiased for any $\pi(\theta)$. The expected length of the trajectory is $\frac{1}{1-\gamma}$. We provide its proof here for completeness.

**Lemma D.5.** *Consider the output $(s_h, a_h)$ and $\widehat{Q}_{s_h,a_h}(\theta)$ of Algorithm 13. It follows that*

$$\begin{aligned}
&\mathbb{E}\,[h+1] = \frac{1}{1-\gamma}, \\
&\Pr(s_h = s, a_h = a) = \tilde{d}_{s,a}^{\theta}(\nu), \\
&\mathbb{E}\left[\widehat{Q}_{s_h,a_h}(\theta) \;\middle|\; s_h, a_h\right] = Q_{s_h,a_h}(\theta).
\end{aligned}$$

**Algorithm 13:** Sampler for: $(s, a) \sim \tilde{d}^{\theta}(\nu)$ and unbiased estimate $\widehat{Q}_{s,a}(\theta)$ of $Q_{s,a}(\theta)$

**Input:** Initial state-action distribution $\nu$, policy $\pi(\theta)$, discounted factor $\gamma \in [0, 1)$

Initialize $(s_0, a_0) \sim \nu$, the time step $h, t = 0$, the variable $X = 1$
**while** $X = 1$ **do**
    **With probability** $\gamma$**:**
        Sample $s_{h+1} \sim \mathcal{P}(\cdot \mid s_h, a_h)$
        Sample $a_{h+1} \sim \pi_{s_{h+1}}(\theta)$
        $h \leftarrow h + 1$
    **Otherwise with probability** $(1 - \gamma)$**:**
        $X = 0$ \\ *Accept* $(s_h, a_h)$
$X = 1$
Set the estimate $\widehat{Q}_{s_h,a_h}(\theta) = c(s_h, a_h)$ \\ *Start to estimate* $\widehat{Q}_{s_h,a_h}(\theta)$
$t = h$
**while** $X = 1$ **do**
    **With probability** $\gamma$**:**
        Sample $s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)$
        Sample $a_{t+1} \sim \pi_{s_{t+1}}(\theta)$
        $\widehat{Q}_{s_h,a_h}(\theta) \leftarrow \widehat{Q}_{s_h,a_h}(\theta) + c(s_{t+1}, a_{t+1})$
        $t \leftarrow t + 1$
    **Otherwise with probability** $(1 - \gamma)$**:**
        $X = 0$ \\ *Accept* $\widehat{Q}_{s_h,a_h}(\theta)$

**Output:** $(s_h, a_h)$ and $\widehat{Q}_{s_h,a_h}(\theta)$

*Proof.* The expected length $(h+1)$ of sampling $(s,a)$ is

$$\mathbb{E}\left[h+1\right] = \sum_{k=0}^{\infty} \Pr(h=k)(k+1) = (1-\gamma)\sum_{k=0}^{\infty}\gamma^k(k+1) = \frac{1}{1-\gamma}.$$

The probability of the state-action pair $(s,a)$ being sampled by Algorithm 13 is

$$\begin{aligned}\Pr(s_h=s,a_h=a) &= \sum_{(s_0,a_0)\in\mathcal{S}\times\mathcal{A}} \nu_{s_0,a_0}\sum_{k=0}^{\infty}\Pr(h=k)\Pr^{\pi(\theta)}(s_h=s,a_h=a\mid h=k,s_0,a_0)\\ &= \sum_{(s_0,a_0)\in\mathcal{S}\times\mathcal{A}} \nu_{s_0,a_0}(1-\gamma)\sum_{k=0}^{\infty}\gamma^k\Pr^{\pi(\theta)}(s_k=s,a_k=a\mid s_0,a_0) \overset{(5.5)}{=} \tilde{d}^{\theta}_{s,a}(\nu).\end{aligned}$$

Now we verify that $\widehat{Q}_{s_h,a_h}(\theta)$ obtained from Algorithm 13 is an unbiased estimate of $Q_{s_h,a_h}(\theta)$. Indeed, from Algorithm 13, we have

$$\widehat{Q}_{s_h,a_h}(\theta) = \sum_{t=0}^{H} c(s_{t+h},a_{t+h}), \tag{D.8}$$

where $(H+1)$ is the length of the horizon executed between lines 13 and 19 in Algorithm 13 for calculating $\widehat{Q}_{s_h,a_h}(\theta)$. To simplify notation, we consider the estimate of $\widehat{Q}_{s,a}$ for any $(s,a)\in\mathcal{S}\times\mathcal{A}$ following the same procedure starting from line 10 in Algorithm 13. Taking expectation, we have

$$\begin{aligned}\mathbb{E}\left[\widehat{Q}_{s,a}(\theta)\ \mid\ s,a\right] &= \mathbb{E}\left[\sum_{t=0}^{H} c(s_t,a_t)\mid s_0=s,a_0=a\right]\\ &= \sum_{k=0}^{\infty}\Pr(H=k)\mathbb{E}\left[\sum_{t=0}^{H} c(s_t,a_t)\mid s_0=s,a_0=a,H=k\right]\\ &= \sum_{k=0}^{\infty}(1-\gamma)\gamma^k\mathbb{E}\left[\sum_{t=0}^{k} c(s_t,a_t)\mid s_0=s,a_0=a\right]\\ &= (1-\gamma)\mathbb{E}\left[\sum_{t=0}^{\infty} c(s_t,a_t)\sum_{k=t}^{\infty}\gamma^k\mid s_0=s,a_0=a\right]\\ &= \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^k c(s_t,a_t)\mid s_0=s,a_0=a\right] \overset{(5.2)}{=} Q_{s,a}(\theta).\end{aligned}$$

The desired result is obtained by setting $s=s_h$ and $a=a_h$. □

Similar to Algorithm 13, to solve (5.33), we sample $(s,a) \sim \tilde{d}^{(k)}$ by the same procedure and estimate $\widehat{A}_{s,a}^{(k)}$ with a slight modification, namely Algorithm 14 (also see Agarwal et al., 2021, Algorithm 3).

Notice that the sampling procedure for estimating $Q_{s,a}(\theta)$ in Algorithm 13 is simpler than that for estimating $A_{s,a}(\theta)$ in Algorithm 14, since Algorithm 14 requires an additional estimation of $V_s(\theta)$ and thus doubles the number of samples to estimate $A_{s,a}(\theta)$. As in Lemma D.5, we verify in the following lemma that the output $(s_h, a_h)$ is sampled from the distribution $\tilde{d}^{\theta}$ and $\widehat{A}_{s_h,a_h}(\theta)$ in Algorithm 14 is an unbiased estimator of $A_{s_h,a_h}(\theta)$ for all policy $\pi(\theta)$.

**Lemma D.6.** *Consider the output $(s_h, a_h)$ and $\widehat{A}_{s_h,a_h}(\theta)$ of Algorithm 14. It follows that*

$$\begin{aligned}
&\mathbb{E}\left[h+1\right] = \frac{1}{1-\gamma}, \\
&\Pr(s_h = s, a_h = a) = \tilde{d}_{s,a}^{\theta}(\nu), \\
&\mathbb{E}\left[\widehat{A}_{s_h,a_h}(\theta) \;\middle|\; s_h, a_h\right] = A_{s_h,a_h}(\theta).
\end{aligned}$$

*Proof.* Since the procedure of sampling $(s_h, a_h)$ in Algorithm 14 is identical to the one in Algorithm 13, from Lemma D.5, the first two results are verified. It remains to show that $\widehat{A}_{s_h,a_h}(\theta)$ is unbiased.

The estimation of $\widehat{A}_{s_h,a_h}(\theta)$ is decomposed into the estimations of $\widehat{Q}_{s_h,a_h}(\theta)$ and $\widehat{V}_{s_h}(\theta)$. The procedure of estimating $\widehat{Q}_{s_h,a_h}(\theta)$ is also identical to the one in Algorithm 13. Thus, from Lemma D.5, we have

$$\mathbb{E}\left[\widehat{Q}_{s_h,a_h}(\theta) \;\middle|\; s_h, a_h\right] = Q_{s_h,a_h}(\theta).$$

By following the similar arguments of Lemma D.5, one can verify that

$$\mathbb{E}\left[\widehat{V}_{s_h}(\theta) \;\middle|\; s_h, a_h\right] = V_{s_h}(\theta).$$

Combine the above two equalities and obtain that

$$\mathbb{E}\left[\widehat{A}_{s_h,a_h}(\theta) \;\middle|\; s_h, a_h\right] = \mathbb{E}\left[\widehat{Q}_{s_h,a_h}(\theta) - \widehat{V}_{s_h}(\theta) \;\middle|\; s_h, a_h\right] = Q_{s_h,a_h}(\theta) - V_{s_h}(\theta) \overset{(5.3)}{=} A_{s_h,a_h}(\theta).$$

□

**Algorithm 14:** Sampler for: $(s,a) \sim \tilde{d}^{\theta}(\nu)$ and unbiased estimate $\widehat{A}_{s,a}(\theta)$ of $A_{s,a}(\theta)$

**Input:** Initial state-action distribution $\nu$, policy $\pi(\theta)$, discounted factor $\gamma \in [0,1)$

Initialize $(s_0, a_0) \sim \nu$, the time step $h, t = 0$, the variable $X = 1$
**while** $X = 1$ **do**
  **With probability** $\gamma$**:**
    Sample $s_{h+1} \sim \mathcal{P}(\cdot \mid s_h, a_h)$
    Sample $a_{h+1} \sim \pi_{s_{h+1}}(\theta)$
    $h \leftarrow h + 1$
  **Otherwise with probability** $(1-\gamma)$**:**
    $X = 0$ \\ *Accept* $(s_h, a_h)$
$X = 1$
Set the estimate $\widehat{Q}_{s_h,a_h}(\theta) = c(s_h, a_h)$ \\ *Start to estimate* $\widehat{Q}_{s_h,a_h}(\theta)$
$t = h$
**while** $X = 1$ **do**
  **With probability** $\gamma$**:**
    Sample $s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)$
    Sample $a_{t+1} \sim \pi_{s_{t+1}}(\theta)$
    $\widehat{Q}_{s_h,a_h}(\theta) \leftarrow \widehat{Q}_{s_h,a_h}(\theta) + c(s_{t+1}, a_{t+1})$
    $t \leftarrow t + 1$
  **Otherwise with probability** $(1-\gamma)$**:**
    $X = 0$ \\ *Accept* $\widehat{Q}_{s_h,a_h}(\theta)$
$X = 1$
Set the estimate $\widehat{V}_{s_h}(\theta) = 0$ \\ *Start to estimate* $\widehat{V}_{s_h}(\theta)$
$t = h$
**while** $X = 1$ **do**
  Sample $a_t \sim \pi_{s_t}(\theta)$
  $\widehat{V}_{s_h}(\theta) \leftarrow \widehat{V}_{s_h}(\theta) + c(s_t, a_t)$
  **With probability** $\gamma$**:**
    Sample $s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)$
    $t \leftarrow t + 1$
  **Otherwise with probability** $(1-\gamma)$**:**
    $X = 0$ \\ *Accept* $\widehat{V}_{s_h}(\theta)$

**Output:** $(s_h, a_h)$ and $\widehat{A}_{s_h,a_h}(\theta) = \widehat{Q}_{s_h,a_h}(\theta) - \widehat{V}_{s_h}(\theta)$

### D.3.3 SGD procedures for solving the regression problems of NPG and Q-NPG

Once we obtain the sampled $(s,a)$ and $\widehat{A}_{s,a}(\theta^{(k)})$ from Algorithm 14, we can apply the averaged SGD algorithm as in Bach and Moulines (2013) to solve the regression problem (5.33) of NPG for every iteration $k$.

Here we suppress the superscript $(k)$. For any parameter $\theta \in \mathbb{R}^m$, recall the compatible function approximation $L_A$ in (5.33)

$$L_A(w,\theta,\tilde{d}^\theta) = \mathbb{E}_{(s,a)\sim\tilde{d}^\theta}\left[\left(w^\top \bar{\phi}_{s,a}(\theta) - A_{s,a}(\theta)\right)^2\right].$$

With the output $(s,a)\sim\tilde{d}^\theta$ and $\widehat{A}_{s,a}(\theta)$ from Algorithm 14 (here we suppress the subscript $h$), we compute the stochastic gradient estimator of the function $L_A$ in (5.33) by

$$\widehat{\nabla}_w L_A(w,\theta,\tilde{d}^\theta) \stackrel{\text{def}}{=} 2\left(w^\top \bar{\phi}_{s,a}(\theta) - \widehat{A}_{s,a}(\theta)\right)\bar{\phi}_{s,a}(\theta). \tag{D.9}$$

Next, we show that (D.9) is an unbiased gradient estimator of the loss function $L_A$.

**Lemma D.7.** *Consider the output $(s,a)$ and $\widehat{A}_{s,a}(\theta)$ of Algorithm 14 and the stochastic gradient (D.9). It follows that*

$$\mathbb{E}\left[\widehat{\nabla}_w L_A(w,\theta,\tilde{d}^\theta)\right] = \nabla_w L_A(w,\theta,\tilde{d}^\theta),$$

*where the expectation is with respect to the randomness in the sequence of the sampled $s_0,a_0,\cdots,s_t,a_t$ from Algorithm 14.*

*Proof.* The total expectation of the stochastic gradient is given by

$$\begin{aligned}\mathbb{E}\left[\widehat{\nabla}_w L_A(w,\theta,\tilde{d}^\theta)\right] &\overset{(D.9)}{=} \mathbb{E}_{s,\,a,\,\widehat{A}_{s,a}(\theta)}\left[2\left(w^\top \bar{\phi}_{s,a}(\theta) - \widehat{A}_{s,a}(\theta)\right)\bar{\phi}_{s,a}(\theta)\right] \\ &= \mathbb{E}_{(s,a)\sim\tilde{d}^\theta,\,\widehat{A}_{s,a}(\theta)}\left[2\left(w^\top \bar{\phi}_{s,a}(\theta) - \widehat{A}_{s,a}(\theta)\right)\bar{\phi}_{s,a}(\theta)\mid s,a\right],\end{aligned} \tag{D.10}$$

where the second line is obtained by $(s,a)\sim\tilde{d}^\theta$ from Lemma D.6.

From Lemma D.6, we have

$$\mathbb{E}_{s_0,a_0,\cdots,s_t,a_t}\left[\widehat{A}_{s,a}(\theta)\mid s_0=s,a_0=a\right] = A_{s,a}(\theta). \tag{D.11}$$

Combining the above two equalities yield

$$\mathbb{E}\left[\widehat{\nabla}_w L_A(w,\theta,\tilde{d}^\theta)\right] \overset{(D.10)}{=} \mathbb{E}_{(s,a)\sim\tilde{d}^\theta}\left[2\left(w^\top \bar{\phi}_{s,a}(\theta) - \mathbb{E}\left[\widehat{A}_{s,a}(\theta)\mid s,a\right]\right)\bar{\phi}_{s,a}(\theta)\right]$$

$$\begin{aligned} &\overset{\text{(D.11)}}{=} \quad \mathbb{E}_{(s,a)\sim\tilde{d}^\theta}\left[2\left(w^\top\bar{\phi}_{s,a}(\theta) - A_{s,a}(\theta)\right)\bar{\phi}_{s,a}(\theta)\right] \\ &= \quad \nabla_w L_A(w,\theta,\tilde{d}^\theta), \end{aligned}$$

as desired. □

Since (D.9) is unbiased shown in Lemma D.7, we can use it for the averaged SGD algorithm to minimize $L_A$, called `NPG-SGD` in Algorithm 15 (also see Agarwal et al., 2021, Algorithm 4).

**Algorithm 15:** NPG-SGD

**Input:** Number of iterations $T$, step size $\alpha > 0$, initialization $w_0 \in \mathbb{R}^m$, initial state-action measure $\nu$, policy $\pi(\theta)$, discounted factor $\gamma \in [0,1)$

**for** $t = 0$ ***to*** $T-1$ **do**

  Call Algorithm 14 with the inputs $(\nu, \pi(\theta), \gamma)$ to sample $(s,a) \sim \tilde{d}^\theta$ and $\widehat{A}_{s,a}(\theta)$

  Update $w_{t+1} = w_t - \alpha\widehat{\nabla}_w L_A(w,\theta,\tilde{d}^\theta)$ by using (D.9)

**Output:** $w_{\text{out}} = \frac{1}{T}\sum_{t=1}^T w_t$

Similar to Algorithm 15, once we obtain the sampled $(s,a)$ and $\widehat{Q}_{s,a}(\theta)$ from Algorithm 13, we can apply the averaged SGD algorithm to solve (5.19) of Q-NPG.

Recall the compatible function approximation $L_Q$ in (5.19)

$$L_Q(w,\theta,\tilde{d}^\theta) = \mathbb{E}_{(s,a)\sim\tilde{d}^\theta}\left[\left(w^\top\phi_{s,a} - Q_{s,a}(\theta)\right)^2\right].$$

With the output $(s,a) \sim \tilde{d}^\theta$ and $\widehat{Q}_{s,a}(\theta)$ from Algorithm 13, we compute the stochastic gradient estimator of the function $L_Q$ in (5.19) by

$$\widehat{\nabla}_w L_Q(w,\theta,\tilde{d}^\theta) \overset{\text{def}}{=} 2\left(w^\top\phi_{s,a} - \widehat{Q}_{s,a}(\theta)\right)\phi_{s,a}, \tag{D.12}$$

and use it for the averaged SGD algorithm to minimize $L_Q$, called `Q-NPG-SGD` in Algorithm 16 (also see Agarwal et al., 2021, Algorithm 2). Compared to (D.9), the cost of computing (D.12) is $|\mathcal{A}|$ times cheaper than that of computing (D.12). Indeed, to compute (D.12), we only need one single action for $\phi_{s,a}$, while to compute (D.9), one needs to go through all the actions to compute $\bar{\phi}_{s,a}(\theta)$. Thus, the computational cost of `Q-NPG-SGD` is $|\mathcal{A}|$ times cheaper than that of `NPG-SGD`.

The estimator $\widehat{\nabla}_w L_Q(w,\theta,\tilde{d}^\theta)$ is also unbiased following the similar argument of the proof of Lemma D.7. We formalize this in the following and omit the proof.

**Algorithm 16:** Q-NPG-SGD

**Input:** Number of iterations $T$, step size $\alpha > 0$, initialization $w_0 \in \mathbb{R}^m$, initial state-action measure $\nu$, policy $\pi(\theta)$, discounted factor $\gamma \in [0, 1)$

**for** $t = 0$ ***to*** $T - 1$ **do**
- Call Algorithm 13 with the inputs $(\nu, \pi(\theta), \gamma)$ to sample $(s, a) \sim \tilde{d}^{\theta}$ and $\widehat{Q}_{s,a}(\theta)$
- Update $w_{t+1} = w_t - \alpha \widehat{\nabla}_w L_Q(w, \theta, \tilde{d}^{\theta})$ by using (D.12)

**Output:** $w_{\text{out}} = \frac{1}{T} \sum_{t=1}^{T} w_t$

**Lemma D.8.** *Consider the output $(s, a)$ and $\widehat{Q}_{s,a}(\theta)$ of Algorithm 13 and the stochastic gradient (D.12). It follows that*

$$\mathbb{E}\left[\widehat{\nabla}_w L_Q(w, \theta, \tilde{d}^{\theta})\right] = \nabla_w L_Q(w, \theta, \tilde{d}^{\theta}),$$

*where the expectation is with respect to the randomness in the sequence of the sampled $s_0, a_0, \cdots, s_t, a_t$ from Algorithm 13.*

## D.4 Proof of Section 5.4

Throughout this section and the next, we use the shorthand $V_\rho^{(k)}$ for $V_\rho(\theta^{(k)})$ and similarly, $Q_{s,a}^{(k)}$ for $Q_{s,a}(\theta^{(k)})$ and $A_{s,a}^{(k)}$ for $A_{s,a}(\theta^{(k)})$. We also use the shorthand $Q_s^{(k)}$ for the vector $\left[Q_{s,a}^{(k)}\right]_{a \in \mathcal{A}} \in \mathbb{R}^{|\mathcal{A}|}$ and $A_s^{(k)}$ for the vector $\left[A_{s,a}^{(k)}\right]_{a \in \mathcal{A}} \in \mathbb{R}^{|\mathcal{A}|}$.

We first provide the one step analysis of the Q-NPG update, which will be helpful for proving Theorem 5.5, 5.6 and 5.9.

### D.4.1 The one step Q-NPG lemma

The following one step analysis of Q-NPG is based on the mirror descent approach of Xiao (2022).

**Lemma D.9** (One step Q-NPG lemma)**.** *Fix a state distribution $\rho$; an initial state-action distribution $\nu$; an arbitrary comparator policy $\pi^*$. Let $w_\star^{(k)} \in \operatorname{argmin}_w L_Q(w, \theta^{(k)}, \tilde{d}^{(k)})$ denote the exact minimizer. Consider the $w^{(k)}$ and $\pi^{(k)}$ given in (5.19) and (5.17) respectively. We have that*

$$\vartheta_\rho(1-\gamma)\left(V_\rho^{(k+1)} - V_\rho^{(k)}\right) + (1-\gamma)\left(V_\rho^{(k)} - V_\rho(\pi^*)\right)$$
$$+ \vartheta_\rho \underbrace{\left(\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} d_s^{(k+1)} \pi_{s,a}^{(k+1)} \phi_{s,a}^\top \left(w^{(k)} - w_\star^{(k)}\right)\right)}_{①} + \underbrace{\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} d_s^{(k+1)} \pi_{s,a}^{(k+1)} \left(\phi_{s,a}^\top w_\star^{(k)} - Q_{s,a}^{(k)}\right)}_{②}$$

$$
\begin{aligned}
&+ \underbrace{\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_s^{(k+1)}\pi_{s,a}^{(k)}\phi_{s,a}^\top\left(w_\star^{(k)} - w^{(k)}\right)}_{③} + \underbrace{\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_s^{(k+1)}\pi_{s,a}^{(k)}\left(Q_{s,a}^{(k)} - \phi_{s,a}^\top w_\star^{(k)}\right)}_{④}\Bigg) \\
&+ \underbrace{\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} d_s^*\pi_{s,a}^{(k)}\phi_{s,a}^\top\left(w^{(k)} - w_\star^{(k)}\right)}_{ⓐ} + \underbrace{\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} d_s^*\pi_{s,a}^{(k)}\left(\phi_{s,a}^\top w_\star^{(k)} - Q_{s,a}^{(k)}\right)}_{ⓑ} \\
&+ \underbrace{\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} d_s^*\pi_{s,a}^*\phi_{s,a}^\top\left(w_\star^{(k)} - w^{(k)}\right)}_{ⓒ} + \underbrace{\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} d_s^*\pi_{s,a}^*\left(Q_{s,a}^{(k)} - \phi_{s,a}^\top w_\star^{(k)}\right)}_{ⓓ} \\
&\leq \frac{1}{\eta_k}D_k^* - \frac{1}{\eta_k}D_{k+1}^*. \qquad \text{(D.13)}
\end{aligned}
$$

*Proof.* As discussed in Section 5.3.1 and from Lemma D.3, we know that the corresponding update from $\pi^{(k)}$ to $\pi^{(k+1)}$ can be described by the PMD method (5.17). In the context of the PMD method (5.17), we apply the three-point descent lemma (Lemma D.14) with $\mathcal{C} = \Delta(\mathcal{A})$, $f$ is the linear function $\eta_k \left\langle \Phi_s w^{(k)}, \cdot \right\rangle$ and $h : \Delta(\mathcal{A}) \to \mathbb{R}$ is the negative entropy with $h(p) = \sum_{a\in\mathcal{A}} p_a \log p_a$. Thus, $h$ is of Legendre type with $\operatorname{rint}\operatorname{dom} h \cap \mathcal{C} = \operatorname{rint}\Delta(\mathcal{A}) \neq \emptyset$ and $D_h(\cdot,\cdot)$ is the KL divergence $D(\cdot,\cdot)$. From Lemma D.14, we obtain that for any $p \in \Delta(\mathcal{A})$, we have

$$
\eta_k \left\langle \Phi_s w^{(k)}, \pi_s^{(k+1)} \right\rangle + D(\pi_s^{(k+1)}, \pi_s^{(k)}) \leq \eta_k \left\langle \Phi_s w^{(k)}, p \right\rangle + D(p, \pi_s^{(k)}) - D(p, \pi_s^{(k+1)}).
$$

Rearranging terms and dividing both sides by $\eta_k$, we get

$$
\left\langle \Phi_s w^{(k)}, \pi_s^{(k+1)} - p \right\rangle + \frac{1}{\eta_k} D(\pi_s^{(k+1)}, \pi_s^{(k)}) \leq \frac{1}{\eta_k} D(p, \pi_s^{(k)}) - \frac{1}{\eta_k} D(p, \pi_s^{(k+1)}). \qquad \text{(D.14)}
$$

Letting $p = \pi_s^{(k)}$ yields

$$
\left\langle \Phi_s w^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)} \right\rangle \leq -\frac{1}{\eta_k} D(\pi_s^{(k+1)}, \pi_s^{(k)}) - \frac{1}{\eta_k} D(\pi_s^{(k)}, \pi_s^{(k+1)}) \leq 0. \qquad \text{(D.15)}
$$

Letting $p = \pi_s^*$ and subtract and add $\pi_s^{(k)}$ within the inner product term in (D.14) yields

$$
\left\langle \Phi_s w^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)} \right\rangle + \left\langle \Phi_s w^{(k)}, \pi_s^{(k)} - \pi_s^* \right\rangle \leq \frac{1}{\eta_k} D(\pi_s^*, \pi_s^{(k)}) - \frac{1}{\eta_k} D(\pi_s^*, \pi_s^{(k+1)}).
$$

Note that we dropped the nonnegative term $\frac{1}{\eta_k} D(\pi_s^{(k+1)}, \pi_s^{(k)})$ on the left hand side to the inequality.

Taking expectation with respect to the distribution $d^*$, we have

$$\mathbb{E}_{s\sim d^*}\left[\left\langle \Phi_s w^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)}\right\rangle\right] + \mathbb{E}_{s\sim d^*}\left[\left\langle \Phi_s w^{(k)}, \pi_s^{(k)} - \pi_s^{*}\right\rangle\right] \leq \frac{1}{\eta_k} D_k^* - \frac{1}{\eta_k} D_{k+1}^*. \tag{D.16}$$

For the first expectation in (D.16), we have

$$\begin{aligned}
& \mathbb{E}_{s\sim d^*}\left[\left\langle \Phi_s w^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)}\right\rangle\right] \\
= & \sum_{s\in\mathcal{S}} d_s^* \left\langle \Phi_s w^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)}\right\rangle \\
= & \sum_{s\in\mathcal{S}} \frac{d_s^*}{d_s^{(k+1)}} d_s^{(k+1)} \left\langle \Phi_s w^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)}\right\rangle \\
\geq & \vartheta_{k+1} \sum_{s\in\mathcal{S}} d_s^{(k+1)} \left\langle \Phi_s w^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)}\right\rangle \\
\geq & \vartheta_{\rho} \sum_{s\in\mathcal{S}} d_s^{(k+1)} \left\langle \Phi_s w^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)}\right\rangle \\
= & \vartheta_{\rho} \sum_{s\in\mathcal{S}} d_s^{(k+1)} \left\langle Q_s^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)}\right\rangle + \vartheta_{\rho} \sum_{s\in\mathcal{S}} d_s^{(k+1)} \left\langle \Phi_s w^{(k)} - Q_s^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)}\right\rangle \\
= & \vartheta_{\rho}(1-\gamma)\left(V_{\rho}^{(k+1)} - V_{\rho}^{(k)}\right) + \vartheta_{\rho} \sum_{s\in\mathcal{S}} d_s^{(k+1)} \left\langle \Phi_s w^{(k)} - Q_s^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)}\right\rangle,
\end{aligned} \tag{D.17}$$

where the last equality is due to the performance difference lemma (D.7) in Lemma D.4 and the two inequalities above are obtained by the negative sign of $\left\langle \Phi_s w^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)}\right\rangle$ shown in (D.15) and by using the following inequality

$$\frac{d_s^*}{d_s^{(k+1)}} \overset{(5.21)}{\leq} \vartheta_{k+1} \overset{(5.21)}{\leq} \vartheta_{\rho}.$$

The second term of (D.17) can be decomposed into four terms. That is,

$$\begin{aligned}
& \sum_{s\in\mathcal{S}} d_s^{(k+1)} \left\langle \Phi_s w^{(k)} - Q_s^{(k)}, \pi_s^{(k+1)} - \pi_s^{(k)}\right\rangle \\
= & \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_s^{(k+1)} \pi_{s,a}^{(k+1)} \left(\phi_{s,a}^{\top} w^{(k)} - Q_{s,a}^{(k)}\right) + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_s^{(k+1)} \pi_{s,a}^{(k)} \left(Q_{s,a}^{(k)} - \phi_{s,a}^{\top} w^{(k)}\right) \\
= & \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_s^{(k+1)} \pi_{s,a}^{(k+1)} \phi_{s,a}^{\top} \left(w^{(k)} - w_{\star}^{(k)}\right) + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_s^{(k+1)} \pi_{s,a}^{(k+1)} \left(\phi_{s,a}^{\top} w_{\star}^{(k)} - Q_{s,a}^{(k)}\right) \\
& + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_s^{(k+1)} \pi_{s,a}^{(k)} \phi_{s,a}^{\top} \left(w_{\star}^{(k)} - w^{(k)}\right) + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_s^{(k+1)} \pi_{s,a}^{(k)} \left(Q_{s,a}^{(k)} - \phi_{s,a}^{\top} w_{\star}^{(k)}\right) \\
= & \text{①} + \text{②} + \text{③} + \text{④},
\end{aligned} \tag{D.18}$$

where ①, ②, ③ and ④ are defined in (D.13).

For the second expectation in (D.16), by applying again the performance difference lemma (D.7), we have

$$\begin{aligned} & \mathbb{E}_{s\sim d^*}\left[\left\langle \Phi_s w^{(k)}, \pi_s^{(k)} - \pi_s^* \right\rangle\right] \\ = \; & \mathbb{E}_{s\sim d^*}\left[\left\langle Q_s^{(k)}, \pi_s^{(k)} - \pi_s^* \right\rangle\right] + \mathbb{E}_{s\sim d^*}\left[\left\langle \Phi_s w^{(k)} - Q_s^{(k)}, \pi_s^{(k)} - \pi_s^* \right\rangle\right] \\ \overset{(D.7)}{=} \; & (1-\gamma)\left(V_\rho^{(k)} - V_\rho(\pi^*)\right) + \mathbb{E}_{s\sim d^*}\left[\left\langle \Phi_s w^{(k)} - Q_s^{(k)}, \pi_s^{(k)} - \pi_s^* \right\rangle\right]. \end{aligned} \tag{D.19}$$

Similarly, we decompose the second term of (D.19) into four terms. That is,

$$\begin{aligned} & \mathbb{E}_{s\sim d^*}\left[\left\langle \Phi_s w^{(k)} - Q_s^{(k)}, \pi_s^{(k)} - \pi_s^* \right\rangle\right] \\ = & \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_s^* \pi_{s,a}^{(k)}\left(\phi_{s,a}^\top w^{(k)} - Q_{s,a}^{(k)}\right) + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_s^* \pi_{s,a}^*\left(Q_{s,a}^{(k)} - \phi_{s,a}^\top w^{(k)}\right) \\ = & \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} d_s^* \pi_{s,a}^{(k)} \phi_{s,a}^\top\left(w^{(k)} - w_\star^{(k)}\right) + \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} d_s^* \pi_{s,a}^{(k)}\left(\phi_{s,a}^\top w_\star^{(k)} - Q_{s,a}^{(k)}\right) \\ & + \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} d_s^* \pi_{s,a}^* \phi_{s,a}^\top\left(w_\star^{(k)} - w^{(k)}\right) + \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} d_s^* \pi_{s,a}^*\left(Q_{s,a}^{(k)} - \phi_{s,a}^\top w_\star^{(k)}\right) \\ = & \text{ⓐ} + \text{ⓑ} + \text{ⓒ} + \text{ⓓ}, \end{aligned} \tag{D.20}$$

where ⓐ, ⓑ, ⓒ and ⓓ are defined in (D.13).

Plugging (D.17) with the decomposition (D.18) and (D.19) with the decomposition (D.20) into (D.16) concludes the proof. □

Consequently, the convergence analysis of Q-NPG (Theorem 5.5, 5.6 and 5.9) will be obtained by upper bounding the absolute values of ①, ②, ③, ④, ⓐ, ⓑ, ⓒ, ⓓ in (D.13) with different set of assumptions (assumptions in Theorem 5.5 or assumptions in Theorem 5.9) and with different step size scheme (geometrically increasing step size for Theorem 5.5 and 5.9 or constant step size for Theorem 5.6).

### D.4.2 Proof of Theorem 5.5

*Proof.* From (D.13) in Lemma D.9, we will upper bound |①| and |③| by the statistical error assumption (5.20) and upper bound |②| and |④| by using the transfer error assumption (5.23).

Indeed, to upper bound |①|, by Cauchy-Schwartz's inequality, we have

$$|①| \quad \leq \quad \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_s^{(k+1)} \pi_{s,a}^{(k+1)} \left|\phi_{s,a}^\top\left(w^{(k)} - w_\star^{(k)}\right)\right|$$

$$
\begin{aligned}
&\leq \sqrt{\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \frac{\left(d_s^{(k+1)}\right)^2\left(\pi_{s,a}^{(k+1)}\right)^2}{d_s^*\cdot \mathrm{Unif}_{\mathcal{A}}(a)} \cdot \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} d_s^*\cdot \mathrm{Unif}_{\mathcal{A}}(a)\left(\phi_{s,a}^\top\left(w^{(k)}-w_\star^{(k)}\right)\right)^2} \\
&\overset{(5.24)}{=} \sqrt{\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \frac{\left(d_s^{(k+1)}\right)^2\left(\pi_{s,a}^{(k+1)}\right)^2}{d_s^*\cdot \mathrm{Unif}_{\mathcal{A}}(a)} \left\|w^{(k)}-w_\star^{(k)}\right\|_{\Sigma_{\tilde{d}^*}}^2} \\
&\leq \sqrt{\mathbb{E}_{s\sim d^*}\left[\left(\frac{d_s^{(k+1)}}{d_s^*}\right)^2\right]|\mathcal{A}|\left\|w^{(k)}-w_\star^{(k)}\right\|_{\Sigma_{\tilde{d}^*}}^2} \\
&\overset{(5.26)}{\leq} \sqrt{C_\rho|\mathcal{A}|\left\|w^{(k)}-w_\star^{(k)}\right\|_{\Sigma_{\tilde{d}^*}}^2},
\end{aligned} \tag{D.21}
$$

where the second inequality is obtained by Cauchy-Schwartz's inequality, and the third inequality is obtained by the following inequality

$$
\sum_{a\in\mathcal{A}}\left(\pi_{s,a}^{(k+1)}\right)^2 \leq \sum_{a\in\mathcal{A}}\pi_{s,a}^{(k+1)} = 1. \tag{D.22}
$$

Then, by using Assumption 5.3 with the definition of $\kappa_\nu$, (D.21) is upper bounded by

$$
\begin{aligned}
|①| &\overset{(5.25)}{\leq} \sqrt{C_\rho|\mathcal{A}|\kappa_\nu\left\|w^{(k)}-w_\star^{(k)}\right\|_{\Sigma_\nu}^2} \\
&\overset{(5.6)}{\leq} \sqrt{\frac{C_\rho|\mathcal{A}|\kappa_\nu}{1-\gamma}\left\|w^{(k)}-w_\star^{(k)}\right\|_{\Sigma_{\tilde{d}^{(k)}}}^2},
\end{aligned} \tag{D.23}
$$

where we use the shorthand

$$
\Sigma_{\tilde{d}^{(k)}} \overset{\text{def}}{=} \mathbb{E}_{(s,a)\sim\tilde{d}^{(k)}}\left[\phi_{s,a}\phi_{s,a}^\top\right]. \tag{D.24}
$$

Besides, by the first-order optimality conditions for the optima $w_\star^{(k)} \in \operatorname{argmin}_w L_Q(w,\theta^{(k)},\tilde{d}^{(k)})$, we have

$$
(w-w_\star^{(k)})^\top\nabla_w L_Q(w_\star^{(k)},\theta^{(k)},\tilde{d}^{(k)}) \geq 0, \qquad \text{for all } w\in\mathbb{R}^m. \tag{D.25}
$$

Therefore, for all $w\in\mathbb{R}^m$,

$$
\begin{aligned}
& L_Q(w,\theta^{(k)},\tilde{d}^{(k)}) - L_Q(w_\star^{(k)},\theta^{(k)},\tilde{d}^{(k)}) \\
=\ & \mathbb{E}_{(s,a)\sim\tilde{d}^{(k)}}\left[\left(\phi_{s,a}^\top w-\phi_{s,a}^\top w_\star^{(k)}+\phi_{s,a}^\top w_\star^{(k)}-Q_{s,a}^{(k)}\right)^2\right] - L_Q(w_\star^{(k)},\theta^{(k)},\tilde{d}^{(k)}) \\
=\ & \mathbb{E}_{(s,a)\sim\tilde{d}^{(k)}}\left[(\phi_{s,a}^\top w-\phi_{s,a}^\top w_\star^{(k)})^2\right] + 2(w-w_\star^{(k)})^\top\mathbb{E}_{(s,a)\sim\tilde{d}^{(k)}}\left[(\phi_{s,a}^\top w_\star^{(k)}-Q_{s,a}^{(k)})\phi_{s,a}\right] \\
=\ & \left\|w-w_\star^{(k)}\right\|_{\Sigma_{\tilde{d}^{(k)}}}^2 + (w-w_\star^{(k)})^\top\nabla_w L_Q(w_\star^{(k)},\theta^{(k)},\tilde{d}^{(k)})
\end{aligned}
$$

$$\overset{(D.25)}{\geq} \quad \left\|w - w_\star^{(k)}\right\|^2_{\Sigma_{\tilde{d}^{(k)}}}. \tag{D.26}$$

Define

$$\epsilon_{\text{stat}}^{(k)} \overset{\text{def}}{=} L_Q(w^{(k)}, \theta^{(k)}, \tilde{d}^{(k)}) - L_Q(w_\star^{(k)}, \theta^{(k)}, \tilde{d}^{(k)}).$$

Note that from (5.20), we have

$$\mathbb{E}\left[\epsilon_{\text{stat}}^{(k)}\right] \leq \epsilon_{\text{stat}}. \tag{D.27}$$

Plugging (D.26) into (D.23), we have

$$|①| \leq \sqrt{\frac{C_\rho |\mathcal{A}| \kappa_\nu}{1-\gamma} \epsilon_{\text{stat}}^{(k)}}. \tag{D.28}$$

Similar to (D.21), we get the same upper bound for |③| by just replacing $\pi_{s,a}^{(k+1)}$ into $\pi_{s,a}^{(k)}$. That is,

$$|③| \leq \sqrt{\frac{C_\rho |\mathcal{A}| \kappa_\nu}{1-\gamma} \epsilon_{\text{stat}}^{(k)}}. \tag{D.29}$$

To upper bound |②| and |④|, we introduce the following term

$$\epsilon_{\text{bias}}^{(k)} \overset{\text{def}}{=} L_Q(w_\star^{(k)}, \theta^{(k)}, \tilde{d}^*).$$

Note that from (5.23), we have

$$\mathbb{E}\left[\epsilon_{\text{bias}}^{(k)}\right] \leq \epsilon_{\text{bias}}. \tag{D.30}$$

By Cauchy-Schwartz's inequality, we have

$$\begin{aligned}
|②| \quad &\leq \quad \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} d_s^{(k+1)} \pi_{s,a}^{(k+1)} \left| \phi_{s,a}^\top w_\star^{(k)} - Q_{s,a}^{(k)} \right| \\
&\leq \quad \sqrt{\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \frac{\left(d_s^{(k+1)}\right)^2 \left(\pi_{s,a}^{(k+1)}\right)^2}{d_s^* \cdot \text{Unif}_{\mathcal{A}}(a)} \cdot \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} d_s^* \cdot \text{Unif}_{\mathcal{A}}(a) \left(\phi_{s,a}^\top w_\star^{(k)} - Q_{s,a}^{(k)}\right)^2} \\
&= \quad \sqrt{\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \frac{\left(d_s^{(k+1)}\right)^2 \left(\pi_{s,a}^{(k+1)}\right)^2}{d_s^* \cdot \text{Unif}_{\mathcal{A}}(a)} \cdot \epsilon_{\text{bias}}^{(k)}}
\end{aligned}$$

$$\overset{(D.22)}{\leq} \sqrt{\mathbb{E}_{s\sim d^*}\left[\left(\frac{d_s^{(k+1)}}{d_s^*}\right)^2\right]|\mathcal{A}|\epsilon_{\text{bias}}^{(k)}} \overset{(5.26)}{\leq} \sqrt{C_\rho|\mathcal{A}|\epsilon_{\text{bias}}^{(k)}}. \tag{D.31}$$

Similar to (D.31), we get the same upper bound for $|④|$ by just replacing $\pi_{s,a}^{(k+1)}$ into $\pi_{s,a}^{(k)}$. That is,

$$|④| \leq \sqrt{C_\rho|\mathcal{A}|\epsilon_{\text{bias}}^{(k)}}. \tag{D.32}$$

Next, we will upper bound the absolute values of ⓐ, ⓑ, ⓒ and ⓓ of (D.13) separately by using again the statistical error (5.20) and by using the transfer error assumption (5.23).

Indeed, to upper bound $|ⓐ|$, by Cauchy-Schwartz's inequality, we have

$$\begin{aligned}
|ⓐ| &\leq \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} d_s^*\pi_{s,a}^{(k)}\left|\phi_{s,a}^\top\left(w^{(k)}-w_\star^{(k)}\right)\right| \\
&\leq \sqrt{\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}\frac{(d_s^*)^2\left(\pi_{s,a}^{(k)}\right)^2}{d_s^*\cdot\text{Unif}_{\mathcal{A}}(a)}\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} d_s^*\cdot\text{Unif}_{\mathcal{A}}(a)\left(\phi_{s,a}^\top\left(w^{(k)}-w_\star^{(k)}\right)\right)^2} \\
&\overset{(5.24)}{=} \sqrt{\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}\frac{(d_s^*)^2\left(\pi_{s,a}^{(k)}\right)^2}{d_s^*\cdot\text{Unif}_{\mathcal{A}}(a)}\left\|w^{(k)}-w_\star^{(k)}\right\|_{\Sigma_{\tilde{d}^*}}^2} \\
&\overset{(D.22)}{\leq} \sqrt{|\mathcal{A}|\left\|w^{(k)}-w_\star^{(k)}\right\|_{\Sigma_{\tilde{d}^*}}^2}.
\end{aligned}$$

From the definition of $\kappa_\nu$, we further obtain

$$\begin{aligned}
|ⓐ| &\overset{(5.25)}{\leq} \sqrt{|\mathcal{A}|\kappa_\nu\left\|w^{(k)}-w_\star^{(k)}\right\|_{\Sigma_\nu}^2} \\
&\overset{(5.6)}{\leq} \sqrt{\frac{|\mathcal{A}|\kappa_\nu}{1-\gamma}\left\|w^{(k)}-w_\star^{(k)}\right\|_{\Sigma_{\tilde{d}^{(k)}}}^2} \\
&\overset{(D.26)}{\leq} \sqrt{\frac{|\mathcal{A}|\kappa_\nu}{1-\gamma}\epsilon_{\text{stat}}^{(k)}}.
\end{aligned} \tag{D.33}$$

Similar to (D.33), we get the same upper bound for $|ⓒ|$ by just replacing $\pi_{s,a}^{(k)}$ into $\pi_{s,a}^*$. That is,

$$|ⓒ| \leq \sqrt{\frac{|\mathcal{A}|\kappa_\nu}{1-\gamma}\epsilon_{\text{stat}}^{(k)}}. \tag{D.34}$$

To upper bound $|ⓑ|$, by Cauchy-Schwartz's inequality, we have

$$\begin{aligned}
|ⓑ| &\leq \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} d_s^* \pi_{s,a}^{(k)} \left|\left(\phi_{s,a}^\top w_\star^{(k)} - Q_{s,a}^{(k)}\right)\right| \\
&\leq \sqrt{\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \frac{(d_s^*)^2 \left(\pi_{s,a}^{(k)}\right)^2}{d_s^* \cdot \mathrm{Unif}_{\mathcal{A}}(a)} \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} d_s^* \cdot \mathrm{Unif}_{\mathcal{A}}(a) \left(\phi_{s,a}^\top w_\star^{(k)} - Q_{s,a}^{(k)}\right)^2} \\
&= \sqrt{\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \frac{(d_s^*)^2 \left(\pi_{s,a}^{(k)}\right)^2}{d_s^* \cdot \mathrm{Unif}_{\mathcal{A}}(a)} \epsilon_{\mathrm{bias}}^{(k)}} \\
&\overset{(D.22)}{\leq} \sqrt{|\mathcal{A}|\epsilon_{\mathrm{bias}}^{(k)}}.
\end{aligned} \tag{D.35}$$

Similar to (D.35), we get the same upper bound for $|ⓓ|$ by just replacing $\pi_{s,a}^{(k)}$ into $\pi_{s,a}^*$. That is,

$$|ⓓ| \leq \sqrt{|\mathcal{A}|\epsilon_{\mathrm{bias}}^{(k)}}. \tag{D.36}$$

Plugging all the upper bounds (D.28) of $|①|$, (D.31) of $|②|$, (D.29) of $|③|$, (D.32) of $|④|$, (D.33) of $|ⓐ|$, (D.35) of $|ⓑ|$, (D.34) of $|ⓒ|$ and (D.36) of $|ⓓ|$ into (D.13) yields

$$\vartheta_\rho \left(\delta_{k+1} - \delta_k\right) + \delta_k \leq \frac{D_k^*}{(1-\gamma)\eta_k} - \frac{D_{k+1}^*}{(1-\gamma)\eta_k} + \frac{2\sqrt{|\mathcal{A}|}\left(\vartheta_\rho\sqrt{C_\rho}+1\right)}{1-\gamma}\left(\sqrt{\frac{\kappa_\nu}{1-\gamma}\epsilon_{\mathrm{stat}}^{(k)}} + \sqrt{\epsilon_{\mathrm{bias}}^{(k)}}\right), \tag{D.37}$$

where $\delta_k \overset{\mathrm{def}}{=} V_\rho^{(k)} - V_\rho(\pi^*)$. Dividing both sides by $\vartheta_\rho$ and rearranging terms, we get

$$\begin{aligned}
\delta_{k+1} + \frac{D_{k+1}^*}{(1-\gamma)\eta_k\vartheta_\rho} \leq{}& \left(1 - \frac{1}{\vartheta_\rho}\right)\left(\delta_k + \frac{D_k^*}{(1-\gamma)\eta_k(\vartheta_\rho - 1)}\right) \\
&+ \frac{2\sqrt{|\mathcal{A}|}\left(\sqrt{C_\rho} + \frac{1}{\vartheta_\rho}\right)}{1-\gamma}\left(\sqrt{\frac{\kappa_\nu}{1-\gamma}\epsilon_{\mathrm{stat}}^{(k)}} + \sqrt{\epsilon_{\mathrm{bias}}^{(k)}}\right).
\end{aligned}$$

If the step sizes satisfy $\eta_{k+1}(\vartheta_\rho - 1) \geq \eta_k\vartheta_\rho$, which is implied by $\eta_{k+1} \geq \eta_k/\gamma$ and (5.21), then

$$\begin{aligned}
\delta_{k+1} + \frac{D_{k+1}^*}{(1-\gamma)\eta_{k+1}(\vartheta_\rho - 1)} &\leq \left(1 - \frac{1}{\vartheta_\rho}\right)\left(\delta_k + \frac{D_k^*}{(1-\gamma)\eta_k(\vartheta_\rho - 1)}\right) \\
&\quad + \frac{2\sqrt{|\mathcal{A}|}\left(\sqrt{C_\rho} + \frac{1}{\vartheta_\rho}\right)}{1-\gamma}\left(\sqrt{\frac{\kappa_\nu}{1-\gamma}\epsilon_{\mathrm{stat}}^{(k)}} + \sqrt{\epsilon_{\mathrm{bias}}^{(k)}}\right) \\
&\leq \left(1 - \frac{1}{\vartheta_\rho}\right)^{k+1}\left(\delta_0 + \frac{D_0^*}{(1-\gamma)\eta_0(\vartheta_\rho - 1)}\right)
\end{aligned}$$

$$+\sum_{t=0}^{k}\left(1-\frac{1}{\vartheta_\rho}\right)^{k-t}\frac{2\sqrt{|\mathcal{A}|}\left(\sqrt{C_\rho}+\frac{1}{\vartheta_\rho}\right)}{1-\gamma}\left(\sqrt{\frac{\kappa_\nu}{1-\gamma}\epsilon_{\text{stat}}^{(t)}}+\sqrt{\epsilon_{\text{bias}}^{(t)}}\right).$$

Finally, by choosing $\eta_0 \geq \frac{1-\gamma}{\gamma}D_0^*$ and using the fact that

$$(1-\gamma)(\vartheta_\rho-1) \overset{(5.21)}{\geq} (1-\gamma)\left(\frac{1}{1-\gamma}-1\right)=\gamma,$$

we obtain

$$\begin{aligned}\delta_k &\leq \delta_k + \frac{D_k^*}{(1-\gamma)\eta_k\vartheta_\rho}\\ &\leq \left(1-\frac{1}{\vartheta_\rho}\right)^k\frac{2}{1-\gamma}+\frac{2\sqrt{|\mathcal{A}|}\left(\sqrt{C_\rho}+\frac{1}{\vartheta_\rho}\right)}{1-\gamma}\sum_{t=0}^{k-1}\left(1-\frac{1}{\vartheta_\rho}\right)^{k-1-t}\left(\sqrt{\frac{\kappa_\nu}{1-\gamma}\epsilon_{\text{stat}}^{(t)}}+\sqrt{\epsilon_{\text{bias}}^{(t)}}\right).\end{aligned}$$

Taking the total expectation with respect to the randomness in the sequence of the iterates $w^{(0)},\cdots,w^{(k-1)}$, we have

$$\begin{aligned}&\mathbb{E}\left[V_\rho(\pi^{(k)})\right]-V_\rho(\pi^*)\\ \leq\ &\left(1-\frac{1}{\vartheta_\rho}\right)^k\frac{2}{1-\gamma}\\ &+\frac{2\sqrt{|\mathcal{A}|}\left(\sqrt{C_\rho}+\frac{1}{\vartheta_\rho}\right)}{1-\gamma}\sum_{t=0}^{k-1}\left(1-\frac{1}{\vartheta_\rho}\right)^{k-1-t}\left(\mathbb{E}\left[\sqrt{\frac{\kappa_\nu}{1-\gamma}\epsilon_{\text{stat}}^{(t)}}\right]+\mathbb{E}\left[\sqrt{\epsilon_{\text{bias}}^{(t)}}\right]\right)\\ \leq\ &\left(1-\frac{1}{\vartheta_\rho}\right)^k\frac{2}{1-\gamma}\\ &+\frac{2\sqrt{|\mathcal{A}|}\left(\sqrt{C_\rho}+\frac{1}{\vartheta_\rho}\right)}{1-\gamma}\sum_{t=0}^{k-1}\left(1-\frac{1}{\vartheta_\rho}\right)^{k-1-t}\left(\sqrt{\frac{\kappa_\nu}{1-\gamma}\mathbb{E}\left[\epsilon_{\text{stat}}^{(t)}\right]}+\sqrt{\mathbb{E}\left[\epsilon_{\text{bias}}^{(t)}\right]}\right)\\ \overset{(D.27)+(D.30)}{\leq}\ &\left(1-\frac{1}{\vartheta_\rho}\right)^k\frac{2}{1-\gamma}\\ &+\frac{2\sqrt{|\mathcal{A}|}\left(\sqrt{C_\rho}+\frac{1}{\vartheta_\rho}\right)}{1-\gamma}\sum_{t=0}^{k-1}\left(1-\frac{1}{\vartheta_\rho}\right)^{k-1-t}\left(\sqrt{\frac{\kappa_\nu}{1-\gamma}\epsilon_{\text{stat}}}+\sqrt{\epsilon_{\text{bias}}}\right)\\ \leq\ &\left(1-\frac{1}{\vartheta_\rho}\right)^k\frac{2}{1-\gamma}+\frac{2\sqrt{|\mathcal{A}|}\left(\vartheta_\rho\sqrt{C_\rho}+1\right)}{1-\gamma}\left(\sqrt{\frac{\kappa_\nu}{1-\gamma}\epsilon_{\text{stat}}}+\sqrt{\epsilon_{\text{bias}}}\right),\end{aligned}$$

where the second inequality is obtained by Jensen's inequality. This concludes the proof. □

### D.4.3 Proof of Theorem 5.6

*Proof.* By (D.37) and using a constant step size $\eta$, we have

$$\vartheta_\rho \left(\delta_{k+1} - \delta_k\right) + \delta_k \leq \frac{D_k^*}{(1-\gamma)\eta} - \frac{D_{k+1}^*}{(1-\gamma)\eta} + \frac{2\sqrt{|\mathcal{A}|}\left(\vartheta_\rho\sqrt{C_\rho}+1\right)}{1-\gamma}\left(\sqrt{\frac{\kappa_\nu}{1-\gamma}\epsilon_{\text{stat}}^{(k)}} + \sqrt{\epsilon_{\text{bias}}^{(k)}}\right).$$

Taking the total expectation with respect to the randomness in the sequence of the iterates $w^{(0)}, \cdots, w^{(k-1)}$, summing up from 0 to $k-1$ and rearranging terms, we have

$$\vartheta_\rho \mathbb{E}\left[\delta_k\right] + \sum_{t=0}^{k-1} \mathbb{E}\left[\delta_t\right] \leq \frac{D_0^*}{(1-\gamma)\eta} + \vartheta_\rho \delta_0 + k \cdot \frac{2\sqrt{|\mathcal{A}|}\left(\vartheta_\rho\sqrt{C_\rho}+1\right)}{1-\gamma}\left(\sqrt{\frac{\kappa_\nu}{1-\gamma}\epsilon_{\text{stat}}} + \sqrt{\epsilon_{\text{bias}}}\right),$$

where we use the following inequalities

$$\mathbb{E}\left[\sqrt{\epsilon_{\text{stat}}^{(t)}}\right] \leq \sqrt{\mathbb{E}\left[\epsilon_{\text{stat}}^{(t)}\right]} \overset{\text{(D.27)}}{\leq} \sqrt{\epsilon_{\text{stat}}},$$
$$\mathbb{E}\left[\sqrt{\epsilon_{\text{bias}}^{(t)}}\right] \leq \sqrt{\mathbb{E}\left[\epsilon_{\text{bias}}^{(t)}\right]} \overset{\text{(D.30)}}{\leq} \sqrt{\epsilon_{\text{bias}}}.$$

Finally, dropping the positive term $\mathbb{E}\left[\delta_k\right]$ on the left hand side as $\pi^*$ is the optimal policy and dividing both side by $k$ yields

$$\frac{1}{k}\sum_{t=0}^{k-1} \mathbb{E}\left[V_\rho(\pi^{(t)})\right] - V_\rho(\pi^*) \leq \frac{D_0^*}{(1-\gamma)\eta k} + \frac{2\vartheta_\rho}{(1-\gamma)k} + \frac{2\sqrt{|\mathcal{A}|}\left(\vartheta_\rho\sqrt{C_\rho}+1\right)}{1-\gamma}\left(\sqrt{\frac{\kappa_\nu}{1-\gamma}\epsilon_{\text{stat}}} + \sqrt{\epsilon_{\text{bias}}}\right).$$

□

### D.4.4 Proof of Theorem 5.9

*Proof.* Similar to the proof of Theorem 5.5, by Lemma D.9, we upper bound the absolute values of ①, ②, ③, ④, ⓐ, ⓑ, ⓒ, ⓓ introduced in (D.13), separately, with the set of assumptions in Theorem 5.9.

In comparison with the proof of Theorem 5.5, we will also upper bound |①|, |③|, |ⓐ| and |ⓒ| by the statistical error assumption (5.20) as in the proof of Theorem 5.5. However, we will upper bound |②|, |④|, |ⓑ| and |ⓓ| by using the approximation error assumption (5.28) instead of the transfer error assumption (5.23).

To upper bound |①|, by Cauchy-Schwartz's inequality, we get

$$\begin{aligned}
|①| &\leq \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_s^{(k+1)}\pi_{s,a}^{(k+1)}\left|\phi_{s,a}^\top\left(w^{(k)}-w_\star^{(k)}\right)\right| \\
&\leq \sqrt{\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}\frac{\left(d_s^{(k+1)}\right)^2\left(\pi_{s,a}^{(k+1)}\right)^2}{\tilde{d}_{s,a}^{(k)}}\cdot\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}\tilde{d}_{s,a}^{(k)}\left(\phi_{s,a}^\top\left(w^{(k)}-w_\star^{(k)}\right)\right)^2} \\
&\overset{(D.24)}{=} \sqrt{\mathbb{E}_{(s,a)\sim\tilde{d}^{(k)}}\left[\left(\frac{d_s^{(k+1)}\pi_{s,a}^{(k+1)}}{\tilde{d}_{s,a}^{(k)}}\right)^2\right]\left\|w^{(k)}-w_\star^{(k)}\right\|_{\Sigma_{\tilde{d}^{(k)}}}^2} \\
&\overset{(5.29)}{\leq} \sqrt{C_\nu\left\|w^{(k)}-w_\star^{(k)}\right\|_{\Sigma_{\tilde{d}^{(k)}}}^2} \\
&\overset{(D.26)}{\leq} \sqrt{C_\nu\epsilon_{\text{stat}}^{(k)}}.
\end{aligned}$$

Similar to |①|, by using Assumption 5.8 and Cauchy-Schwartz's inequality, and by simply replacing $\pi^{(k+1)}$ into $\pi^{(k)}$ or $\pi^*$ and replacing $d^{(k+1)}$ into $d^*$, we obtain the same upper bound of |③|, |ⓐ| and |ⓒ|, that is

$$|③|, |ⓐ|, |ⓒ| \leq \sqrt{C_\nu\epsilon_{\text{stat}}^{(k)}}.$$

Next, we define

$$\epsilon_{\text{approx}}^{(k)} \overset{\text{def}}{=} L_Q(w_\star^{(k)}, \theta^{(k)}, \tilde{d}^{(k)})$$

By Assumption 5.7, we know that

$$\mathbb{E}\left[\epsilon_{\text{approx}}^{(k)}\right] \leq \epsilon_{\text{approx}}.$$

To upper bound |②|, by Cauchy-Schwartz's inequality, we have

$$\begin{aligned}
|②| &\leq \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} d_s^{(k+1)}\pi_{s,a}^{(k+1)}\left|\phi_{s,a}^\top w_\star^{(k)}-Q_{s,a}^{(k)}\right| \\
&\leq \sqrt{\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}\frac{\left(d_s^{(k+1)}\right)^2\left(\pi_{s,a}^{(k+1)}\right)^2}{\tilde{d}_{s,a}^{(k)}}\cdot\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}}\tilde{d}_{s,a}^{(k)}\left(\phi_{s,a}^\top w_\star^{(k)}-Q_{s,a}^{(k)}\right)^2} \\
&= \sqrt{\mathbb{E}_{(s,a)\sim\tilde{d}^{(k)}}\left[\left(\frac{d_s^{(k+1)}\pi_{s,a}^{(k+1)}}{\tilde{d}_{s,a}^{(k)}}\right)^2\right]\cdot\epsilon_{\text{approx}}^{(k)}} \\
&\overset{(5.29)}{\leq} \sqrt{C_\nu\epsilon_{\text{approx}}^{(k)}}.
\end{aligned}$$

Similar to |②|, by using Assumption 5.7 and Cauchy-Schwartz's inequality, and by simply replacing $\pi^{(k+1)}$ into $\pi^{(k)}$ or $\pi^*$ and replacing $d^{(k+1)}$ into $d^*$, we obtain the same upper bound for |④|, |ⓑ| and |ⓓ|, that is

$$|④|, |ⓑ|, |ⓓ| \leq \sqrt{C_\nu \epsilon_{\text{approx}}^{(k)}}.$$

Consequently, plugging all these upper bounds into (D.13) leads to the following recurrent inequality

$$\vartheta_\rho \left(\delta_{k+1} - \delta_k\right) + \delta_k \leq \frac{D_k^*}{(1-\gamma)\eta_k} - \frac{D_{k+1}^*}{(1-\gamma)\eta_k} + \frac{2\sqrt{C_\nu}\left(\vartheta_\rho + 1\right)}{1-\gamma}\left(\sqrt{\epsilon_{\text{stat}}^{(k)}} + \sqrt{\epsilon_{\text{approx}}^{(k)}}\right).$$

By using the same increasing step size as in Theorem 5.5 and following the same arguments in the proof of Theorem 5.5 after (D.37), we obtain the final performance bound with the linear convergence rate

$$\mathbb{E}\left[V_\rho(\pi^{(k)})\right] - V_\rho(\pi^*) \leq \left(1 - \frac{1}{\vartheta_\rho}\right)^k \frac{2}{1-\gamma} + \frac{2\sqrt{C_\nu}\left(\vartheta_\rho + 1\right)}{1-\gamma}\left(\sqrt{\epsilon_{\text{stat}}} + \sqrt{\epsilon_{\text{approx}}}\right).$$

□

### D.4.5 Proof of Corollary 5.11

In order to better understand our proof, we first identify an issue appeared in the sample complexity analysis of Q-NPG in Agarwal et al. (2021, Corollay 26). Agarwal et al. (2021) adopts the optimization results of Shalev-Shwartz and Ben-David (2014, Theorem 14.8) where the stochastic gradient $\widehat{\nabla} L_Q(w, \theta, \tilde{d}^\theta)$ in (D.12) needs to be bounded. However, although they consider a projection step for the iterate $w_t$ and assume that the feature map $\phi_{s,a}$ is bounded, $\widehat{\nabla} L_Q(w, \theta, \tilde{d}^\theta)$ is still not guaranteed to be bounded. Indeed, recall the stochastic gradient of the function $L_Q$ in (D.12)

$$\widehat{\nabla}_w L_Q(w, \theta, \tilde{d}^\theta) = 2\left(w^\top \phi_{s,a} - \widehat{Q}_{s,a}(\theta)\right)\phi_{s,a}.$$

They incorrectly use the argument that $w$, $\phi_{s,a}$ and $\widehat{Q}_{s,a}(\theta)$ are bounded to imply that $\left\|\widehat{\nabla}_w L_Q(w, \theta, \tilde{d}^\theta)\right\|$ is bounded. In fact, $\widehat{Q}_{s,a}(\theta)$ can be unbounded even though $\mathbb{E}\left[\widehat{Q}_{s,a}(\theta)\right] = Q_{s,a}(\theta) \in \left[0, \frac{1}{1-\gamma}\right]$ is bounded. To see this, we can rewrite $\widehat{Q}_{s,a}(\theta)$ from (D.8) as

$$\widehat{Q}_{s,a}(\theta) = \sum_{t=0}^{H} c(s_t, a_t),$$

with $(s_0, a_0) = (s, a) \sim \tilde{d}^\theta$ and $H$ is the length of the sampled trajectory for estimating $Q_{s,a}(\theta)$ in Algorithm 13. From Algorithm 13 and from the proof of Lemma D.5, we know that the probability of $H = k + 1$ is that

$$\Pr(H = k + 1) = (1 - \gamma)\gamma^k.$$

So, with exponentially decreasing low probability, $H$ can be unbounded. Consequently, $|\widehat{Q}_{s,a}(\theta)|$ upper bounded by $H$ is not guaranteed to be bounded.

**Proof sketch.** Instead, we adopt the optimization results of Bach and Moulines (2013, Theorem 1) (see also Theorem D.15), which does not require the boundedness of the stochastic gradient. However, in our following proof, we can verify that $\mathbb{E}\left[\widehat{Q}_{s,a}(\theta)^2\right]$ is bounded even though $\widehat{Q}_{s,a}(\theta)$ is unbounded. As to verify the condition (vi) in Theorem D.15 in our proof, i.e., the covariance of the stochastic gradient at the optimum is upper bounded by the covariance of the feature map up to a finite constant, we use a conditional expectation computation trick to separate the correlated random variables $\widehat{Q}_{s,a}(\theta)$ and $\phi_{s,a}$ with $(s, a) \sim \tilde{d}^\theta$ appeared in the stochastic gradient.

*Proof.* From Theorem 5.9, it remains to upper bound the statistical error $\sqrt{\epsilon_{\text{stat}}}$ produced from the `Q-NPG-SGD` procedure (Algorithm 16) for each iteration $k$. We suppress the superscript $(k)$. Let $w_{\text{out}}$ be the output of $T$ steps `Q-NPG-SGD` with the constant step size $\frac{1}{2B^2}$ and the initialization $w_0 = 0$, and let $w_\star \in \operatorname{argmin}_w L_Q(w, \theta, \tilde{d}^\theta)$ be the exact minimizer. To upper bound $\epsilon_{\text{stat}}$ from (5.20), we aim to apply the standard analysis for the averaged SGD, i.e., Theorem D.15. Now we verify all the assumptions in order for `Q-NPG-SGD`.

First, (i) is verified by considering the Euclidean space $\mathcal{H} = \mathbb{R}^m$.

The observations $\left(\phi_{s,a}\,,\ \widehat{Q}_{s,a}(\theta)\phi_{s,a}\right) \in \mathbb{R}^m \times \mathbb{R}^m$ are independent and identically distributed, sampled from Algorithm 13. Thus, (ii) is verified with $x_n = \phi_{s,a} \in \mathbb{R}^m$ and $z_n = \widehat{Q}_{s,a}(\theta)\phi_{s,a} \in \mathbb{R}^m$.

As the feature map $\|\phi_{s,a}\| \le B$, we have $\mathbb{E}\left[\|\phi_{s,a}\|^2\right]$ finite. From (5.32), we know that the covariance $\mathbb{E}\left[\phi_{s,a}\phi_{s,a}^\top\right]$ is invertible. To verify (iii), it remains to verify that $\mathbb{E}\left[\left\|\widehat{Q}_{s,a}(\theta)\phi_{s,a}\right\|^2\right]$ is finite. Indeed, by using $\|\phi_{s,a}\| \le B$, we have

$$\mathbb{E}\left[\left\|\widehat{Q}_{s,a}(\theta)\phi_{s,a}\right\|^2\right] \le B^2\mathbb{E}\left[\widehat{Q}_{s,a}(\theta)^2\right].$$

Thus, it remains to show $\mathbb{E}\left[\left(\widehat{Q}_{s,a}(\theta)\right)^2\right]$ finite for (iii). From (D.8), we rewrite $\widehat{Q}_{s,a}(\theta)$ as

$$\widehat{Q}_{s,a}(\theta) = \sum_{t=0}^{H} c(s_t, a_t),$$

with $(s_0, a_0) = (s, a) \sim \tilde{d}^\theta$ and $H$ is the length of the trajectory for estimating $Q_{s,a}(\theta)$. Thus, (iii) is verified as the variance of $\widehat{Q}_{s,a}(\theta)$ is upper bounded by

$$\begin{aligned}
\mathbb{E}\left[\left(\widehat{Q}_{s,a}(\theta)\right)^2\right] &= \mathbb{E}_{(s,a)\sim\tilde{d}^\theta}\left[\sum_{k=0}^{\infty} \Pr(H=k)\mathbb{E}\left[\left(\sum_{t=0}^{k} c(s_t,a_t)\right)^2 \mid H=k, s_0=s, a_0=a\right]\right] \\
&= \mathbb{E}_{(s,a)\sim\tilde{d}^\theta}\left[(1-\gamma)\sum_{k=0}^{\infty}\gamma^k \mathbb{E}\left[\left(\sum_{t=0}^{k} c(s_t,a_t)\right)^2 \mid H=k, s_0=s, a_0=a\right]\right] \\
&\leq \mathbb{E}_{(s,a)\sim\tilde{d}^\theta}\left[(1-\gamma)\sum_{k=0}^{\infty}\gamma^k (k+1)^2\right] \overset{\text{Lemma C.2}}{\leq} \frac{2}{(1-\gamma)^2},
\end{aligned} \tag{D.38}$$

where the first inequality is obtained as $|c(s_t, a_t)| \in [0, 1]$ for all $(s_t, a_t) \in \mathcal{S} \times \mathcal{A}$.

Next, we introduce the residual

$$\xi \overset{\text{def}}{=} \left(\widehat{Q}_{s,a}(\theta) - w_\star^\top \phi_{s,a}\right)\phi_{s,a} \overset{\text{(D.12)}}{=} \frac{1}{2}\widehat{\nabla}_w L_Q(w_\star, \theta, \tilde{d}^\theta). \tag{D.39}$$

From Lemma D.8, we know that

$$\mathbb{E}\left[\widehat{\nabla}_w L_Q(w_\star, \theta, \tilde{d}^\theta)\right] = \nabla_w L_Q(w_\star, \theta, \tilde{d}^\theta).$$

So, we have that

$$\mathbb{E}\left[\xi\right] = \frac{1}{2}\nabla_w L_Q(w_\star, \theta, \tilde{d}^\theta) = 0,$$

where the last equality is obtained as $w_\star$ is the exact minimizer of the loss function $L_Q$. Thus, (iv) is verified with that $f$ is $\frac{1}{2}L_Q$, $\xi_n$ is $\xi$ and $\theta$ is $w$ in our context.

From `Q-NPG-SGD` update D.12, we have (v) verified with step size $\alpha/2$ in our context.

Finally, for (vi), from the boundedness of the feature map $\|\phi_{s,a}\| \leq B$, we take $R = B$ such that $\mathbb{E}\left[\|\phi_{s,a}\|^2 \phi_{s,a}\phi_{s,a}^\top\right] \leq B^2\mathbb{E}\left[\phi_{s,a}\phi_{s,a}^\top\right]$. It remains to find $\sigma > 0$ such that

$$\mathbb{E}\left[\xi\xi^\top\right] \leq \sigma^2\mathbb{E}\left[\phi_{s,a}\phi_{s,a}^\top\right].$$

We rewrite the covariance of $\xi$ as

$$\mathbb{E}\left[\xi\xi^\top\right] \overset{\text{(D.39)}}{=} \mathbb{E}\left[\left(\widehat{Q}_{s,a}(\theta) - w_\star^\top\phi_{s,a}\right)^2 \phi_{s,a}\phi_{s,a}^\top\right]$$

$$
\begin{aligned}
&= \mathbb{E}_{(s,a)\sim\tilde{d}^\theta}\left[\left(\widehat{Q}_{s,a}(\theta) - w_\star^\top \phi_{s,a}\right)^2 \phi_{s,a}\phi_{s,a}^\top \mid s,a\right] \\
&= \mathbb{E}_{(s,a)\sim\tilde{d}^\theta}\left[\mathbb{E}\left[\left(\widehat{Q}_{s,a}(\theta) - w_\star^\top \phi_{s,a}\right)^2 \mid s,a\right]\phi_{s,a}\phi_{s,a}^\top\right].
\end{aligned}
$$

Thus, it suffices to find $\sigma > 0$ such that

$$
\mathbb{E}\left[\left(\widehat{Q}_{s,a}(\theta) - w_\star^\top \phi_{s,a}\right)^2 \mid s,a\right] = \mathbb{E}\left[\left(\widehat{Q}_{s,a}(\theta)\right)^2 \mid s,a\right] - 2Q_{s,a}(\theta) w_\star^\top \phi_{s,a} + \left(w_\star^\top \phi_{s,a}\right)^2 \le \sigma^2 \tag{D.40}
$$

for all $(s,a) \in \mathcal{S}\times\mathcal{A}$ to verify (vi). Besides, we know that

$$
\mathbb{E}\left[\left(\widehat{Q}_{s,a}(\theta)\right)^2 \mid s,a\right] \overset{(D.38)}{\le} \frac{2}{(1-\gamma)^2}.
$$

We also know that $|Q_{s,a}(\theta)| \le \frac{1}{1-\gamma}$ and $\|\phi_{s,a}\| \le B$. Now we need to bound $\|w_\star\|$. Again, since $w_\star$ is the exact minimizer, we have $\nabla_w L_Q(w_\star, \theta, \tilde{d}^\theta) = 0$. That is

$$
\mathbb{E}_{(s,a)\sim\tilde{d}^\theta}\left[\left(w_\star^\top \phi_{s,a} - Q_{s,a}(\theta)\right)\phi_{s,a}\right] = 0,
$$

which implies

$$
\begin{aligned}
w_\star &= \left(\mathbb{E}_{(s,a)\sim\tilde{d}^\theta}\left[\phi_{s,a}\phi_{s,a}^\top\right]\right)^\dagger \mathbb{E}_{(s,a)\sim\tilde{d}^\theta}\left[Q_{s,a}(\theta)\phi_{s,a}\right] \\
&\overset{(5.6)}{\le} \frac{1}{1-\gamma}\left(\mathbb{E}_{(s,a)\sim\nu}\left[\phi_{s,a}\phi_{s,a}^\top\right]\right)^\dagger \mathbb{E}_{(s,a)\sim\tilde{d}^\theta}\left[Q_{s,a}(\theta)\phi_{s,a}\right].
\end{aligned}
$$

By the boundness of the feature map $\|\phi_{s,a}\| \le B$ and the Q-function $|Q_{s,a}(\theta)| \le \frac{1}{1-\gamma}$, and the condition (5.32), we have the minimizer $w_\star$ bounded by

$$
\|w_\star\| \overset{(5.32)}{\le} \frac{B}{\mu(1-\gamma)^2}.
$$

By using the upper bounds of $\mathbb{E}\left[\left(\widehat{Q}_{s,a}(\theta)\right)^2 \mid s,a\right]$, $|Q_{s,a}(\theta)|$, $\|w_\star\|$ and $\|\phi_{s,a}\|$, the left hand side of (D.40) can be upper bounded by

$$
\begin{aligned}
\mathbb{E}\left[\left(\widehat{Q}_{s,a}(\theta) - w_\star^\top \phi_{s,a}\right)^2 \mid s,a\right] &\le \frac{2}{(1-\gamma)^2} + \frac{2B^2}{\mu(1-\gamma)^3} + \frac{B^4}{\mu^2(1-\gamma)^4} \\
&= \frac{1}{(1-\gamma)^2}\left(\left(\frac{B^2}{\mu(1-\gamma)} + 1\right)^2 + 1\right) \\
&\le \frac{2}{(1-\gamma)^2}\left(\frac{B^2}{\mu(1-\gamma)} + 1\right)^2.
\end{aligned}
$$

Thus, in order to satisfy (D.40), we choose

$$\sigma = \frac{\sqrt{2}}{1-\gamma}\left(\frac{B^2}{\mu(1-\gamma)}+1\right).$$

Now all the conditions (i) - (vi) in Theorem D.15 are verified. With step size $\alpha = \frac{1}{2B^2}$, the initialization $w_0 = 0$ and $T$ steps of `Q-NPG-SGD` updates (D.12), we have

$$\begin{aligned}\mathbb{E}\left[L_Q(w_{\text{out}}, \theta, \tilde{d}^\theta)\right] - L_Q(w_\star, \theta, \tilde{d}^\theta) &\le \frac{4}{T}\left(\sigma\sqrt{m} + B\left\|w_\star\right\|\right)^2 \\ &\le \frac{4}{T}\left(\frac{\sqrt{2m}}{1-\gamma}\left(\frac{B^2}{\mu(1-\gamma)}+1\right) + \frac{B^2}{\mu(1-\gamma)^2}\right)^2\end{aligned}$$

Consequently, Assumption 5.1 is verified by

$$\sqrt{\epsilon_{\text{stat}}} \le \frac{2}{(1-\gamma)\sqrt{T}}\left(\frac{B^2}{\mu(1-\gamma)}\left(\sqrt{2m}+1\right) + \sqrt{2m}\right).$$

The proof is completed by replacing the above upper bound of $\sqrt{\epsilon_{\text{stat}}}$ in the results of Theorem 5.9. □

## D.5 Proof of Section 5.5

### D.5.1 The one step NPG lemma

To prove Theorem 5.15 and 5.16, we start from providing the one step analysis of the NPG update.

**Lemma D.10** (One step NPG lemma)**.** *Fix a state distribution $\rho$; an initial state-action distribution $\nu$; an arbitrary comparator policy $\pi^*$. At the $k$-th iteration, let $w_\star^{(k)} \in \operatorname{argmin}_w L_A(w, \theta^{(k)}, \tilde{d}^{(k)})$ denote the exact minimizer. Consider the $w^{(k)}$ and $\pi^{(k)}$ NPG iterates given in (5.33) and (5.18) respectively. Note*

$$\epsilon_{\text{stat}}^{(k)} \overset{def}{=} L_A(w^{(k)}, \theta^{(k)}, \tilde{d}^{(k)}) - L_A(w_\star^{(k)}, \theta^{(k)}, \tilde{d}^{(k)}), \tag{D.41}$$

$$\epsilon_{\text{approx}}^{(k)} \overset{def}{=} L_A(w_\star^{(k)}, \theta^{(k)}, \tilde{d}^{(k)}), \tag{D.42}$$

$$\delta_k \overset{def}{=} V_\rho^{(k)} - V_\rho(\pi^*).$$

*If Assumptions 5.12, 5.13 and 5.14 hold for all $k \geq 0$, then we have that*

$$\vartheta_\rho \left(\delta_{k+1} - \delta_k\right) + \delta_k \leq \frac{D_k^*}{(1-\gamma)\eta_k} - \frac{D_{k+1}^*}{(1-\gamma)\eta_k} + \frac{\sqrt{C_\nu}\left(\vartheta_\rho + 1\right)}{1-\gamma}\left(\sqrt{\epsilon_{\text{stat}}^{(k)}} + \sqrt{\epsilon_{\text{approx}}^{(k)}}\right). \tag{D.43}$$

*Proof.* As discussed in Section 5.3.1 and from Lemma D.3, we know that the corresponding update from $\pi^{(k)}$ to $\pi^{(k+1)}$ can be described by the PMD method (5.18). From the three-point descent lemma (Lemma D.14) and (5.18), we obtain that for any $p \in \Delta(\mathcal{A})$, we have

$$\eta_k \left\langle \bar{\Phi}_s^{(k)} w^{(k)}, \pi_s^{(k+1)} \right\rangle + D(\pi_s^{(k+1)}, \pi_s^{(k)}) \leq \eta_k \left\langle \bar{\Phi}_s^{(k)} w^{(k)}, p \right\rangle + D(p, \pi_s^{(k)}) - D(p, \pi_s^{(k+1)}).$$

Rearranging terms and dividing both sides by $\eta_k$, we get

$$\left\langle \bar{\Phi}_s^{(k)} w^{(k)}, \pi_s^{(k+1)} - p \right\rangle + \frac{1}{\eta_k} D(\pi_s^{(k+1)}, \pi_s^{(k)}) \leq \frac{1}{\eta_k} D(p, \pi_s^{(k)}) - \frac{1}{\eta_k} D(p, \pi_s^{(k+1)}).$$

Letting $p = \pi_s^{(k)}$ and knowing that

$$\left\langle \bar{\Phi}_s^{(k)} w^{(k)}, \pi_s^{(k)} \right\rangle = 0 \qquad \text{for all } k \geq 0,$$

which is due to (5.13), we have

$$\left\langle \bar{\Phi}_s^{(k)} w^{(k)}, \pi_s^{(k+1)} \right\rangle \leq -\frac{1}{\eta_k} D(\pi_s^{(k+1)}, \pi_s^{(k)}) - \frac{1}{\eta_k} D(\pi_s^{(k)}, \pi_s^{(k+1)}) \leq 0. \tag{D.44}$$

Letting $p = \pi_s^*$ yields

$$\left\langle \bar{\Phi}_s^{(k)} w^{(k)}, \pi_s^{(k+1)} - \pi_s^* \right\rangle \leq \frac{1}{\eta_k} D(\pi_s^*, \pi_s^{(k)}) - \frac{1}{\eta_k} D(\pi_s^*, \pi_s^{(k+1)}).$$

Note that we dropped the nonnegative term $\frac{1}{\eta_k} D(\pi_s^{(k+1)}, \pi_s^{(k)})$ on the left hand side to the inequality.

Taking expectation with respect to the distribution $d^*$, we have

$$\mathbb{E}_{s\sim d^*}\left[\left\langle \bar{\Phi}_s^{(k)} w^{(k)}, \pi_s^{(k+1)} \right\rangle\right] - \mathbb{E}_{s\sim d^*}\left[\left\langle \bar{\Phi}_s^{(k)} w^{(k)}, \pi_s^{*} \right\rangle\right] \leq \frac{1}{\eta_k} D_k^* - \frac{1}{\eta_k} D_{k+1}^*. \tag{D.45}$$

For the first expectation in (D.45), we have

$$\mathbb{E}_{s\sim d^*}\left[\left\langle \bar{\Phi}_s^{(k)} w^{(k)}, \pi_s^{(k+1)} \right\rangle\right]$$

$$
\begin{aligned}
&= \sum_{s\in\mathcal{S}} d_s^* \left\langle \bar{\Phi}_s^{(k)} w^{(k)}, \pi_s^{(k+1)} \right\rangle \\
&= \sum_{s\in\mathcal{S}} \frac{d_s^*}{d_s^{(k+1)}} d_s^{(k+1)} \left\langle \bar{\Phi}_s^{(k)} w^{(k)}, \pi_s^{(k+1)} \right\rangle \\
&\overset{(5.21)+(D.44)}{\geq} \vartheta_{k+1} \sum_{s\in\mathcal{S}} d_s^{(k+1)} \left\langle \bar{\Phi}_s^{(k)} w^{(k)}, \pi_s^{(k+1)} \right\rangle \\
&\overset{(5.21)+(D.44)}{\geq} \vartheta_{\rho} \sum_{s\in\mathcal{S}} d_s^{(k+1)} \left\langle \bar{\Phi}_s^{(k)} w^{(k)}, \pi_s^{(k+1)} \right\rangle \\
&= \vartheta_\rho \mathbb{E}_{(s,a)\sim\bar{d}^{(k+1)}} \left[ (\bar{\phi}_{s,a}^{(k)})^\top w^{(k)} \right] \\
&= \vartheta_\rho \mathbb{E}_{(s,a)\sim\bar{d}^{(k+1)}} \left[ A_{s,a}^{(k)} \right] + \vartheta_\rho \mathbb{E}_{(s,a)\sim\bar{d}^{(k+1)}} \left[ (\bar{\phi}_{s,a}^{(k)})^\top w^{(k)} - A_{s,a}^{(k)} \right] \\
&= \vartheta_\rho (1-\gamma) \left( V_\rho^{(k+1)} - V_\rho^{(k)} \right) + \vartheta_\rho \mathbb{E}_{(s,a)\sim\bar{d}^{(k+1)}} \left[ (\bar{\phi}_{s,a}^{(k)})^\top w^{(k)} - A_{s,a}^{(k)} \right], \quad \text{(D.46)}
\end{aligned}
$$

where the last line is obtained by the performance difference lemma (D.6), and we use the shorthand $\bar{\phi}_{s,a}^{(k)}$ as $\bar{\phi}_{s,a}(\theta^{(k)})$.

The second term of (D.46) can be lower bounded. To do it, we first decompose it into two terms. That is,

$$
\begin{aligned}
\mathbb{E}_{(s,a)\sim\bar{d}^{(k+1)}} \left[ (\bar{\phi}_{s,a}^{(k)})^\top w^{(k)} - A_{s,a}^{(k)} \right] = & \underbrace{\mathbb{E}_{(s,a)\sim\bar{d}^{(k+1)}} \left[ (\bar{\phi}_{s,a}^{(k)})^\top (w^{(k)} - w_\star^{(k)}) \right]}_{①} \\
& + \underbrace{\mathbb{E}_{(s,a)\sim\bar{d}^{(k+1)}} \left[ (\bar{\phi}_{s,a}^{(k)})^\top w_\star^{(k)} - A_{s,a}^{(k)} \right]}_{②}. \quad \text{(D.47)}
\end{aligned}
$$

We will upper bound the absolute values of the above two terms $|①|$ and $|②|$ separately. More precisely, similar to the proof of Theorem 5.9, we will upper bound the first term $|①|$ by the statistical error assumption (5.34) and upper bound the second term $|②|$ by using the approximation error assumption (5.35).

To upper bound ①, we first define the following covariance matrix of the centered feature map

$$
\Sigma_{\tilde{d}^{(k)}}^{(k)} \overset{\text{def}}{=} \mathbb{E}_{(s,a)\sim\tilde{d}^{(k)}} \left[ \bar{\phi}_{s,a}^{(k)} (\bar{\phi}_{s,a}^{(k)})^\top \right]. \quad \text{(D.48)}
$$

Here we use the superscript $(k)$ for $\Sigma_{\tilde{d}^{(k)}}^{(k)}$ to distinguish the covariance matrix of the feature map $\Sigma_{\tilde{d}^{(k)}}$ defined in (D.24) in the proof of Theorem 5.5, as the centered feature map $\bar{\phi}_{s,a}^{(k)}$ depends on the iterates $\theta^{(k)}$.

By Cauchy-Schwartz's inequality, we have

$$\begin{aligned}
|①| &\leq \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \bar{d}_{s,a}^{(k+1)} \left|(\bar{\phi}_{s,a}^{(k)})^\top (w^{(k)} - w_\star^{(k)})\right| \\
&\leq \sqrt{\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \frac{\left(\bar{d}_{s,a}^{(k+1)}\right)^2}{\tilde{d}_{s,a}^{(k)}} \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \tilde{d}_{s,a}^{(k)} \left((\bar{\phi}_{s,a}^{(k)})^\top (w^{(k)} - w_\star^{(k)})\right)^2} \\
&\overset{(D.48)}{=} \sqrt{\mathbb{E}_{(s,a)\sim\tilde{d}^{(k)}} \left[\left(\frac{\bar{d}_{s,a}^{(k+1)}}{\tilde{d}_{s,a}^{(k)}}\right)^2\right] \left\|w^{(k)} - w_\star^{(k)}\right\|_{\Sigma_{\tilde{d}^{(k)}}^{(k)}}^2}.
\end{aligned}$$

By further using the concentrability assumption 5.14, we have

$$\begin{aligned}
|①| &\overset{(5.36)}{\leq} \sqrt{C_\nu \left\|w^{(k)} - w_\star^{(k)}\right\|_{\Sigma_{\tilde{d}^{(k)}}^{(k)}}^2} \\
&\leq \sqrt{C_\nu \left(L_A(w^{(k)}, \theta^{(k)}, \tilde{d}^{(k)}) - L_A(w_\star^{(k)}, \theta^{(k)}, \tilde{d}^{(k)})\right)} \qquad \text{(D.49)} \\
&\overset{(D.41)}{=} \sqrt{C_\nu \epsilon_{\text{stat}}^{(k)}}, \qquad \text{(D.50)}
\end{aligned}$$

where (D.49) uses that $w_\star^{(k)}$ is a minimizer of $L_A$ and $w_\star^{(k)}$ is feasible (see the same arguments of (D.26) in the proof of Theorem 5.5).

For the second term |②| in (D.47), by Cauchy-Schwartz's inequality, we have

$$\begin{aligned}
|②| &\leq \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \bar{d}_{s,a}^{(k+1)} \left|(\bar{\phi}_{s,a}^{(k)})^\top w_\star^{(k)} - A_{s,a}^{(k)}\right| \\
&\leq \sqrt{\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \frac{\left(\bar{d}_{s,a}^{(k+1)}\right)^2}{\tilde{d}_{s,a}^{(k)}} \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \tilde{d}_{s,a}^{(k)} \left((\bar{\phi}_{s,a}^{(k)})^\top w_\star^{(k)} - A_{s,a}^{(k)}\right)^2} \\
&= \sqrt{\mathbb{E}_{(s,a)\sim\tilde{d}^{(k)}} \left[\left(\frac{\bar{d}_{s,a}^{(k+1)}}{\tilde{d}_{s,a}^{(k)}}\right)^2\right] L_A(w_\star^{(k)}, \theta^{(k)}, \tilde{d}^{(k)})} \\
&\overset{(5.36)+(D.42)}{\leq} \sqrt{C_\nu \epsilon_{\text{approx}}^{(k)}}. \qquad \text{(D.51)}
\end{aligned}$$

Plugging (D.50) and (D.51) into (D.46) yields

$$\mathbb{E}_{s\sim d^*}\left[\left\langle \bar{\Phi}_s^{(k)} w^{(k)}, \pi_s^{(k+1)} \right\rangle\right] \geq \vartheta_\rho (1-\gamma)\left(V_\rho^{(k+1)} - V_\rho^{(k)}\right) - \vartheta_\rho \sqrt{C_\nu}\left(\sqrt{\epsilon_{\text{stat}}^{(k)}} + \sqrt{\epsilon_{\text{approx}}^{(k)}}\right). \qquad \text{(D.52)}$$

Now for the second expectation in (D.45), by using the performance difference lemma (D.6) in Lemma D.4, we have

$$\begin{aligned}-\mathbb{E}_{s\sim d^*}\left[\left\langle \bar{\Phi}_s^{(k)} w^{(k)}, \pi_s^* \right\rangle\right] &= -\mathbb{E}_{(s,a)\sim \bar{d}^{\pi^*}}\left[A_{s,a}^{(k)}\right] + \mathbb{E}_{(s,a)\sim \bar{d}^{\pi^*}}\left[A_{s,a}^{(k)} - (\bar{\phi}_{s,a}^{(k)})^\top w^{(k)}\right] \\ &= (1-\gamma)\left(V_\rho^{(k)} - V_\rho(\pi^*)\right) + \mathbb{E}_{(s,a)\sim \bar{d}^{\pi^*}}\left[A_{s,a}^{(k)} - (\bar{\phi}_{s,a}^{(k)})^\top w^{(k)}\right]. \quad \text{(D.53)}\end{aligned}$$

The second term of (D.53) can be lower bounded. We first decompose it into two terms. That is,

$$\begin{aligned}\mathbb{E}_{(s,a)\sim \bar{d}^{\pi^*}}\left[A_{s,a}^{(k)} - (\bar{\phi}_{s,a}^{(k)})^\top w^{(k)}\right] &= \underbrace{\mathbb{E}_{(s,a)\sim \bar{d}^{\pi^*}}\left[A_{s,a}^{(k)} - (\bar{\phi}_{s,a}^{(k)})^\top w_\star^{(k)}\right]}_{\text{ⓐ}} \\ &\quad + \underbrace{\mathbb{E}_{(s,a)\sim \bar{d}^{\pi^*}}\left[(\bar{\phi}_{s,a}^{(k)})^\top (w_\star^{(k)} - w^{(k)})\right]}_{\text{ⓑ}}. \quad \text{(D.54)}\end{aligned}$$

Now we will upper bound the absolute values of the above two terms |ⓐ| and |ⓑ| separately.

For the first one |ⓐ|, by Cauchy-Schwartz's inequality, we have

$$\begin{aligned}|\text{ⓐ}| &\le \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \bar{d}_{s,a}^{\pi^*}\left|A_{s,a}^{(k)} - (\bar{\phi}_{s,a}^{(k)})^\top w_\star^{(k)}\right| \\ &\le \sqrt{\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \frac{\left(\bar{d}_{s,a}^{\pi^*}\right)^2}{\tilde{d}_{s,a}^{(k)}} \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \tilde{d}_{s,a}^{(k)}\left((\bar{\phi}_{s,a}^{(k)})^\top w_\star^{(k)} - A_{s,a}^{(k)}\right)^2} \\ &= \sqrt{\mathbb{E}_{(s,a)\sim\tilde{d}^{(k)}}\left[\left(\frac{\bar{d}_{s,a}^{\pi^*}}{\tilde{d}_{s,a}^{(k)}}\right)^2\right] L_A(w_\star^{(k)}, \theta^{(k)}, \tilde{d}^{(k)})} \\ &\overset{(5.36)+(D.42)}{\le} \sqrt{C_\nu \epsilon_{\text{approx}}^{(k)}}. \quad \text{(D.55)}\end{aligned}$$

For the second term |ⓑ| in (D.54), by Cauchy-Schwartz's inequality, we have

$$\begin{aligned}|\text{ⓑ}| &\le \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \bar{d}_{s,a}^{\pi^*}\left|(\bar{\phi}_{s,a}^{(k)})^\top (w_\star^{(k)} - w^{(k)})\right| \\ &\le \sqrt{\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \frac{\left(\bar{d}_{s,a}^{\pi^*}\right)^2}{\tilde{d}_{s,a}^{(k)}} \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \tilde{d}_{s,a}^{(k)}\left((\bar{\phi}_{s,a}^{(k)})^\top (w^{(k)} - w_\star^{(k)})\right)^2} \\ &\overset{(D.48)}{=} \sqrt{\mathbb{E}_{(s,a)\sim\tilde{d}^{(k)}}\left[\left(\frac{\bar{d}_{s,a}^{\pi^*}}{\tilde{d}_{s,a}^{(k)}}\right)^2\right] \left\|w^{(k)} - w_\star^{(k)}\right\|^2_{\Sigma_{\tilde{d}^{(k)}}^{(k)}}}\end{aligned}$$

$$\begin{aligned}
&\overset{(5.36)}{\leq} && \sqrt{C_\nu \left\| w^{(k)} - w_\star^{(k)} \right\|^2_{\Sigma^{(k)}_{\tilde{d}^{(k)}}}} \\
&\overset{(D.49)}{\leq} && \sqrt{C_\nu \left( L_A(w^{(k)}, \theta^{(k)}, \tilde{d}^{(k)}) - L_A(w_\star^{(k)}, \theta^{(k)}, \tilde{d}^{(k)}) \right)} \\
&\overset{(D.41)}{=} && \sqrt{C_\nu \epsilon^{(k)}_{\text{stat}}}.
\end{aligned} \tag{D.56}$$

Thus, we lower bound (D.54) by

$$-\mathbb{E}_{s\sim d^*}\left[\left\langle \bar{\Phi}_s^{(k)} w^{(k)}, \pi_s^* \right\rangle\right] \overset{(D.55)+(D.56)}{\geq} (1-\gamma)\left(V_\rho^{(k)} - V_\rho(\pi^*)\right) - \sqrt{C_\nu}\left(\sqrt{\epsilon^{(k)}_{\text{stat}}} + \sqrt{\epsilon^{(k)}_{\text{approx}}}\right). \tag{D.57}$$

Substituting (D.52) and (D.57) into (D.45), dividing both side by $1-\gamma$ and rearranging terms, we get

$$\vartheta_\rho\left(\delta_{k+1} - \delta_k\right) + \delta_k \leq \frac{D_k^*}{(1-\gamma)\eta_k} - \frac{D_{k+1}^*}{(1-\gamma)\eta_k} + \frac{\sqrt{C_\nu}\left(\vartheta_\rho + 1\right)}{1-\gamma}\left(\sqrt{\epsilon^{(k)}_{\text{stat}}} + \sqrt{\epsilon^{(k)}_{\text{approx}}}\right).$$

□

### D.5.2 Proof of Theorem 5.15

*Proof.* From (D.43) in Lemma D.10, by using the same increasing step size as in Theorem 5.5, i.e. $\eta_0 \geq \frac{1-\gamma}{\gamma} D_0^*$ and $\eta_{k+1} \geq \eta_k/\gamma$, and following the same arguments in the proof of Theorem 5.5 after (D.37), we obtain the final performance bound with the linear convergence rate

$$\mathbb{E}\left[V_\rho(\pi^{(k)})\right] - V_\rho(\pi^*) \leq \left(1 - \frac{1}{\vartheta_\rho}\right)^k \frac{2}{1-\gamma} + \frac{\sqrt{C_\nu}\left(\vartheta_\rho + 1\right)}{1-\gamma}\left(\sqrt{\epsilon_{\text{stat}}} + \sqrt{\epsilon_{\text{approx}}}\right).$$

□

### D.5.3 Proof of Theorem 5.16

*Proof.* From (D.43) in Lemma D.10 with the constant step size, we have

$$\vartheta_\rho\left(\delta_{k+1} - \delta_k\right) + \delta_k \leq \frac{D_k^*}{(1-\gamma)\eta} - \frac{D_{k+1}^*}{(1-\gamma)\eta} + \frac{\sqrt{C_\nu}\left(\vartheta_\rho + 1\right)}{1-\gamma}\left(\sqrt{\epsilon^{(k)}_{\text{stat}}} + \sqrt{\epsilon^{(k)}_{\text{approx}}}\right).$$

Taking the total expectation with respect to the randomness in the sequence of the iterates $w^{(0)}, \cdots, w^{(k-1)}$ yields

$$\begin{aligned}
\vartheta_\rho \left(\mathbb{E}\left[\delta_{k+1}\right] - \mathbb{E}\left[\delta_k\right]\right) + \mathbb{E}\left[\delta_k\right] \quad &\leq \quad \frac{\mathbb{E}\left[D_k^*\right]}{(1-\gamma)\eta} - \frac{\mathbb{E}\left[D_{k+1}^*\right]}{(1-\gamma)\eta} \\
&\quad + \frac{\sqrt{C_\nu}\left(\vartheta_\rho + 1\right)}{1-\gamma}\left(\mathbb{E}\left[\sqrt{\epsilon_{\text{stat}}^{(k)}}\right] + \mathbb{E}\left[\sqrt{\epsilon_{\text{approx}}^{(k)}}\right]\right) \\
&\leq \quad \frac{\mathbb{E}\left[D_k^*\right]}{(1-\gamma)\eta} - \frac{\mathbb{E}\left[D_{k+1}^*\right]}{(1-\gamma)\eta} \\
&\quad + \frac{\sqrt{C_\nu}\left(\vartheta_\rho + 1\right)}{1-\gamma}\left(\sqrt{\mathbb{E}\left[\epsilon_{\text{stat}}^{(k)}\right]} + \sqrt{\mathbb{E}\left[\epsilon_{\text{approx}}^{(k)}\right]}\right) \\
&\overset{(5.34)+(5.35)}{\leq} \quad \frac{\mathbb{E}\left[D_k^*\right]}{(1-\gamma)\eta} - \frac{\mathbb{E}\left[D_{k+1}^*\right]}{(1-\gamma)\eta} \\
&\quad + \frac{\sqrt{C_\nu}\left(\vartheta_\rho + 1\right)}{1-\gamma}\left(\sqrt{\epsilon_{\text{stat}}} + \sqrt{\epsilon_{\text{approx}}}\right).
\end{aligned}$$

By summing up from $0$ to $k-1$, we get

$$\vartheta_\rho \mathbb{E}\left[\delta_k\right] + \sum_{t=0}^{k-1} \mathbb{E}\left[\delta_t\right] \leq \frac{D_0^*}{(1-\gamma)\eta} + \vartheta_\rho \delta_0 + k \cdot \frac{\sqrt{C_\nu}\left(\vartheta_\rho + 1\right)}{1-\gamma}\left(\sqrt{\epsilon_{\text{stat}}} + \sqrt{\epsilon_{\text{approx}}}\right).$$

Finally, dropping the positive term $\mathbb{E}\left[\delta_k\right]$ on the left hand side as $\pi^*$ is the optimal policy and dividing both side by $k$ yields

$$\frac{1}{k}\sum_{t=0}^{k-1} \mathbb{E}\left[V_\rho(\pi^{(t)})\right] - V_\rho(\pi^*) \leq \frac{D_0^*}{(1-\gamma)\eta k} + \frac{2\vartheta_\rho}{(1-\gamma)k} + \frac{\sqrt{C_\nu}\left(\vartheta_\rho + 1\right)}{1-\gamma}\left(\sqrt{\epsilon_{\text{stat}}} + \sqrt{\epsilon_{\text{approx}}}\right).$$

□

### D.5.4 Proof of Corollary 5.17

There is a similar remark for the proof of Corollary 5.17 to the one right before the proof of Corollary 5.11 in Appendix D.4.5. We notice that there is the same error occurred for the proof of NPG sample complexity analysis in Agarwal et al. (2021). Recall the stochastic gradient of $L_A$ in (D.9)

$$\widehat{\nabla}_w L_A(w, \theta, \tilde{d}^\theta) = 2\left(w^\top \bar{\phi}_{s,a}(\theta) - \widehat{A}_{s,a}(\theta)\right)\bar{\phi}_{s,a}(\theta).$$

It turns out that $\widehat{\nabla}_w L_A(w, \theta, \tilde{d}^\theta)$ is unbounded, since the estimate $\widehat{A}_{s,a}(\theta)$ of $A_{s,a}(\theta)$ can be unbounded due to the unbounded length of the trajectory sampled in the sampling procedure, Algorithm 14. Thus, Agarwal et al. (2021) incorrectly verify $\widehat{\nabla} L_A(w, \theta, \tilde{d}^\theta)$ bounded by claiming that $\widehat{A}_{s,a}(\theta)$ is bounded by $\frac{2}{1-\gamma}$.

**Proof sketch.** Despite the difference of using either $\tilde{d}^\theta$ or $\bar{d}^\theta$ in the loss function $L_A$, we use the same assumptions of Liu et al. (2020), i.e., the Fisher-non-degeneracy (5.37) and the boundedness of the feature map, and verify all the conditions of Theorem D.15 *without* relying on the boundedness of the stochastic gradient. In particular, similar to the proof of Corollary 5.11, we verify that $\mathbb{E}[\widehat{A}_{s,a}(\theta)^2]$ is bounded even though $\widehat{A}_{s,a}(\theta)$ is unbounded. To verify the condition (vi) in Theorem D.15 in our proof, we use the same conditional expectation computation trick as in the proof of Corollary 5.11 to separate the correlated random variables $\widehat{A}_{s,a}(\theta)$ and $\bar{\phi}_{s,a}(\theta)$ with $(s,a) \sim \tilde{d}^\theta$ appeared in the stochastic gradient. Thanks to this trick, we fix a flaw in the previous proof of Liu et al. (2020, Proposition G.1) [2].

*Proof.* Similar to the proof of Corollary 5.11, we suppress the subscript $k$. First, the centered feature map is bounded by $\left\|\bar{\phi}_{s,a}(\theta)\right\| \le 2B$. In order to apply Theorem D.15, it remains to upper bound $\mathbb{E}[\|\widehat{A}_{s,a}(\theta)\bar{\phi}_{s,a}(\theta)\|^2]$ and $\|w_\star\|$ with $w_\star \in \operatorname{argmin}_w L_A(w,\theta,\tilde{d}^\theta)$, and find $\sigma > 0$ such that

$$\mathbb{E}\left[\left(\widehat{A}_{s,a}(\theta) - w_\star^\top \bar{\phi}_{s,a}(\theta)\right)^2 \mid s,a\right] = \mathbb{E}\left[\left(\widehat{A}_{s,a}(\theta)\right)^2 \mid s,a\right] - 2A_{s,a}(\theta) w_\star^\top \bar{\phi}_{s,a}(\theta) + \left(w_\star^\top \bar{\phi}_{s,a}(\theta)\right)^2 \le \sigma^2 \tag{D.58}$$

holds for all $(s,a) \in \mathcal{S} \times \mathcal{A}$ and $\theta \in \mathbb{R}^m$.

Similar to the proof of Corollary 5.11, the closed form solution of $w_\star$ can be written as

$$w_\star = \left(\mathbb{E}_{(s,a)\sim\tilde{d}^\theta}\left[\bar{\phi}_{s,a}(\theta)\bar{\phi}_{s,a}(\theta)^\top\right]\right)^\dagger \mathbb{E}_{(s,a)\sim\tilde{d}^\theta}\left[Q_{s,a}(\theta)\bar{\phi}_{s,a}(\theta)\right].$$

From (5.37), we have

$$\|w_\star\| \le \frac{2B}{\mu(1-\gamma)}.$$

[2] In a previous version of the proof in Section G, Liu et al. (2020, Proposition G.1) use the inequality

$$\mathbb{E}\left[\left(\widehat{A}_{s,a}(\theta) - w_\star^\top \bar{\phi}_{s,a}(\theta)\right)^2 \bar{\phi}_{s,a}(\theta)\left(\bar{\phi}_{s,a}(\theta)\right)^\top\right] \le \mathbb{E}\left[\left(\widehat{A}_{s,a}(\theta) - w_\star^\top \bar{\phi}_{s,a}(\theta)\right)^2\right] \mathbb{E}\left[\bar{\phi}_{s,a}(\theta)\left(\bar{\phi}_{s,a}(\theta)\right)^\top\right]$$

which is incorrect since $\widehat{A}_{s,a}(\theta)$ and $\bar{\phi}_{s,a}(\theta)$ are correlated random variables. To fix it, we use the following conditional expectation computation trick

$$\mathbb{E}\left[\left(\widehat{A}_{s,a}(\theta) - w_\star^\top \bar{\phi}_{s,a}(\theta)\right)^2 \bar{\phi}_{s,a}(\theta)\left(\bar{\phi}_{s,a}(\theta)\right)^\top\right] = \mathbb{E}\left[\mathbb{E}\left[\left(\widehat{A}_{s,a}(\theta) - w_\star^\top \bar{\phi}_{s,a}(\theta)\right)^2 \mid s,a\right] \bar{\phi}_{s,a}(\theta)\left(\bar{\phi}_{s,a}(\theta)\right)^\top\right],$$

and bound the term $\mathbb{E}\left[\left(\widehat{A}_{s,a}(\theta) - w_\star^\top \bar{\phi}_{s,a}(\theta)\right)^2 \mid s,a\right]$ in (D.58). This error is recently fixed by Liu et al. (2020) on https://arxiv.org/pdf/2211.07937.pdf in their original paper.

Now we need to upper bound $\mathbb{E}\left[\left(\widehat{A}_{s,a}(\theta)\right)^2 \mid s,a\right]$ from (D.58). Indeed, by using $\widehat{A}_{s,a}(\theta) = \widehat{Q}_{s,a}(\theta) - \widehat{V}_s(\theta)$, we have

$$\begin{aligned}\mathbb{E}\left[\left(\widehat{A}_{s,a}(\theta)\right)^2 \mid s,a\right] &\leq 2\mathbb{E}\left[\left(\widehat{Q}_{s,a}(\theta)\right)^2 \mid s,a\right] + 2\mathbb{E}\left[\left(\widehat{V}_{s,a}(\theta)\right)^2 \mid s,a\right] \\ &\overset{\text{(D.38)}}{\leq} \frac{8}{(1-\gamma)^2},\end{aligned} \tag{D.59}$$

where the last line is obtained, as $\mathbb{E}\left[\left(\widehat{V}_{s,a}(\theta)\right)^2 \mid s,a\right]$ shares the same upper bound (D.38) of $\mathbb{E}\left[\left(\widehat{Q}_{s,a}(\theta)\right)^2 \mid s,a\right]$ by using the similar argument.

From (D.59) and $\bar{\phi}_{s,a}(\theta) \leq 2B$, we verify $\mathbb{E}\left[\left\|\widehat{A}_{s,a}(\theta)\bar{\phi}_{s,a}(\theta)\right\|^2\right]$ bounded as well.

By using the upper bounds of $\mathbb{E}\left[\left(\widehat{A}_{s,a}(\theta)\right)^2 \mid s,a\right]$, $\|w_\star\|$, $|A_{s,a}(\theta)| \leq \frac{2}{1-\gamma}$ and $\left\|\bar{\phi}_{s,a}(\theta)\right\| \leq 2B$, the left hand side of (D.58) is upper bounded by

$$\begin{aligned}\mathbb{E}\left[\left(\widehat{A}_{s,a}(\theta) - w_\star^\top \bar{\phi}_{s,a}(\theta)\right)^2 \mid s,a\right] &\leq \frac{8}{(1-\gamma)^2} + \frac{16B^2}{\mu(1-\gamma)^2} + \frac{16B^4}{\mu^2(1-\gamma)^2} \\ &= \frac{4}{(1-\gamma)^2}\left(\left(\frac{2B^2}{\mu}+1\right)^2 + 1\right) \\ &\leq \frac{8}{(1-\gamma)^2}\left(\frac{2B^2}{\mu}+1\right)^2.\end{aligned}$$

Thus, we choose

$$\sigma = \frac{2\sqrt{2}}{1-\gamma}\left(\frac{2B^2}{\mu}+1\right).$$

Now all the conditions (i) - (vi) in Theorem D.15 are verified. The reminder of the proof follows that of Corollary 5.11. □

## D.6 Discussion on the distribution mismatch coefficients and the concentrability coefficients

We have already mentioned in the comparison with Agarwal et al. (2021) right after Theorem 5.5 that, although we have linear convergence rates, the magnitude of our error floor is worse (larger) by a factor of $\vartheta_\rho\sqrt{C_\rho}$ ($\vartheta_\rho\sqrt{C_\nu}$ for Theorem 5.9 and 5.15), due to the concentrability $C_\rho$ and the distribution mismatch coefficients $\vartheta_\rho$ used in our proof. Such difference comes from different nature of the proof techniques. Here the distribution mismatch coefficients $\vartheta_\rho$ and

the concentrability coefficients $C_\rho$ and $C_\nu$ are potentially large in our convergence theories. We give extensive discussions on them, respectively.

### D.6.1 Distribution mismatch coefficients $\vartheta_\rho$

Our distribution mismatch coefficient $\vartheta_\rho$ in (5.21) is the same as the one in Xiao (2022). It contains both an upper bound and a lower bound. The linear convergence rate in our theories is $1 - \frac{1}{\vartheta_\rho} > 0$. Thus, the smaller $\vartheta_\rho$ is, the faster the resulting linear convergence rate. The best linear convergence rate is achieved when $\vartheta_\rho$ achieves its lower bound. Here our analysis is general that it includes all the distribution mismatch coefficient $\vartheta_\rho$ induced by any target state distribution $\rho$. Our results generalizes and sometimes also improves with respect to prior results.

A very pessimistic and trivial upper bound on $\vartheta_\rho$ is

$$\vartheta_\rho \leq \frac{1}{(1-\gamma)\rho_{\min}}.$$

However, if the target state distribution $\rho \in \Delta(\mathcal{S})$ does not have full support, i.e., $\rho_s = 0$ for some $s \in \mathcal{S}$, then $\vartheta_\rho$ might be infinite from this upper bound. Xiao (2022) just assumes that $\vartheta_\rho$ is finite. We further propose a solution to this particular issue. Indeed, if $\rho$ does not have full support, consider $\pi^*$ as an optimal policy. We can always convert the convergence guarantees for some state distribution $\rho' \in \Delta(\mathcal{S})$ with full support, i.e., $\rho'_s > 0$ for all $s \in \mathcal{S}$ as follows:

$$\begin{aligned} V_\rho(\pi^{(k)}) - V_\rho(\pi^*) &= \sum_{s\in\mathcal{S}} \rho_s \left(V_s(\pi^{(k)}) - V_s(\pi^*)\right) = \sum_{s\in\mathcal{S}} \frac{\rho_s}{\rho'_s}\rho'_s \left(V_s(\pi^{(k)}) - V_s(\pi^*)\right) \\ &\leq \left\|\frac{\rho}{\rho'}\right\|_\infty \sum_{s\in\mathcal{S}} \rho'_s \left(V_s(\pi^{(k)}) - V_s(\pi^*)\right) = \left\|\frac{\rho}{\rho'}\right\|_\infty \left(V_{\rho'}(\pi^{(k)}) - V_{\rho'}(\pi^*)\right). \end{aligned}$$

Then we only need convergence guarantees of $V_{\rho'}(\pi^{(k)}) - V_{\rho'}(\pi^*)$ for arbitrary $\rho'$ obtained from all our convergence analysis above. In this case, the linear convergence rate depends on

$$\vartheta_{\rho'} \overset{\text{def}}{=} \frac{1}{1-\gamma}\left\|\frac{d^{\pi^*}(\rho')}{\rho'}\right\|_\infty < \infty.$$

Equation (5.21) provides the lower bound $\frac{1}{1-\gamma}$ for $\vartheta_\rho$. Such lower bound can be achieved when the target state distribution $\rho$ satisfies that $\rho = d^{\pi^*}(\rho)$ where $\pi^*$ is an optimal policy. The advantage of this case is that, not only it implies the best linear convergence rate, more importantly, the fast linear convergence rate is known to be $\gamma$. So we know the convergence rate explicitly without any estimation, even though the optimal policy or the policy iterates are unknown before training. Hence, we know when to stop running the algorithm. Lan (2022)

only considers the case when $\rho = d^{\pi^*}(\rho)$ and we are able to recover the same linear convergence rate $\gamma$ in their result.

Furthermore, the convergence performance $V_\rho(\pi^{(k)}) - V_\rho(\pi^*)$ depends on the target state distribution $\rho$. If the optimal policy $\pi^*$ is independent to the target state distribution $\rho$ which is usually the case in RL problems, then we are always allowed to fix $\rho = d^{\pi^*}(\rho)$ for the analysis without knowing $\rho$ and $\pi^*$ and derive this best linear convergence performance with rate $\gamma$, because we use the initial state-action distribution $\nu$ in training which is independent to $\rho$.

Finally, from (5.21), if $d^{(k)}$ converges to $d^*$, then $\vartheta_k$ converges to 1. This might imply superlinear convergence results as Section 4.3 in Xiao (2022). In this case, the notion of the distribution mismatch coefficients $\vartheta_\rho$ no longer exists for the superlinear convergence analysis. In other words, it is no longer concerned.

### D.6.2 Concentrability coefficients $C_\nu$

The issue of having (potentially large) concentrability coefficients is unavoidable in all the fast linear convergence analysis of the inexact NPG that we are aware of, including even the tabular setting (e.g., Lan (2022) and Xiao (2022)) and the log-linear policy setting (Cayci et al. (2021), Chen and Theja Maguluri (2022) and ours).

First, in the fast linear convergence analysis of inexact NPG, the concentrability coefficients appear from the errors, including the statistical error and the approximation error. Thus, one way to avoid having the concentrability coefficients appear is to consider the exact NPG in the tabular setting (See Theorem 10 in Xiao (2022)). Because the tabular setting makes no approximation error and the exact NPG makes no statistical error. We consider the *inexact* NPG with the *log-linear* policy. Consequently, we have the concentrability coefficients multiplied by both the statistical error $\epsilon_{\text{stat}}$ and the approximation error ($\epsilon_{\text{bias}}$ in Assumption 5.2 or $\epsilon_{\text{approx}}$ in Assumption 5.7 and 5.13).

To remove the concentrability coefficients, one has to make strong assumptions on the errors with the $L_\infty$ supremum norm. In the tabular setting, Lan (2022) and Xiao (2022) assume that $\|\widehat{Q}(\pi) - Q(\pi)\|_\infty \leq \epsilon_{\text{stat}}$. The cons of such strong assumption requires high sample complexity and is already explained in Appendix D.1.1. In the log-linear policy setting, Chen and Theja Maguluri (2022) assume that $\|Q_s(\theta^{(k)}) - \Phi w_\star^{(k)}\|_\infty \leq \epsilon_{\text{bias}}$ for the approximation error, which is a very strong assumption in the function approximation regime. Due to the supremum norm, $\epsilon_{\text{bias}}$ is unlikely to be small, especially for large action spaces. Under this strong assumption, Lan (2022), Xiao (2022) and Chen and Theja Maguluri (2022) are able to eliminate the concentrability coefficients. To avoid assuming such strong assumptions, both Cayci et al. (2021) and our work consider the expected $L_2$ errors in the log-linear policy setting, which are much weaker assumptions, especially much more reasonable for the approximation

error $\epsilon_{\text{bias}}$ compared to the one in Chen and Theja Maguluri (2022). The tradeoff is that, the concentrability coefficients can not be eliminated in this case both in Cayci et al. (2021) and our results.

Furthermore, as mentioned right after Theorem 5.15, under the expected error assumptions (Assumption 5.12 and 5.13), our concentrability coefficient $C_\nu$ is better presented than the one in Assumption 2 in Cayci et al. (2021) in the sense that it is independent to the policies throughout the iterations thanks to the use of $\tilde{d}^{(k)}$ instead of $\bar{d}^{(k)}$ (which is mentioned in Remark 5.10 as well) and is controllable to be finite by $\nu$, while the one in Cayci et al. (2021) depends on the iterates, thus is unknown and is not guaranteed to be finite.

Finally, like the distribution mismatch coefficient, the upper bound of $C_\nu$ in (5.31) is very pessimistic. By the definition of $C_\nu$ in (5.29), one can expect that $C_\nu$ is closed to 1, when $\pi^{(k)}$ and $\pi^{(k+1)}$ converge to $\pi^*$ with $\pi^*$ the optimal policy.

So our concentrability coefficient $C_\nu$ is the "best" one among all concentrability coefficients in the sense that it takes the weakest assumptions on errors compared to Lan (2022) and Xiao (2022) and Chen and Theja Maguluri (2022), it does not impose any restrictions on the MDP dynamics compared to Cayci et al. (2021) and it can be controlled to be finite by $\nu$ when other concentrability coefficients are infinite (Scherrer, 2014).

It is still an open question whether we can obtain fast linear convergence results of the inexact NPG in the log-linear policy setting, with small error floor and a much improved concentrability coefficient, e.g., as the same magnitude as the one in Agarwal et al. (2021).

## D.7 Standard optimization results

In this section, we present the standard optimization results from Beck (2017), Xiao (2022), and Bach and Moulines (2013) used in our proofs.

First, we present the closed form update of mirror descent with KL divergence on the simplex. We provide its proof for the completeness.

**Lemma D.11** (Mirror descent on the simplex, Example 9.10 in Beck (2017)). *Let $g \in \mathbb{R}^n$ which will often be a gradient and let $\eta > 0$. For $p, q$ in the unit $n$-simplex $\Delta^n$, the mirror descent step with respect to the KL divergence*

$$\min_{p \in \Delta^n} \ \eta \langle g, p \rangle + D(p, q) \tag{D.60}$$

*is given by*

$$p = \frac{q \odot e^{-\eta g}}{\sum_{i=1}^n q_i e^{-\eta g_i}}, \tag{D.61}$$

*where $\odot$ is the element-wise product between vectors.*

*Proof.* The Lagrangian of (D.60) is given by

$$L(p, \mu, \lambda) = \eta \langle g, p \rangle + D(p, q) + \mu(1 - \sum_{i=1}^n p_i) - \sum_{i=1}^n \lambda_i p_i,$$

where $\mu \in \mathbb{R}$ and $\lambda \in \mathbb{R}^n$ with non-negative coordinates are the Lagrangian multipliers. Thus the Karush–Kuhn–Tucker conditions are given by

$$\begin{aligned} \eta g + \log(p/q) + \mathbf{1}_n &= \mu \mathbf{1}_n + \lambda, \\ \mathbf{1}_n^\top p &= 1, \\ \lambda_i = 0 \text{ or } p_i &= 0, \qquad \text{for all } i = 1, \cdots, n, \end{aligned}$$

where the division $p/q$ is element-wise. Isolating $p$ in the top equation gives

$$p = q \odot e^{(\mu-1)\mathbf{1}_n + \lambda - \eta g} = e^{\mu - 1} q \odot e^{\lambda - \eta g}.$$

Using the second constraint $\mathbf{1}_n^\top p = 1$ gives that

$$1 = e^{\mu-1} \sum_{i=1}^n q_i e^{\lambda_i - \eta g_i} \implies e^{\mu - 1} = \frac{1}{\sum_{i=1}^n q_i e^{\lambda_i - \eta g_i}}.$$

Consequently, by plugging the above term into $p$, we have that

$$p = \frac{q \odot e^{\lambda - \eta g}}{\sum_{i=1}^n q_i e^{\lambda_i - \eta g_i}}.$$

It remains to determine $\lambda$. If $q_i = 0$ then $p_i = 0$ and thus $\lambda_i > 0$. Conversely, if $q_i > 0$ then $p_i > 0$ and thus $\lambda_i = 0$. In either of these cases, we have that the solution is given by (D.61). □

Now we present the three-point descent lemma on proximal optimization with Bregman divergences, which is another key ingredient for our PMD analysis. Following Xiao (2022, Lemma 6), we adopt a slight variation of Lemma 3.2 in Chen and Teboulle (1993). First, we say a convex function $f$ is *proper* if $\operatorname{dom} f$ is nonempty and for all $x \in \operatorname{dom} f$, $f(x) > -\infty$; we say a convex function is *closed*, if it is lower semi-continuous. Before presenting the lemma, we still need some technical conditions.

**Definition D.12** (Legendre function, Section 26 in Rockafellar (1970)). *We say a function $h$ is of* Legendre type *or a* Legendre function *if the following properties are satisfied:*

(*i*) *$h$ is strictly convex in the relative interior of* $\operatorname{dom} h$*, denoted as* $\operatorname{rint}\operatorname{dom} h$.

(*ii*) *$h$ is essentially smooth, i.e., $h$ is differentiable in* $\operatorname{rint}\operatorname{dom} h$ *and, for any boundary point $x_b$ of* $\operatorname{rint}\operatorname{dom} h$, $\lim_{x\to x_b} \|\nabla h(x)\| \to \infty$ *where* $x \in \operatorname{rint}\operatorname{dom} h$.

**Definition D.13** (Bregman divergence (Bregman, 1967; Censor and Zenios, 1997)). *Let $h : \operatorname{dom} h \to \mathbb{R}$ be a Legendre function and assume that* $\operatorname{rint}\operatorname{dom} h$ *is nonempty. The* Bregman divergence *$D_h(\cdot,\cdot) : \operatorname{dom} h \times \operatorname{rint}\operatorname{dom} h \to [0,\infty)$ generated by $h$ is a distance-like function defined as*

$$D_h(p,p') \stackrel{def}{=} h(p) - h(p') - \langle \nabla h(p'), p - p' \rangle . \tag{D.62}$$

Under the above conditions, we have the following result. We also provide its proof for self-containment. (Xiao (2022) does not provide a formal proof.)

**Lemma D.14** (Three-point descent lemma, Lemma 6 in Xiao (2022)). *Suppose that $\mathcal{C} \subset \mathbb{R}^m$ is a closed convex set, $f : \mathcal{C} \to \mathbb{R}$ is a proper, closed convex function, $D_h(\cdot,\cdot)$ is the Bregman divergence generated by a function $h$ of Lengendre type and* $\operatorname{rint}\operatorname{dom} h \cap \mathcal{C} \neq \emptyset$*. For any $x \in \operatorname{rint}\operatorname{dom} h$, let*

$$x^+ \in \arg\min_{u \in \operatorname{dom} h \cap \mathcal{C}} \{f(u) + D_h(u,x)\}.$$

*Then $x^+ \in \operatorname{rint}\operatorname{dom} h \cap \mathcal{C}$ and for any $u \in \operatorname{dom} h \cap \mathcal{C}$,*

$$f(x^+) + D_h(x^+,x) \le f(u) + D_h(u,x) - D_h(u,x^+).$$

*Proof.* First, we prove that for any $a, b \in \operatorname{rint}\operatorname{dom} h$ and $c \in \operatorname{dom} h$, the following identity holds:

$$D_h(c,a) + D_h(a,b) - D_h(c,b) = \langle \nabla h(b) - \nabla h(a), c - a \rangle . \tag{D.63}$$

Indeed, using the definition of $D_h$ in (D.62), we have

$$\langle \nabla h(a), c - a \rangle = h(c) - h(a) - D_h(c,a), \tag{D.64}$$
$$\langle \nabla h(b), a - b \rangle = h(a) - h(b) - D_h(a,b), \tag{D.65}$$

$$\langle \nabla h(b), c - b \rangle = h(c) - h(b) - D_h(c, b). \tag{D.66}$$

Subtracting (D.64) and (D.65) from (D.66) yields (D.63).

Next, since $h$ is of Legendre type, we have $x^+ \in \operatorname{rint} \operatorname{dom} h \cap \mathcal{C}$. Otherwise, $x^+$ is a boundary point of $\operatorname{dom} h$. From the definition of Legendre function, $\|\nabla h(x^+)\| = \infty$ which is not possible, as $x^+$ is also the minimum point of $f(u) + D_h(u, x)$. By the first-order optimality condition, we have

$$\left\langle u - x^+, g^+ + \nabla_y D_h(y, x)|_{y=x^+} \right\rangle \geq 0,$$

where $g^+ \in \partial f(x^+)$ is the subdifferential of $f$ at $x^+$. From the definition of $D_h$, the above inequality is equivalent to

$$\left\langle u - x^+, \nabla h(x^+) - \nabla h(x) \right\rangle \geq \left\langle x^+ - u, g^+ \right\rangle. \tag{D.67}$$

Besides, plugging $c = u, a = x^+$ and $b = x$ into (D.63), we obtain

$$\left\langle u - x^+, \nabla h(x^+) - \nabla h(x) \right\rangle = D_h(u, x) - D_h(u, x^+) - D_h(x^+, x) \overset{\text{(D.67)}}{\geq} \left\langle x^+ - u, g^+ \right\rangle.$$

Rearranging terms and adding $f(u)$ on both sides, we have

$$\begin{aligned} D_h(u, x) - D_h(u, x^+) + f(u) &\geq f(u) + \left\langle x^+ - u, g^+ \right\rangle + D_h(x^+, x) \\ &\geq f(x^+) + D_h(x^+, x), \end{aligned}$$

which concludes the proof. The last inequality is obtained by the convexity of $f$ and $g^+ \in \partial f(x^+)$. □

Finally, we use the following linear regression analysis for the proof of our sample complexity results, i.e., Corollary 5.11 and 5.17.

**Theorem D.15** (Theorem 1 in Bach and Moulines (2013)). *Consider the following assumptions:*

(*i*) *$\mathcal{H}$ is a $m$-dimensional Euclidean space.*

(*ii*) *The observations $(x_n, z_n) \in \mathcal{H} \times \mathcal{H}$ are independent and identically distributed.*

(*iii*) *$\mathbb{E}\left[\|x_n\|^2\right]$ and $\mathbb{E}\left[\|z_n\|^2\right]$ are finite. The covariance $\mathbb{E}\left[x_n x_n^\top\right]$ is assumed invertible.*

(*iv*) *The global minimum of $f(\theta) = \frac{1}{2}\mathbb{E}\left[\langle \theta, x_n \rangle^2 - 2 \langle \theta, z_n \rangle\right]$ is attained at a certain $\theta_* \in \mathcal{H}$. Let $\xi_n = z_n - \langle \theta_*, x_n \rangle x_n$ denote the residual. We have $\mathbb{E}[\xi_n] = 0$.*

(*v*) *Consider the stochastic gradient recursion defined as*

$$\theta_n = \theta_{n-1} - \eta(\langle \theta_{n-1}, x_n \rangle \, x_n - z_n),$$

*started from* $\theta_0 \in \mathcal{H}$ *and also consider the averaged iterates* $\theta_{\text{out}} = \frac{1}{n+1} \sum_{k=0}^{n} \theta_k$.

(*vi*) *There exists* $R > 0$ *and* $\sigma > 0$ *such that* $\mathbb{E}\left[\xi_n \xi_n^\top\right] \leq \sigma^2 \mathbb{E}\left[x_n x_n^\top\right]$ *and* $\mathbb{E}\left[\|x_n\|^2 \, x_n x_n^\top\right] \leq R^2 \mathbb{E}\left[x_n x_n^\top\right]$.

*When* $\eta = \frac{1}{4R^2}$, *we have*

$$\mathbb{E}\left[f(\theta_{\text{out}}) - f(\theta_*)\right] \leq \frac{2}{n} \left(\sigma\sqrt{m} + R\,\|\theta_0 - \theta_*\|\right)^2. \tag{D.68}$$

# List of Figures

# List of Algorithms

# List of Tables

# List of References


Agarwal, Alekh, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan (2021). On the Theory of Policy Gradient Methods: Optimality, Approximation, and Distribution Shift. *Journal of Machine Learning Research* 22.98, pp. 1–76 (Cited on pages 8, 10, 67, 69, 74, 80–84, 87, 88, 90–92, 94–99, 101–104, 106, 123, 125, 213, 216, 249–251, 265–267, 269, 273, 274, 277, 280, 292, 302, 304, 307).

Agarwal, Mridul, Vaneet Aggarwal, and Tian Lan (2022). Multi-Objective Reinforcement Learning with Non-Linear Scalarization. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. AAMAS '22. Virtual Event, New Zealand: International Foundation for Autonomous Agents and Multiagent Systems, pp. 9–17 (Cited on page 113).

Agarwal, Naman, Brian Bullins, and Elad Hazan (2017). Second-Order Stochastic Optimization for Machine Learning in Linear Time. *Journal of Machine Learning Research* 18.116, pp. 1–40 (Cited on pages 20, 21, 47).

Ailon, Nir and Bernard Chazelle (May 2009). The Fast Johnson-Lindenstrauss Transform and Approximate Nearest Neighbors. *SIAM J. Comput.* 39.1, pp. 302–322 (Cited on page 23).

Alfano, Carlo and Patrick Rebeschini (2022). *Linear Convergence for Natural Policy Gradient with Log-linear Policy Parametrization* (Cited on pages 106, 268, 269).

Alfano, Carlo, Rui Yuan, and Patrick Rebeschini (2023). *A Novel Framework for Policy Mirror Descent with General Parametrization and Linear Convergence* (Cited on page 106).

Amari, Shun-ichi (Feb. 1998). Natural Gradient Works Efficiently in Learning. *Neural Computation* 10.2, pp. 251–276 (Cited on pages 7, 8, 87, 122, 123).

An, Hengbin and Z. Bai (2007). A globally convergent Newton-GMRES method for large sparse systems of nonlinear equations. *Applied Numerical Mathematics* 57, pp. 235–252 (Cited on page 22).

Arjevani, Yossi, Yair Carmon, John C. Duchi, Dylan J. Foster, Nathan Srebro, and Blake Woodworth (June 2022). Lower bounds for non-convex stochastic optimization. *Mathematical Programming* (Cited on page 112).

Bach, Francis (2021). *Learning Theory from First Principles*. The MIT Press. DRAFT (Cited on page 33).

Bach, Francis and Eric Moulines (2013). Non-strongly-convex smooth stochastic approximation with convergence rate O(1/n). In *Advances in Neural Information Processing Systems*. Vol. 26. Curran Associates, Inc. (Cited on pages 101, 279, 293, 307, 310).

Bai, Qinbo, Mridul Agarwal, and Vaneet Aggarwal (2021). *Joint Optimization of Multi-Objective Reinforcement Learning with Policy Gradient Based Algorithm* (Cited on page 113).

Bauschke, Heinz H. and Patrick L. Combettes (2017). *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. 2nd edition. Springer (Cited on page 209).

Baxter, J. and P. L. Bartlett (Nov. 2001). Infinite-Horizon Policy-Gradient Estimation. *Journal of Artificial Intelligence Research* 15, pp. 319–350 (Cited on pages 7, 9, 67, 70, 87, 122, 124).

Beck, A and M Teboulle (2003). Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters* 31.3, pp. 167–175 (Cited on pages 7, 87, 122).

Beck, Amir (2017). *First-Order Methods in Optimization*. Philadelphia, PA, USA: SIAM-Society for Industrial and Applied Mathematics (Cited on pages 74, 307).

Bellavia, S. and B. Morini (2001). A Globally Convergent Newton-GMRES Subspace Method for Systems of Nonlinear Equations. *SIAM J. Sci. Comput.* 23, pp. 940–960 (Cited on page 22).

Ben-Israel, Adi and A. Charnes (1963). Contributions to the Theory of Generalized Inverses. *Journal of the Society for Industrial and Applied Mathematics* 11.3, pp. 667–699 (Cited on page 173).

Berrada, Leonard, Andrew Zisserman, and M. Pawan Kumar (13–18 Jul 2020). Training Neural Networks for and by Interpolation. In *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 799–809 (Cited on page 42).

Bertsekas, D. (2012). *Dynamic Programming and Optimal Control: Volume II; Approximate Dynamic Programming*. Athena Scientific optimization and computation series. Athena Scientific (Cited on pages 8, 87, 98, 123).

Bhandari, Jalaj and Daniel Russo (2019). *Global Optimality Guarantees For Policy Gradient Methods* (Cited on pages 213, 268).

— (13–15 Apr 2021). On the Linear Convergence of Policy Gradient Methods for Finite MDPs. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. Vol. 130. Proceedings of Machine Learning Research. PMLR, pp. 2386–2394 (Cited on pages 267, 268).

Bhatnagar, Shalabh, Richard S. Sutton, Mohammad Ghavamzadeh, and Mark Lee (2009). Natural actor–critic algorithms. *Automatica* 45.11, pp. 2471–2482 (Cited on pages 67, 105).

Bjerhammar, Arne (1951). Application of calculus of matrices to method of least squares; with special references to geodetic calculations. *Transactions of Royal Institute of Technology* 49 (Cited on page 18).

Björklund, Andreas, Petteri Kaski, and Ryan Williams (2019). Solving Systems of Polynomial Equations over GF(2) by a Parity-Counting Self-Reduction. In *46th International Colloquium on Automata, Languages, and Programming*. Vol. 132. LIPIcs, 26:1–26:13 (Cited on page 18).

Blackard, Jock and Denis Dean (1999). Comparative Accuracies of Artificial Neural Networks and Discriminant Analysis in Predicting Forest Cover Types from Cartographic Variables. *Computers and Electronics in Agriculture* 24, pp. 131–151 (Cited on pages 40, 54).

Bollapragada, Raghu, Richard H Byrd, and Jorge Nocedal (Apr. 2018). Exact and inexact subsampled Newton methods for optimization. *IMA Journal of Numerical Analysis* 39.2, pp. 545–578 (Cited on pages 20, 21, 46).

Bottou, Léon, Frank E. Curtis, and Jorge Nocedal (2018). Optimization methods for large-scale machine learning. English (US). *SIAM Review* 60.2, pp. 223–311 (Cited on page 72).

Bregman, L.M. (1967). The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics* 7.3, pp. 200–217 (Cited on page 309).

Byrd, R. H., S. L. Hansen, Jorge Nocedal, and Y. Singer (2016). A Stochastic Quasi-Newton Method for Large-Scale Optimization. *SIAM Journal on Optimization* 26.2, pp. 1008–1031 (Cited on page 105).

Byrd, Richard H, Gillian M Chin, Will Neveitt, and Jorge Nocedal (2011). On the Use of Stochastic Hessian Information in Optimization Methods for Machine Learning. *SIAM Journal on Optimization* 21.3, pp. 977–995 (Cited on page 47).

Calandriello, Daniele, Alessandro Lazaric, and Michal Valko (2017). Efficient Second-Order Online Kernel Learning with Adaptive Embedding. In *Advances in Neural Information Processing Systems 30*, pp. 6140–6150 (Cited on page 21).

Candès, E. J., X. Li, and M. Soltanolkotabi (2015). Phase Retrieval via Wirtinger Flow: Theory and Algorithms. *IEEE Transactions on Information Theory* 61.4, pp. 1985–2007 (Cited on page 18).

Cartis, Coralia, Nicholas I M Gould, and Philippe L Toint (May 2009). Adaptive cubic regularisation methods for unconstrained optimization . Part I : motivation , convergence and numerical results. *Mathematical Programming* 127.2, pp. 1–38 (Cited on page 22).

Cayci, Semih, Niao He, and R. Srikant (2021). *Linear Convergence of Entropy-Regularized Natural Policy Gradient with Linear Function Approximation* (Cited on pages 95, 101, 103, 105, 267, 269, 306, 307).

— (2022a). *Finite-Time Analysis of Entropy-Regularized Neural Natural Actor-Critic Algorithm* (Cited on pages 111, 267, 269).

— (2022b). *Learning to Control Partially Observed Systems with Finite Memory* (Cited on page 113).

Cen, Shicong, Chen Cheng, Yuxin Chen, Yuting Wei, and Yuejie Chi (2021a). Fast Global Convergence of Natural Policy Gradient Methods with Entropy Regularization. In *Operations Research* (Cited on pages 87, 266).

Cen, Shicong, Yuejie Chi, Simon S. Du, and Lin Xiao (2022). *Faster Last-iterate Convergence of Policy Optimization in Zero-Sum Markov Games* (Cited on page 113).

Cen, Shicong, Yuting Wei, and Yuejie Chi (2021b). Fast Policy Extragradient Methods for Competitive Games with Entropy Regularization. In *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., pp. 27952–27964 (Cited on page 113).

Censor, Y. and S.A. Zenios (1997). *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford University Press, USA (Cited on page 309).

Cettolo, Mauro, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico (Dec. 2014). Report on the 11th IWSLT evaluation campaign. In *Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign*. Lake Tahoe, California, pp. 2–17 (Cited on page 43).

Chang, Chih-Chung and Chih-Jen Lin (2001). IJCNN 2001 challenge: generalization ability and text decoding. In *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222)*. Vol. 2, 1031–1036 vol.2 (Cited on pages 40, 54).

— (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2 (3), 27:1–27:27 (Cited on pages 40, 54, 179).

Chen, Gong and Marc Teboulle (1993). Convergence Analysis of a Proximal-Like Minimization Algorithm Using Bregman Functions. *SIAM Journal on Optimization* 3.3, pp. 538–543 (Cited on pages 97, 308).

Chen, Jiabin, Rui Yuan, Guillaume Garrigos, and Robert M. Gower (28–30 Mar 2022a). SAN: Stochastic Average Newton Algorithm for Minimizing Finite Sums. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. Vol. 151. Proceedings of Machine Learning Research. PMLR, pp. 279–318 (Cited on pages 43, 45, 150).

Chen, Zaiwei, Sajad Khodadadian, and Siva Theja Maguluri (2022b). Finite-Sample Analysis of Off-Policy Natural Actor–Critic With Linear Function Approximation. *IEEE Control Systems Letters* 6, pp. 2611–2616 (Cited on pages 111, 267, 269).

Chen, Zaiwei and Siva Theja Maguluri (28–30 Mar 2022). Sample Complexity of Policy-Based Methods under Off-Policy Sampling and Linear Function Approximation. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. Vol. 151. Proceedings of Machine Learning Research. PMLR, pp. 11195–11214 (Cited on pages 87, 105, 267, 269, 306, 307).

Christianson, Bruce (1992). Automatic Hessians by reverse accumulation. *IMA Journal of Numerical Analysis* 12.2, pp. 135–150 (Cited on pages 19, 51).

Clarkson, Kenneth L. and David P. Woodruff (Jan. 2017). Low-Rank Approximation and Regression in Input Sparsity Time. *J. ACM* 63.6 (Cited on page 110).

Conn, Andrew R., Nicholas I. M. Gould, and Philippe L. Toint (2000). *Trust-region Methods*. Society for Industrial and Applied Mathematics (Cited on page 22).

Crammer, Koby, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer (2006). Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research* 7.19, pp. 551–585 (Cited on page 42).

Cutkosky, Ashok and Francesco Orabona (2019). Momentum-Based Variance Reduction in Non-Convex SGD. In *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc. (Cited on pages 7, 112, 122).

De Farias, D. P. and B. Van Roy (Nov. 2003). The Linear Programming Approach to Approximate Dynamic Programming. *Oper. Res.* 51.6, pp. 850–865 (Cited on page 87).

Defazio, Aaron, Francis Bach, and Simon Lacoste-julien (2014). SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives. In *Advances in Neural Information Processing Systems 27*, pp. 1646–1654 (Cited on pages 21, 53, 56, 61).

Derezinski, Michal, Feynman T Liang, Zhenyu Liao, and Michael W Mahoney (2020). Precise expressions for random projections: Low-rank approximation and randomized Newton. In *Advances in Neural Information Processing Systems* (Cited on page 149).

Deuflhard, Peter (2011). *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*. Springer Publishing Company, Incorporated (Cited on pages 5, 22, 31, 33, 34, 119, 139, 140, 142).

Ding, Dongsheng, Kaiqing Zhang, Tamer Basar, and Mihailo Jovanovic (2020). Natural Policy Gradient Primal-Dual Method for Constrained Markov Decision Processes. In *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., pp. 8378–8390 (Cited on page 113).

Ding, Yuhao, Junzi Zhang, and Javad Lavaei (2021). *Beyond Exact Gradients: Convergence of Stochastic Soft-Max Policy Gradient Methods with Entropy Regularization* (Cited on page 82).

— (28–30 Mar 2022). On the Global Optimum Convergence of Momentum-based Policy Gradient. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. Vol. 151. Proceedings of Machine Learning Research. PMLR, pp. 1910–1934 (Cited on pages 68, 82, 83, 105, 112, 253).

Doikov, Nikita, Konstantin Mishchenko, and Yurii Nesterov (2022). Super-Universal Regularized Newton Method. *arXiv preprint arXiv:2208.05888* (Cited on page 110).

Doikov, Nikita and Yurii Nesterov (2021). Gradient Regularization of Newton Method with Bregman Distances. In (Cited on page 110).

Dua, Dheeru and Casey Graff (2017). *UCI Machine Learning Repository* (Cited on pages 40, 54).

Duan, Yan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel (2016). Benchmarking Deep Reinforcement Learning for Continuous Control. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. New York, NY, USA: JMLR.org, pp. 1329–1338 (Cited on page 105).

Duchi, John, Elad Hazan, and Yoram Singer (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research* 12.61, pp. 2121–2159 (Cited on pages 3, 117).

Erdogdu, Murat A. and Andrea Montanari (2015). Convergence rates of sub-sampled Newton methods. In *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., pp. 3052–3060 (Cited on pages 20, 21, 47).

Fang, Cong, Chris Junchi Li, Zhouchen Lin, and Tong Zhang (2018). SPIDER: Near-Optimal Non-Convex Optimization via Stochastic Path-Integrated Differential Estimator. In *Advances in Neural Information Processing Systems*. Vol. 31, pp. 689–699 (Cited on pages 7, 112, 122).

Fatkhullin, Ilyas, Jalal Etesami, Niao He, and Negar Kiyavash (2022). Sharp Analysis of Stochastic Optimization under Global Kurdyka-Łojasiewicz Inequality. In *Advances in Neural Information Processing Systems* (Cited on pages 84, 111, 112).

Fazel, Maryam, Rong Ge, Sham Kakade, and Mehran Mesbahi (Oct. 2018). Global Convergence of Policy Gradient Methods for the Linear Quadratic Regulator. In *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 1467–1476 (Cited on pages 67, 69, 84, 212, 259, 268).

Fountoulakis, Kimon and Jacek Gondzio (2016). A second-order method for strongly convex L1 -regularization problems. *Math. Program.* 156.1-2, pp. 189–219 (Cited on pages 54, 180).

Freund, Roland W., Gene H. Golub, and Noël M. Nachtigal (1992). Iterative solution of linear systems. *Acta Numerica* 1, pp. 57–100 (Cited on page 51).

Gao, Wenbo and Donald Goldfarb (2019). Quasi-Newton methods: superlinear convergence without line searches for self-concordant functions. *Optimization Methods and Software* 34.1, pp. 194–217 (Cited on page 22).

Gao, Zhan, Alec Koppel, and Alejandro Ribeiro (2020). Incremental Greedy BFGS: An Incremental Quasi-Newton Method with Explicit Superlinear Rate. In *Advances in neural information processing systems, 12th OPT Workshop on Optimization for Machine Learning* (Cited on page 47).

Gazagnadou, Nidham, Mark Ibrahim, and Robert M. Gower (2022). RidgeSketch: A Fast Sketching Based Solver for Large Scale Ridge Regression. *SIAM Journal on Matrix Analysis and Applications* 43.3, pp. 1440–1468 (Cited on page 110).

Geist, Matthieu, Bruno Scherrer, and Olivier Pietquin (Sept. 2019). A Theory of Regularized Markov Decision Processes. In *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 2160–2169 (Cited on page 93).

Ghadimi, Saeed and Guanghui Lan (2013). Stochastic First- and Zeroth-Order Methods for Nonconvex Stochastic Programming. eng. *SIAM journal on optimization* 23.4, pp. 2341–2368 (Cited on pages 72, 112).

Goldfarb, Donald (1970). A Family of Variable-Metric Methods Derived by Variational Means. *Mathematics of Computation* 24.109, pp. 23–26 (Cited on page 51).

Gower, Robert, Donald Goldfarb, and Peter Richtarik (20–22 Jun 2016). Stochastic Block BFGS: Squeezing More Curvature out of Data. In *Proceedings of The 33rd International Conference on Machine Learning*. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, pp. 1869–1878 (Cited on pages 21, 47, 105).

Gower, Robert, Dmitry Koralev, Felix Lieder, and Peter Richtarik (2019a). RSN: Randomized Subspace Newton. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., pp. 614–623 (Cited on pages 21, 57, 109, 134, 149–151, 193).

Gower, Robert M., Mathieu Blondel, Nidham Gazagnadou, and Fabian Pedregosa (2022). *Cutting Some Slack for SGD with Adaptive Polyak Stepsizes* (Cited on page 43).

Gower, Robert M., Aaron Defazio, and Mike Rabbat (2021a). Stochastic Polyak Stepsize with a Moving Target. In *Advances in neural information processing systems, 13th Annual Workshop on Optimization for Machine Learning (OPT2021)* (Cited on pages 42, 109).

Gower, Robert M. and Peter Richtárik (2015a). Stochastic Dual Ascent for Solving Linear Systems. *arXiv:1512.06890* (Cited on pages 22, 149, 198, 200).

Gower, Robert M., Peter Richtárik, and Francis Bach (July 2021b). Stochastic quasi-gradient methods: variance reduction via Jacobian sketching. *Mathematical Programming* 188.1, pp. 135–192 (Cited on page 72).

Gower, Robert M., Othmane Sebbouh, and Nicolas Loizou (2021c). SGD for Structured Nonconvex Functions: Learning Rates, Minibatching and Interpolation. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. Vol. 130. PMLR, pp. 1315–1323 (Cited on pages 46, 112).

Gower, Robert Mansel, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik (Sept. 2019b). SGD: General Analysis and Improved Rates. In *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 5200–5209 (Cited on pages 28, 72, 112).

Gower, Robert Mansel and Peter Richtárik (2015b). Randomized Iterative Methods for Linear Systems. *SIAM Journal on Matrix Analysis and Applications* 36.4, pp. 1660–1690 (Cited on pages 4, 19, 22, 48, 59, 118, 119, 198, 200).

Grudzien, Jakub, Christian A Schroeder De Witt, and Jakob Foerster (17–23 Jul 2022). Mirror Learning: A Unifying Framework of Policy Optimisation. In *Proceedings of the 39th International Conference on Machine Learning*. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 7825–7844 (Cited on page 87).

Gu*, Shixiang, Ethan Holly*, Timothy Lillicrap, and Sergey Levine (May 2017). Deep Reinforcement Learning for Robotic Manipulation with Asynchronous Off-Policy Updates. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. *equal contribution. Piscataway, NJ, USA: IEEE (Cited on pages 6, 120).

Gupta, Vineet, Tomer Koren, and Yoram Singer (Oct. 2018). Shampoo: Preconditioned Stochastic Tensor Optimization. In *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 1842–1850 (Cited on pages 3, 118).

Gürbüzbalaban, M., A. Ozdaglar, and P. Parrilo (Apr. 2015). A globally convergent incremental Newton method. *Mathematical Programming* 151.1, pp. 283–313 (Cited on page 21).

Haarnoja, Tuomas, Aurick Zhou, Pieter Abbeel, and Sergey Levine (Oct. 2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 1861–1870 (Cited on pages 8, 80, 87, 123).

Hinder, Oliver, Aaron Sidford, and Nimit Sohoni (2020). Near-Optimal Methods for Minimizing Star-Convex Functions and Beyond. In *Proceedings of Thirty Third Conference on Learning Theory*, pp. 1894–1938 (Cited on pages 22, 27).

Hsu, Daniel, Sham M. Kakade, and Tong Zhang (25–27 Jun 2012). Random Design Analysis of Ridge Regression. In *Proceedings of the 25th Annual Conference on Learning Theory*. Ed. by Shie Mannor, Nathan Srebro, and Robert C. Williamson. Vol. 23. Proceedings of Machine Learning Research. Edinburgh, Scotland: PMLR, pp. 9.1–9.24 (Cited on page 101).

Hu, Yuzheng, Ziwei Ji, and Matus Telgarsky (2022). Actor-critic is implicitly biased towards high entropy optimal policies. In *International Conference on Learning Representations* (Cited on pages 111, 267, 269).

Huang, Feihu, Shangqian Gao, and Heng Huang (2022). Bregman Gradient Policy Optimization. In *International Conference on Learning Representations* (Cited on pages 8, 67, 112, 122).

Huang, Feihu, Shangqian Gao, Jian Pei, and Heng Huang (13–18 Jul 2020). Momentum-Based Policy Gradient Methods. In *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 4422–4433 (Cited on pages 7, 67, 76, 105, 122, 216).

Ioffe, Sergey and Christian Szegedy (July 2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 448–456 (Cited on page 46).

Jiang, Nan, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E. Schapire (June 2017). Contextual Decision Processes with low Bellman rank are PAC-Learnable. In *Proceedings of the 34th International Conference on Machine Learning*. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 1704–1713 (Cited on page 96).

Jin, Chi, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan (Sept. 2020). Provably efficient reinforcement learning with linear function approximation. In *Proceedings of Thirty Third Conference on Learning Theory*. Vol. 125. Proceedings of Machine Learning Research. PMLR, pp. 2137–2143 (Cited on pages 96, 98, 111, 267).

Jin, Qiujiang and Aryan Mokhtari (Sept. 2022). Non-asymptotic superlinear convergence of standard quasi-Newton methods. *Mathematical Programming* (Cited on page 58).

Johnson, Rie and Tong Zhang (2013). Accelerating Stochastic Gradient Descent using Predictive Variance Reduction. In *Advances in Neural Information Processing Systems*. Vol. 26. Curran Associates, Inc., pp. 315–323 (Cited on pages 5, 7, 21, 41, 46, 53, 54, 61, 112, 119, 120, 122, 154, 160, 182).

Kaczmarz, Stefan Marian (1937). Angenäherte Auflösung von Systemen linearer Gleichungen. *Bulletin International de l'Académie Polonaise des Sciences et des Lettres. Classe des Sciences Mathématiques et Naturelles. Série A, Sciences Mathématiques* 35, pp. 355–357 (Cited on page 34).

Kakade, Sham and John Langford (2002). Approximately Optimal Approximate Reinforcement Learning. In *Proceedings of 19th International Conference on Machine Learning*, pp. 267–274 (Cited on pages 98, 268, 273).

Kakade, Sham M (2001). A Natural Policy Gradient. In *Advances in Neural Information Processing Systems*. Vol. 14. MIT Press (Cited on pages 7, 8, 67, 87, 91, 105, 122, 123, 271).

Kantorovitch, L. (1939). The method of successive approximation for functional equations. *Acta Math.* 71, pp. 63–97 (Cited on page 22).

Karush, William (1939). Minima of Functions of Several Variables with Inequalities as Side Conditions. MA thesis. Chicago, IL, USA: Department of Mathematics, University of Chicago (Cited on pages 18, 167).

Kawaguchi, Kenji (2016). Deep Learning without Poor Local Minima. In *Advances in Neural Information Processing Systems 29*, pp. 586–594 (Cited on page 18).

Kelley, C. T. (2018). Numerical methods for nonlinear equations. *Acta Numerica* 27, pp. 207–287 (Cited on page 22).

Khaled, Ahmed and Peter Richtárik (2023). Better Theory for SGD in the Nonconvex World. *Transactions on Machine Learning Research*. Survey Certification (Cited on pages xiv, 9, 72, 73, 84, 111, 112, 124, 211, 212, 223, 259).

Khodadadian, Sajad, Zaiwei Chen, and Siva Theja Maguluri (18–24 Jul 2021a). Finite-Sample Analysis of Off-Policy Natural Actor-Critic Algorithm. In *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 5420–5431 (Cited on page 266).

Khodadadian, Sajad, Thinh T. Doan, Justin Romberg, and Siva Theja Maguluri (2022a). Finite Sample Analysis of Two-Time-Scale Natural Actor-Critic Algorithm. *IEEE Transactions on Automatic Control*, pp. 1–16 (Cited on page 266).

Khodadadian, Sajad, Prakirt Raj Jhunjhunwala, Sushil Mahavir Varma, and Siva Theja Maguluri (2021b). On the Linear Convergence of Natural Policy Gradient Algorithm. In *2021 60th IEEE Conference on Decision and Control (CDC)*. Austin, TX, USA: IEEE Press, pp. 3794–3799 (Cited on pages 87, 267).

— (2022b). On linear and super-linear convergence of Natural Policy Gradient algorithm. *Systems & Control Letters* 164, p. 105214 (Cited on page 267).

Kingma, Diederik P. and Jimmy Ba (2015). Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015* (Cited on pages 3, 5, 43, 46, 117, 119).

Kiran, B Ravi, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A. Al Sallab, Senthil Yogamani, and Patrick Pérez (2022). Deep Reinforcement Learning for Autonomous Driving: A Survey. *IEEE Transactions on Intelligent Transportation Systems* 23.6, pp. 4909–4926 (Cited on pages 6, 120).

Kober, Jens, J. Andrew Bagnell, and Jan Peters (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research* 32.11, pp. 1238–1274 (Cited on pages 6, 120).

Kohavi, Ron (1996). Scaling up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid. In *International Conference on Knowledge Discovery and Data Mining*. AAAI Press, pp. 202–207 (Cited on page 40).

Kohler, Jonas Moritz and Aurélien Lucchi (2017). Sub-sampled Cubic Regularization for Non-convex Optimization. In *Proceedings of the 34th International Conference on Machine Learning*. Vol. 70, pp. 1895–1904 (Cited on pages 20, 21, 47).

Konda, Vijay and John Tsitsiklis (2000). Actor-Critic Algorithms. In *Advances in Neural Information Processing Systems*. Vol. 12. MIT Press, pp. 1008–1014 (Cited on pages 7, 67, 87, 122).

Kovalev, Dmitry, Konstantin Mishchenko, and Peter Richtarik (2019). Stochastic Newton and Cubic Newton Methods with Simple Local Linear-Quadratic Rates. In *Advances in neural information processing systems, Workshop on Beyond First Order Methods in ML* (Cited on pages 4, 20–22, 34–37, 47, 48, 52, 109, 119, 144, 184, 186).

Kozuno, Tadashi, Wenhao Yang, Nino Vieillard, Toshinori Kitamura, Yunhao Tang, Jincheng Mei, Pierre Ménard, Mohammad Gheshlaghi Azar, Michal Valko, Rémi Munos, Olivier Pietquin, Matthieu Geist, and Csaba Szepesvári (2022). *KL-Entropy-Regularized RL with a Generative Model is Minimax Optimal* (Cited on page 93).

Krizhevsky, Alex (2009). Learning Multiple Layers of Features from Tiny Images. In (Cited on page 43).

Kuhn, Harold W. and Albert W. Tucker (1951). Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley and Los Angeles, pp. 481–492 (Cited on pages 18, 167).

Kumar, Harshat, Alec Koppel, and Alejandro Ribeiro (2021). *On the Sample Complexity of Actor-Critic Method for Reinforcement Learning with Function Approximation* (Cited on page 111).

Kurdyka, Krzysztof (1998). On gradients of functions definable in o-minimal structures. *Annales de l'institut Fourier* 48.3, pp. 769–783 (Cited on page 74).

Lan, Guanghui (Apr. 2022). Policy mirror descent for reinforcement learning: linear convergence, new sampling complexity, and generalized problem classes. *Mathematical Programming* (Cited on pages 7, 9, 87, 93, 107, 122, 124, 266, 305–307).

Lazaric, Alessandro, Mohammad Ghavamzadeh, and Rémi Munos (2016). Analysis of Classification-based Policy Iteration Algorithms. *Journal of Machine Learning Research* 17.19, pp. 1–30 (Cited on page 101).

Lee, Jasper C. H. and Paul Valiant (2016). Optimizing Star-Convex Functions. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS*, pp. 603–614 (Cited on pages 22, 27, 31, 35).

Lei, Yunwen, Ting Hu, Guiying Li, and Ke Tang (2020). Stochastic Gradient Descent for Non-convex Learning Without Bounded Gradient Assumptions. *IEEE Transactions on Neural Networks and Learning Systems* 31.10, pp. 4394–4400 (Cited on page 72).

Leitão, A and B F Svaiter (Jan. 2016). On projective Landweber–Kaczmarz methods for solving systems of nonlinear ill-posed equations. *Inverse Problems* 32.2, p. 025004 (Cited on page 34).

Leonardos, Stefanos, Will Overman, Ioannis Panageas, and Georgios Piliouras (2022). Global Convergence of Multi-Agent Policy Gradient in Markov Potential Games. In *International Conference on Learning Representations* (Cited on page 112).

Leventhal, D. and A. S. Lewis (2010). Randomized Methods for Linear Constraints: Convergence Rates and Conditioning. *Mathematics of Operations Research* 35.3, pp. 641–654 (Cited on page 200).

Levine, Sergey, Chelsea Finn, Trevor Darrell, and Pieter Abbeel (2016). End-to-End Training of Deep Visuomotor Policies. *Journal of Machine Learning Research* 17.39, pp. 1–40 (Cited on pages 6, 120).

Levine, Sergey, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen (2018). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research* 37.4-5, pp. 421–436 (Cited on pages 6, 120).

Lewis, D. D., Y. Yang, T. G. Rose, and F. Li (2004). RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research* 5.Apr, pp. 361–397 (Cited on page 54).

Li, Gen, Yuting Wei, Yuejie Chi, Yuantao Gu, and Yuxin Chen (2020). Sample Complexity of Asynchronous Q-Learning: Sharper Analysis and Variance Reduction. In *Advances in Neural Information Processing Systems*. Vol. 33, pp. 7031–7043 (Cited on page 101).

— (15–19 Aug 2021a). Softmax Policy Gradient Methods Can Take Exponential Time to Converge. In *Proceedings of Thirty Fourth Conference on Learning Theory*. Vol. 134. Proceedings of Machine Learning Research. PMLR, pp. 3107–3110 (Cited on pages 80, 213).

Li, Shuang, William J. Swartworth, Martin Takáč, Deanna Needell, and Robert M. Gower (2022a). *SP2: A Second Order Stochastic Polyak Method* (Cited on page 43).

Li, Tianjiao, Feiyang Wu, and Guanghui Lan (2022b). *Stochastic first-order methods for average-reward Markov decision processes* (Cited on pages 107, 266).

Li, Yan, Tuo Zhao, and Guanghui Lan (2022c). *Homotopic Policy Mirror Descent: Policy Convergence, Implicit Regularization, and Improved Sample Complexity* (Cited on page 266).

Li, Zhize, Hongyan Bao, Xiangliang Zhang, and Peter Richtarik (18–24 Jul 2021b). PAGE: A Simple and Optimal Probabilistic Gradient Estimator for Nonconvex Optimization. In *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 6286–6295 (Cited on pages 7, 112, 122).

Lillicrap, Timothy P., Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra (2016). Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings* (Cited on pages 8, 87, 123).

Liu, Yang and Fred Roosta (2021). Convergence of Newton-MR under Inexact Hessian Information. *SIAM J. Optim.* 31.1, pp. 59–90 (Cited on page 46).

Liu, Yanli, Kaiqing Zhang, Tamer Basar, and Wotao Yin (2020). An Improved Analysis of (Variance-Reduced) Policy Gradient and Natural Policy Gradient Methods. In *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., pp. 7624–7636 (Cited on pages 8, 67–69, 74, 76, 77, 79, 83, 104, 105, 112, 123, 215, 216, 254, 266, 267, 269, 303).

Loizou, Nicolas, Sharan Vaswani, Issam Hadj Laradji, and Simon Lacoste-Julien (13–15 Apr 2021). Stochastic Polyak Step-size for SGD: An Adaptive Learning Rate for Fast Convergence. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. Vol. 130. Proceedings of Machine Learning Research. PMLR, pp. 1306–1314 (Cited on page 42).

Łojasiewicz, Stanisław (1959). Sur le problème de la division. *Studia Mathematica* 18, pp. 87–136 (Cited on page 259).

— (1963). *Une propriété topologique des sous-ensembles analytiques réels*. French. Equ. Derivees partielles, Paris 1962, Colloques internat. Centre nat. Rech. sci. 117, 87-89 (1963). (Cited on pages 212, 259, 268).

Lu, Sha, Zengxin Wei, and Lue Li (Oct. 2010). A trust region algorithm with adaptive cubic regularization methods for nonsmooth convex minimization. *Computational Optimization and Applications* 51.2, pp. 551–573 (Cited on page 22).

Lu, Yichao, Paramveer Dhillon, Dean P Foster, and Lyle Ungar (2013). Faster Ridge Regression via the Subsampled Randomized Hadamard Transform. In *Advances in Neural Information Processing Systems*. Vol. 26. Curran Associates, Inc. (Cited on page 110).

Luo, Haipeng, Alekh Agarwal, Nicolò Cesa-Bianchi, and John Langford (2016). Efficient Second Order Online Learning by Sketching. In *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., pp. 902–910 (Cited on page 21).

Ma, Siyuan, Raef Bassily, and Mikhail Belkin (2018). The Power of Interpolation: Understanding the Effectiveness of SGD in Modern Over-parametrized Learning. In *Proceedings of the 35th International Conference on Machine Learning* (Cited on pages 4, 21, 26, 46, 119).

Martens, James (2020). New Insights and Perspectives on the Natural Gradient Method. *Journal of Machine Learning Research* 21.146, pp. 1–76 (Cited on pages 8, 87, 123).

Masiha, Saeed, Saber Salehkaleybar, Niao He, Negar Kiyavash, and Patrick Thiran (2022). Stochastic Second-Order Methods Improve Best-Known Sample Complexity of SGD for Gradient-Dominated Functions. In *Advances in Neural Information Processing Systems* (Cited on page 113).

Mei, Jincheng, Yue Gao, Bo Dai, Csaba Szepesvari, and Dale Schuurmans (18–24 Jul 2021). Leveraging Non-uniformity in First-order Non-convex Optimization. In *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 7555–7564 (Cited on pages 74, 268).

Mei, Jincheng, Chenjun Xiao, Ruitong Huang, Dale Schuurmans, and Martin Müller (July 2019). On Principled Entropy Exploration in Policy Optimization. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, pp. 3130–3136 (Cited on pages 68, 80).

Mei, Jincheng, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans (13–18 Jul 2020). On the Global Convergence Rates of Softmax Policy Gradient Methods. In *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 6820–6829 (Cited on pages 8, 67–69, 74, 123, 213, 215, 216, 226, 243, 247, 258–260, 268).

Melo, Francisco S., Sean P. Meyn, and M. Isabel Ribeiro (2008). An analysis of reinforcement learning with function approximation. In *ICML*, pp. 664–671 (Cited on page 101).

Mishchenko, Konstantin (2021). Regularized Newton Method with Global $O(1/k^2)$ Convergence. *arXiv preprint arXiv:2112.02089* (Cited on page 110).

Mitrophanov, A. Yu. (2005). Sensitivity and Convergence of Uniformly Ergodic Markov Chains. *Journal of Applied Probability* 42.4, pp. 1003–1014 (Cited on page 214).

Mnih, Volodymyr, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu (20–22 Jun 2016). Asynchronous Methods for Deep Reinforcement Learning. In *Proceedings of The 33rd International Conference on Machine Learning*. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, pp. 1928–1937 (Cited on pages 8, 67, 80, 87, 123).

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis (Feb. 2015). Human-level control through deep reinforcement learning. *Nature* 518.7540, pp. 529–533 (Cited on pages 6, 120).

Mohammad, Rami, Fadi Thabtah, and T. Mccluskey (Jan. 2012). An assessment of features related to phishing websites using an automated technique. In *2012 International Conference for Internet Technology and Secured Transactions (ICITST 2012)*. IEEE, pp. 492–497 (Cited on pages 40, 54).

Mokhtari, Aryan, Mark Eisen, and Alejandro Ribeiro (2018). IQN: An Incremental Quasi-Newton Method with Local Superlinear Convergence Rate. *SIAM Journal on Optimization* 28.2, pp. 1670–1698 (Cited on pages 47, 184).

Mokhtari, Aryan and Alejandro Ribeiro (2015). Global Convergence of Online Limited Memory BFGS. *The Journal of Machine Learning Research* 16, pp. 3151–3181 (Cited on page 47).

Moore, Eliakim Hastings (1920). On the reciprocal of the general algebraic matrix. *Bulletin of the American Mathematical Society* 26, pp. 394–395 (Cited on page 18).

Moritz, Philipp, Robert Nishihara, and Michael I. Jordan (2016). A Linearly-Convergent Stochastic L-BFGS Algorithm. In *International Conference on Artificial Intelligence and Statistics*. Vol. 51, pp. 249–258 (Cited on page 47).

Munos, Rémi (2003). Error Bounds for Approximate Policy Iteration. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*. ICML'03. Washington, DC, USA: AAAI Press, pp. 560–567 (Cited on page 96).

— (2005). Error Bounds for Approximate Value Iteration. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*. AAAI'05. Pittsburgh, Pennsylvania: AAAI Press, pp. 1006–1011 (Cited on page 96).

Munos, Rémi and Csaba Szepesvári (2008). Finite-Time Bounds for Fitted Value Iteration. *Journal of Machine Learning Research* 9.27, pp. 815–857 (Cited on page 96).

Mutny, Mojmir, Michał Dereziński, and Andreas Krause (2020). Convergence Analysis of Block Coordinate Algorithms with Determinantal Sampling. In *International Conference on Artificial Intelligence and Statistics*, pp. 3110–3120 (Cited on pages 110, 149).

Nachum, Ofir, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans (2017). Bridging the Gap Between Value and Policy Based Reinforcement Learning. In *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc. (Cited on page 80).

Needell, Deanna (June 2010). Randomized Kaczmarz solver for noisy linear systems. *BIT Numerical Mathematics* 50.2, pp. 395–403 (Cited on page 34).

Needell, Deanna, Nathan Srebro, and Rachel Ward (Jan. 2016). Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. *Mathematical Programming* 155.1, pp. 549–573 (Cited on page 34).

Nemirovski, A and D Yudin (1983). *Problem Complexity and Method Efficiency in Optimization*. Wiley Interscience (Cited on pages 7, 87, 122).

Nesterov, Yurii (2014). *Introductory Lectures on Convex Optimization: A Basic Course*. 2nd ed. Springer Publishing Company, Incorporated (Cited on page 27).

Nesterov, Yurii and Arkadii Nemirovskii (1994). *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics (Cited on page 22).

Nesterov, Yurii E. and Boris T. Polyak (2006). Cubic regularization of Newton method and its global performance. *Math. Program.* 108.1, pp. 177–205 (Cited on pages 22, 27, 31).

Netzer, Yuval, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng (2011). Reading Digits in Natural Images with Unsupervised Feature Learning. In *Advances in neural information processing systems, Workshop on Deep Learning and Unsupervised Feature Learning* (Cited on page 43).

Neu, Gergely, Anders Jonsson, and Vicenç Gómez (2017). *A unified view of entropy-regularized Markov decision processes* (Cited on page 87).

Nguyen, Lam M., Jie Liu, Katya Scheinberg, and Martin Takáč (June 2017). SARAH: A Novel Method for Machine Learning Problems Using Stochastic Recursive Gradient. In *Proceedings of the 34th International Conference on Machine Learning*. Vol. 70. Proceedings of Machine Learning Research. International Convention Centre, Sydney, Australia: PMLR, pp. 2613–2621 (Cited on pages 7, 112, 122).

Nocedal, Jorge and Stephen J. Wright (1999). *Numerical Optimization*. Vol. 43. Springer Series in Operations Research. Springer (Cited on page 18).

OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, et al. (2019). Dota 2 with Large Scale Deep Reinforcement Learning (Cited on pages 6, 120).

Ortega, J M and W C Rheinboldt (2000). *Iterative Solution of Nonlinear Equations in Several Variables*. Society for Industrial and Applied Mathematics (Cited on pages 5, 18, 19, 22, 31, 33, 34, 49, 119, 138–140, 142).

Papini, Matteo (2020). Safe Policy Optimization (Cited on pages 8, 69, 74, 79, 123, 214).

Papini, Matteo, Damiano Binaghi, Giuseppe Canonaco, Matteo Pirotta, and Marcello Restelli (2018). Stochastic Variance-Reduced Policy Gradient. In *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. PMLR, pp. 4026–4035 (Cited on pages 7, 67, 76, 105, 112, 122).

Papini, Matteo, Matteo Pirotta, and Marcello Restelli (Oct. 2022). Smoothing policies and safe policy gradients. *Machine Learning* (Cited on pages 75, 77, 216, 217, 230, 232).

Pattathil, Sarath, Kaiqing Zhang, and Asuman Ozdaglar (2022). *Symmetric (Optimistic) Natural Policy Gradient for Multi-agent Learning with Parameter Convergence* (Cited on page 113).

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12, pp. 2825–2830 (Cited on pages 54, 183).

Penrose, Roger (1955). A Generalized Inverse for Matrices. *Mathematical Proceedings of the Cambridge Philosophical Society* 51.3, pp. 406–413 (Cited on pages 18, 173).

Peters, Jan and Stefan Schaal (Mar. 2008a). Natural Actor-Critic. *Neurocomputing* 71.7–9, pp. 1180–1190 (Cited on pages 67, 105).

— (May 2008b). Reinforcement Learning of Motor Skills with Policy Gradients. *Neural Networks* 21.4, pp. 682–697 (Cited on page 70).

Pham, Nhan, Lam Nguyen, Dzung Phan, Phuong Ha Nguyen, Marten van Dijk, and Quoc Tran-Dinh (26–28 Aug 2020). A Hybrid Stochastic Policy Gradient Algorithm for Reinforcement Learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Vol. 108. Proceedings of Machine Learning Research. PMLR, pp. 374–385 (Cited on pages 7, 67, 76, 112, 122, 217).

Pilanci, Mert and Martin J. Wainwright (2015). Randomized Sketches of Convex Programs With Sharp Guarantees. *IEEE Transactions on Information Theory* 61.9, pp. 5096–5115 (Cited on page 19).

— (2016). Iterative Hessian sketch : Fast and Accurate Solution Approximation for Constrained Least-Squares. *Journal of Machine Learning Research* 17, pp. 1–33 (Cited on page 110).

— (2017). Newton Sketch: A Near Linear-Time Optimization Algorithm with Linear-Quadratic Convergence. *SIAM Journal on Optimization* 27.1, pp. 205–245 (Cited on pages 20, 21, 23, 47, 110).

Pirotta, Matteo, Marcello Restelli, and Luca Bascetta (Sept. 2015). Policy gradient in Lipschitz Markov Decision Processes. *Machine Learning* 100.2, pp. 255–283 (Cited on page 76).

Polyak, Boris T. (1987). *Introduction to Optimization*. New York: Optimization Software (Cited on page 42).

Polyak, Boris T. and Yakov Z. Tsypkin (Jan. 1973). Pseudogradient adaptation and training algorithms. *Automation and Remote Control* 34, pp. 377–397 (Cited on page 72).

Polyak, Boris Teodorovich (1963). Gradient methods for the minimisation of functionals. *USSR Computational Mathematics and Mathematical Physics* 3.4, pp. 864–878 (Cited on pages 212, 259, 268).

Puterman, Martin L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. USA: John Wiley and Sons, Inc. (Cited on pages 6, 8, 87, 98, 101, 120, 123).

Qu, Zheng, Peter Richtárik, Martin Takáč, and Olivier Fercoq (2016). SDNA: Stochastic Dual Newton Ascent for Empirical Risk Minimization. In *Proceedings of the 33rd International Conference on Machine Learning* (Cited on page 47).

Qu, Zheng, Peter Richtárik, and Tong Zhang (2015). Quartz: Randomized dual coordinate ascent with arbitrary sampling. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1* (Cited on pages 21, 39, 41, 154, 160).

Renegar, James (2001). *A mathematical view of interior-point methods in convex optimization*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics (Cited on page 51).

Richtárik, Peter and Martin Takáč (2020). Stochastic Reformulations of Linear Systems: Algorithms and Convergence Theory. *SIAM Journal on Matrix Analysis and Applications* 41.2, pp. 487–524 (Cited on pages 22, 23, 131).

Robbins, Herbert and Sutton Monro (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics* 22.3, pp. 400–407 (Cited on pages 3, 46, 117).

Rockafellar, R. Tyrrell (1970). *Convex analysis*. Princeton Mathematical Series. Princeton, N. J.: Princeton University Press (Cited on page 309).

Rodomanov, Anton and Dmitry Kropotov (20–22 Jun 2016). A Superlinearly-Convergent Proximal Newton-type Method for the Optimization of Finite Sums. In *Proceedings of The 33rd International Conference on Machine Learning*. Vol. 48. Proceedings of Machine Learning Research. PMLR, pp. 2597–2605 (Cited on pages 4, 20, 21, 34, 35, 47, 109, 119).

Rodomanov, Anton and Yurii Nesterov (2021a). Greedy Quasi-Newton Methods with Explicit Superlinear Convergence. *SIAM Journal on Optimization* 31.1, pp. 785–811 (Cited on page 58).

— (Feb. 2021b). Rates of superlinear convergence for classical quasi-Newton methods. *Mathematical Programming* (Cited on page 58).

Rodomanov, Anton and Yurii E. Nesterov (2021c). New Results on Superlinear Convergence of Classical Quasi-Newton Methods. *J. Optim. Theory Appl.* 188.3, pp. 744–769 (Cited on page 58).

Roosta-Khorasani, Farbod and Michael W. Mahoney (2019). Sub-sampled Newton methods. *Math. Program.* 174.1-2, pp. 293–326 (Cited on pages 20, 21, 46).

Sa, Christopher De, Satyen Kale, Jason D. Lee, Ayush Sekhari, and Karthik Sridharan (2022). From Gradient Flow on Population Loss to Learning with Stochastic Gradient Descent. In *Advances in Neural Information Processing Systems* (Cited on page 112).

Sankararaman, Karthik Abinav, Soham De, Zheng Xu, W. Ronny Huang, and Tom Goldstein (13–18 Jul 2020). The Impact of Neural Network Overparameterization on Gradient Confusion and Stochastic Gradient Descent. In *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 8469–8479 (Cited on page 72).

Scherrer, Bruno (2014). Approximate Policy Iteration Schemes: A Comparison. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. ICML'14. Beijing, China: JMLR.org, pp. 1314–1322 (Cited on page 307).

Schmidt, Mark, Nicolas Le Roux, and Francis Bach (Mar. 2017). Minimizing finite sums with the stochastic average gradient. *Mathematical Programming* 162.1, pp. 83–112 (Cited on pages 5, 21, 41, 46, 53, 54, 119, 120, 154, 160, 182).

Schmidt, Mark and Nicolas Le Roux (2013). *Fast Convergence of Stochastic Gradient Descent under a Strong Growth Condition* (Cited on page 72).

Schulman, John, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz (July 2015). Trust Region Policy Optimization. In *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 1889–1897 (Cited on pages 8, 67, 87, 122, 123).

Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov (2017). *Proximal Policy Optimization Algorithms* (Cited on pages 8, 67, 87, 122, 123).

Shalev-Shwartz, Shai (20–22 Jun 2016). SDCA without Duality, Regularization, and Individual Convexity. *Proceedings of The 33rd International Conference on Machine Learning*. Proceedings of Machine Learning Research 48, pp. 747–754 (Cited on pages 21, 39, 41, 154, 160).

Shalev-Shwartz, Shai and Shai Ben-David (2014). *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, pp. I–XVI, 1–397 (Cited on pages 101, 292).

Shalev-Shwartz, Shai, Shaked Shammah, and Amnon Shashua (2016). *Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving* (Cited on pages 6, 120).

Shalev-Shwartz, Shai and Tong Zhang (Feb. 2013). Stochastic Dual Coordinate Ascent Methods for Regularized Loss. *Journal of Machine Learning Research* 14.1, pp. 567–599 (Cited on pages 21, 39).

Shani, Lior, Yonathan Efroni, and Shie Mannor (2020). Adaptive Trust Region Policy Optimization: Global Convergence and Faster Rates for Regularized MDPs. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 5668–5675 (Cited on pages 67, 87, 93, 266).

Shen, Zebang, Alejandro Ribeiro, Hamed Hassani, Hui Qian, and Chao Mi (Sept. 2019). Hessian Aided Policy Gradient. In *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 5729–5738 (Cited on pages 7, 67, 76, 122, 216, 217).

Silver, David, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis (Oct. 2017). Mastering the game of Go without human knowledge. *Nature* 550.7676, pp. 354–359 (Cited on pages 6, 120).

Stich, Sebastian U. (2019). *Unified Optimal Analysis of the (Stochastic) Gradient Method* (Cited on page 222).

Strohmer, Thomas and Roman Vershynin (2009). A Randomized Kaczmarz Algorithm with Exponential Convergence. *Journal of Fourier Analysis and Applications* 15.2, pp. 262–278 (Cited on page 34).

Sutton, Richard, Hamid Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvari, and Eric Wiewiora (June 2009). Fast Gradient-Descent Methods for Temporal-Difference Learning with Linear Function Approximation. In *Proceedings of the 26th International Conference on Machine Learning*. Montreal: Omnipress, pp. 993–1000 (Cited on page 101).

Sutton, Richard S, David A. McAllester, Satinder P. Singh, and Yishay Mansour (2000). Policy Gradient Methods for Reinforcement Learning with Function Approximation. In *Advances in Neural Information Processing Systems 12*. MIT Press, pp. 1057–1063 (Cited on pages 7, 9, 67, 70, 87, 90, 91, 122, 124, 214, 270, 271).

Sutton, Richard S. and Andrew G. Barto (2018). *Reinforcement Learning: An Introduction*. Second. The MIT Press (Cited on page 87).

Telgarsky, Matus (Feb. 2022). Stochastic linear optimization never overfits with quadratically-bounded losses on general data. In *Proceedings of Thirty Fifth Conference on Learning Theory*. Vol. 178. Proceedings of Machine Learning Research. PMLR, pp. 5453–5488 (Cited on page 106).

Tomar, Manan, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh (2022). Mirror Descent Policy Optimization. In *International Conference on Learning Representations* (Cited on pages 67, 87).

Torres, G. L. and V. H. Quintana (Aug. 2000). Optimal power flow by a nonlinear complementarity method. *IEEE Transactions on Power Systems* 15.3, pp. 1028–1033 (Cited on page 18).

Tran-Dinh, Quoc, Nhan H. Pham, Dzung T. Phan, and Lam M. Nguyen (Jan. 2021). A hybrid stochastic optimization framework for composite nonconvex optimization. *Mathematical Programming* (Cited on pages 7, 112, 122).

Tsitsiklis, John and Benjamin Van Roy (1996). Analysis of Temporal-Diffference Learning with Function Approximation. In *Advances in Neural Information Processing Systems*. Vol. 9. MIT Press (Cited on page 101).

Vaswani, Sharan, Francis Bach, and Mark Schmidt (16–18 Apr 2019a). Fast and Faster Convergence of SGD for Over-Parameterized Models and an Accelerated Perceptron. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*. Vol. 89. Proceedings of Machine Learning Research. PMLR, pp. 1195–1204 (Cited on pages 4, 21, 26, 28, 42, 46, 72, 119).

Vaswani, Sharan, Olivier Bachem, Simone Totaro, Robert Müller, Shivam Garg, Matthieu Geist, Marlos C. Machado, Pablo Samuel Castro, and Nicolas Le Roux (28–30 Mar 2022). A general class of surrogate functions for stable and efficient reinforcement learning. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. Vol. 151. Proceedings of Machine Learning Research. PMLR, pp. 8619–8649 (Cited on pages 67, 87).

Vaswani, Sharan, Aaron Mishkin, Issam Laradji, Mark Schmidt, Gauthier Gidel, and Simon Lacoste-Julien (2019b). Painless Stochastic Gradient: Interpolation, Line-Search, and Convergence Rates. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., pp. 3732–3745 (Cited on page 162).

Vieillard, Nino, Tadashi Kozuno, Bruno Scherrer, Olivier Pietquin, Remi Munos, and Matthieu Geist (2020). Leverage the Average: an Analysis of KL Regularization in Reinforcement Learning. In *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., pp. 12163–12174 (Cited on page 93).

Vinyals, Oriol, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, et al. (Nov. 2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575.7782, pp. 350–354 (Cited on pages 6, 120).

Wang, De, Danesh Irani, and Calton Pu (Oct. 2012). Evolutionary Study of Web Spam: Webb Spam Corpus 2011 versus Webb Spam Corpus 2006. In *Proc. of 8th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2012)* (Cited on pages 40, 54).

Wang, Lingxiao, Qi Cai, Zhuoran Yang, and Zhaoran Wang (2020). Neural Policy Gradient Methods: Global Optimality and Rates of Convergence. In *International Conference on Learning Representations* (Cited on pages 83, 95, 96, 111, 267, 269).

Wang, Qifeng, Weiguo Li, Wendi Bao, and Xingqi Gao (2022). Nonlinear Kaczmarz algorithms and their convergence. *Journal of Computational and Applied Mathematics* 399, p. 113720 (Cited on pages 4, 43, 109, 119).

Wang, Xiao, Shiqian Ma, Donald Goldfarb, and Wei Liu (2017). Stochastic Quasi-Newton Methods for Nonconvex Stochastic Optimization. *SIAM Journal on Optimization* 27.2, pp. 927–956 (Cited on page 105).

Wang, Zhe, Kaiyi Ji, Yi Zhou, Yingbin Liang, and Vahid Tarokh (2019). SpiderBoost and Momentum: Faster Variance Reduction Algorithms. In *Advances in Neural Information Processing Systems*. Vol. 32, pp. 2406–2416 (Cited on pages 7, 112, 122).

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* 8, pp. 229–256 (Cited on pages 7, 9, 67, 70, 87, 122, 124).

Williams, Ronald J. and Jing Peng (1991). Function Optimization using Connectionist Reinforcement Learning Algorithms. *Connection Science* 3.3, pp. 241–268 (Cited on page 80).

Woodruff, David P (2014). Sketching as a tool for numerical linear algebra. *arXiv preprint arXiv:1411.4357* (Cited on page 23).

Wright, Stephen J. and Jorge Nocedal (2006). Interior-Point Methods for Nonlinear Programming, pp. 563–597 (Cited on page 22).

Xiao, Lin (2022). On the Convergence Rates of Policy Gradient Methods. *Journal of Machine Learning Research* 23.282, pp. 1–36 (Cited on pages xvi, 7–10, 69, 84, 87, 88, 93, 95, 98, 101, 105, 107, 111, 122–125, 213, 264–267, 281, 305–309).

Xiong, Huaqing, Tengyu Xu, Yingbin Liang, and Wei Zhang (May 2021). Non-asymptotic Convergence of Adam-type Reinforcement Learning Algorithms under Markovian Sampling. *Proceedings of the AAAI Conference on Artificial Intelligence* 35.12, pp. 10460–10468 (Cited on pages 8, 67, 74, 79, 123, 214).

Xu, Pan, Felicia Gao, and Quanquan Gu (22–25 Jul 2020a). An Improved Convergence Analysis of Stochastic Variance-Reduced Policy Gradient. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*. Vol. 115. Proceedings of Machine Learning Research. PMLR, pp. 541–551 (Cited on pages 76, 216).

— (2020b). Sample Efficient Policy Gradient Methods with Recursive Variance Reduction. In *International Conference on Learning Representations* (Cited on pages 7, 67, 76, 77, 112, 122, 216, 233, 237).

Xu, Tengyu, Zhe Wang, and Yingbin Liang (2020c). Improving Sample Complexity Bounds for (Natural) Actor-Critic Algorithms. In *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., pp. 4358–4369 (Cited on pages 111, 267, 269).

Yang, Lin and Mengdi Wang (2019). Sample-Optimal Parametric Q-Learning Using Linearly Additive Features. In *Proceedings of the 36th International Conference on Machine Learning*. PMLR, pp. 6995–7004 (Cited on page 96).

— (13–18 Jul 2020). Reinforcement Learning in Feature Space: Matrix Bandit, Kernels, and Regret Bound. In *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, pp. 10746–10756 (Cited on page 96).

Yang, Long, Yu Zhang, Gang Zheng, Qian Zheng, Pengfei Li, Jianhang Huang, and Gang Pan (June 2022). Policy Optimization with Stochastic Mirror Descent. *Proceedings of the AAAI Conference on Artificial Intelligence* 36.8, pp. 8823–8831 (Cited on pages 7, 67, 112, 122).

Yang, Long, Qian Zheng, and Gang Pan (May 2021). Sample Complexity of Policy Gradient Finding Second-Order Stationary Points. *Proceedings of the AAAI Conference on Artificial Intelligence* 35.12, pp. 10630–10638 (Cited on page 112).

Yang, Zhuoran, Yongxin Chen, Mingyi Hong, and Zhaoran Wang (2019). Provably Global Convergence of Actor-Critic: A Case for Linear Quadratic Regulator with Ergodic Cost. In *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc. (Cited on page 111).

Ying, Donghao, Mengzi Amy Guo, Yuhao Ding, Javad Lavaei, and Zuo-Jun Max Shen (2022). *Policy-based Primal-Dual Methods for Convex Constrained Markov Decision Processes* (Cited on page 113).

Yuan, Huizhuo, Xiangru Lian, Ji Liu, and Yuren Zhou (2020). *Stochastic Recursive Momentum for Policy Gradient Methods* (Cited on pages 7, 67, 76, 77, 112, 122, 216, 237).

Yuan, Rui, Simon Shaolei Du, Robert M. Gower, Alessandro Lazaric, and Lin Xiao (2023). Linear Convergence of Natural Policy Gradient Methods with Log-Linear Policies. In *International Conference on Learning Representations* (Cited on page 86).

Yuan, Rui, Robert M. Gower, and Alessandro Lazaric (28–30 Mar 2022a). A general sample complexity analysis of vanilla policy gradient. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. Vol. 151. Proceedings of Machine Learning Research. PMLR, pp. 3332–3380 (Cited on pages 65, 105, 111, 268).

Yuan, Rui, Alessandro Lazaric, and Robert M. Gower (2022b). Sketched Newton–Raphson. *SIAM Journal on Optimization* 32.3, pp. 1555–1583 (Cited on pages 16, 48, 56–58, 189).

Yuan, Ya-Xiang (2011). Recent advances in numerical methods for nonlinear equations and nonlinear least squares. *Numerical Algebra, Control and Optimization* 1.1, pp. 15–34 (Cited on page 22).

Zanette, Andrea, Ching-An Cheng, and Alekh Agarwal (15–19 Aug 2021). Cautiously Optimistic Policy Optimization and Exploration with Linear Function Approximation. In *Proceedings of Thirty Fourth Conference on Learning Theory*. Vol. 134. Proceedings of Machine Learning Research. PMLR, pp. 4473–4525 (Cited on pages 111, 267, 269).

Zeng, Wen-Jun and Jieping Ye (2020). *Successive Projection for Solving Systems of Nonlinear Equations/Inequalities* (Cited on page 43).

Zhan, Wenhao, Shicong Cen, Baihe Huang, Yuxin Chen, Jason D. Lee, and Yuejie Chi (2021). *Policy Mirror Descent for Regularized Reinforcement Learning: A Generalized Framework with Linear Convergence* (Cited on pages 87, 266).

Zhang, Junyu, Alec Koppel, Amrit Singh Bedi, Csaba Szepesvari, and Mengdi Wang (2020a). Variational Policy Gradient Method for Reinforcement Learning with General Utilities. In *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., pp. 4572–4583 (Cited on pages 8, 67, 69, 95, 111, 123, 213, 214, 268).

Zhang, Junyu, Chengzhuo Ni, Zheng Yu, Csaba Szepesvari, and Mengdi Wang (2021a). On the Convergence and Sample Efficiency of Variance-Reduced Policy Gradient Method. In *Advances in Neural Information Processing Systems* (Cited on pages 67, 76, 111, 214, 268).

Zhang, Junzi, Jongho Kim, Brendan O'Donoghue, and Stephen Boyd (May 2021b). Sample Efficient Reinforcement Learning with REINFORCE. *Proceedings of the AAAI Conference on Artificial Intelligence* 35.12, pp. 10887–10895 (Cited on pages 8, 69, 79, 82, 123, 215, 251).

Zhang, Kaiqing, Alec Koppel, Hao Zhu, and Tamer Başar (2020b). Global Convergence of Policy Gradient Methods to (Almost) Locally Optimal Policies. *SIAM Journal on Control and Optimization* 58.6, pp. 3586–3612 (Cited on pages 8, 67, 74, 76, 79, 123, 214).

Zhang, Runyu, Qinghua Liu, Huan Wang, Caiming Xiong, Na Li, and Yu Bai (2022a). *Policy Optimization for Markov Games: Unified Framework and Faster Convergence* (Cited on page 113).

Zhang, Runyu, Jincheng Mei, Bo Dai, Dale Schuurmans, and Na Li (2022b). On the Global Convergence Rates of Decentralized Softmax Gradient Play in Markov Potential Games. In *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (Cited on page 112).

Zhang, Yanjun and Hanyu Li (2022). *Greedy capped nonlinear Kaczmarz methods* (Cited on page 43).

Zhang, Yanjun, Hanyu Li, and Ling Tang (2022c). *Greedy randomized sampling nonlinear Kaczmarz methods* (Cited on page 43).

Zhou, Dongruo and Quanquan Gu (Sept. 2019). Lower Bounds for Smooth Nonconvex Finite-Sum Optimization. In *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 7574–7583 (Cited on page 112).

Zhou, Dongruo, Pan Xu, and Quanquan Gu (Oct. 2018). Stochastic Variance-Reduced Cubic Regularized Newton Methods. In *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80, pp. 5990–5999 (Cited on pages 20, 21).

— (2020). Stochastic Nested Variance Reduction for Nonconvex Optimization. *Journal of Machine Learning Research* 21.103, pp. 1–63 (Cited on pages 7, 112, 122).

Zhou, Weijun (2013). On the convergence of the modified Levenberg–Marquardt method with a nonmonotone second order Armijo type line search. *Journal of Computational and Applied Mathematics* (Cited on page 22).

Zhou, Weijun and Xiaojun Chen (2010). Global Convergence of a New Hybrid Gauss–Newton Structured BFGS Method for Nonlinear Least Squares Problems. *SIAM Journal on Optimization* 20.5, pp. 2422–2441 (Cited on page 22).

Zhou, Yi, Yingbin Liang, and Huishuai Zhang (Jan. 2022). Understanding generalization error of SGD in nonconvex optimization. *Machine Learning* 111.1, pp. 345–375 (Cited on page 112).

Zhou, Yi, Junjie Yang, Huishuai Zhang, Yingbin Liang, and Vahid Tarokh (2019). SGD Converges to Global Minimum in Deep Learning via Star-convex Path. In *International Conference on Learning Representations* (Cited on pages 22, 27).

ECOLE
DOCTORALE
DE MATHEMATIQUES
HADAMARD

**Titre :** Méthodes du second ordre stochastiques et analyse de temps fini des méthodes de policy-gradient

**Mots clés :** Optimisation, méthodes du second ordre stochastiques, apprentissage par renforcement, méthodes de policy-gradient

**Résumé :** Pour résoudre les problèmes de l'apprentissage automatique à grande échelle, les méthodes de premier ordre telles que la descente du gradient stochastique et l'ADAM sont les méthodes de choix en raison de leur coût pas cher par itération. Le problème des méthodes du premier ordre est qu'elles peuvent nécessiter un réglage lourd des paramètres et/ou une connaissance des paramètres du problème. Il existe aujourd'hui un effort considérable pour développer des méthodes du second ordre stochastiques efficaces afin de résoudre des problèmes de l'apprentissage automatique à grande échelle. La motivation est qu'elles demandent moins de réglage des paramètres et qu'elles convergent pour une plus grande variété de modèles et de datasets. Dans la première partie de la thèse, nous avons présenté une approche de principe pour désigner des méthodes de Newton stochastiques à fin de résoudre à la fois des équations non linéaires et des problèmes d'optimisation d'une manière efficace. Notre approche comporte deux étapes. Premièrement, nous pouvons réécrire les équations non linéaires ou le problème d'optimisation sous forme d'équations non linéaires souhaitées. Ensuite, nous appliquons de nouvelles méthodes du second ordre stochastiques pour résoudre ce système d'équations non linéaires. Grâce à notre approche générale, nous présentons de nombreux nouveaux algorithmes spécifiques du second ordre qui peuvent résoudre efficacement les problèmes de l'apprentissage automatique à grande échelle sans nécessiter de connaissance du problème ni de réglage des paramètres. Dans la deuxième partie de la thèse, nous nous concentrons sur les algorithmes d'optimisation appliqués à un domaine spécifique : l'apprentissage par renforcement (RL). Cette partie est indépendante de la première partie de la thèse. Pour atteindre de telles performances dans les problèmes de RL, le policy-gradient (PG) et sa variante, le policy-gradient naturel (NPG), sont les fondements de plusieurs algorithmes de l'état de l'art (par exemple, TRPO et PPO) utilisés dans le RL profond. Malgré le succès empirique des méthodes de RL et de PG, une compréhension théorique solide du PG original a longtemps fait défaut. En utilisant la structure du RL du problème et des techniques modernes de preuve d'optimisation, nous obtenons nouvelles analyses en temps fini de la PG et de la NPG. Grâce à notre analyse, nous apportons également de nouvelles perspectives aux méthodes avec de meilleurs choix d'hyperparamètres.

**Title :** Stochastic Second Order Methods and Finite Time Analysis of Policy Gradient Methods

**Keywords :** Optimization, stochastic second-order methods, reinforcement learning, policy gradient methods

**Abstract :** To solve large scale machine learning problems, first-order methods such as stochastic gradient descent and ADAM are the methods of choice because of their low cost per iteration. The issue with first order methods is that they can require extensive parameter tuning, and/or knowledge of the parameters of the problem. There is now a concerted effort to develop efficient stochastic second order methods to solve large scale machine learning problems. The motivation is that they require less parameter tuning and converge for wider variety of models and datasets. In the first part of the thesis, we presented a principled approach for designing stochastic Newton methods for solving both nonlinear equations and optimization problems in an efficient manner. Our approach has two steps. First, we can re-write the nonlinear equations or the optimization problem as desired nonlinear equations. Second, we apply new stochastic second order methods to solve this system of nonlinear equations. Through our general approach, we showcase many specific new second-order algorithms that can solve the large machine learning problems efficiently without requiring knowledge of the problem nor parameter tuning. In the second part of the thesis, we then focus on optimization algorithms applied in a specific domain: reinforcement learning (RL). This part is independent to the first part of the thesis. To achieve such high performance of RL problems, policy gradient (PG) and its variant, natural policy gradient (NPG), are the foundations of the several state of the art algorithms (e.g., TRPO and PPO) used in deep RL. In spite of the empirical success of RL and PG methods, a solid theoretical understanding of even the "vanilla" PG has long been elusive. By leveraging the RL structure of the problem together with modern optimization proof techniques, we derive new finite time analysis of both PG and NPG. Through our analysis, we also bring new insights to the methods with better hyperparameter choices.

**Institut Polytechnique de Paris**
91120 Palaiseau, France