Non-asymptotic Analysis of Biased Adaptive Stochastic Approximation

Sobihan Surendran, Adeline Fermanian, Antoine Godichon-Baggioni, Sylvain Le Corff

Non-asymptotic Analysis of Biased Adaptive Stochastic Approximation

Sobihan Surendran[*†], Adeline Fermanian[†], Antoine Godichon-Baggioni[*] and Sylvain Le Corff[*]

Abstract

Stochastic Gradient Descent (SGD) with adaptive steps is now widely used for training deep neural networks. Most theoretical results assume access to unbiased gradient estimators, which is not the case in several recent deep learning and reinforcement learning applications that use Monte Carlo methods. This paper provides a comprehensive non-asymptotic analysis of SGD with biased gradients and adaptive steps for non-convex smooth functions. Our study incorporates time-dependent bias and emphasizes the importance of controlling the bias of the gradient estimator. In particular, we establish that Adagrad, RMSProp, and Adam with biased gradients converge to critical points for smooth non-convex functions at a rate similar to existing results in the literature for the unbiased case. Finally, we provide experimental results using Variational Autoenconders (VAE) that illustrate our convergence results and show how the effect of bias can be reduced by appropriate hyperparameter tuning.

Keywords: Stochastic Optimization; Biased Stochastic Approximation; Monte Carlo Methods; Variational Autoenconders; Adam

1 Introduction

Stochastic Gradient Descent (SGD) algorithms are standard methods to train statistical models based on deep architectures. Consider a general optimization problem:

$$\theta_* \in \operatorname*{argmin}_{\theta \in \mathbb{R}^d} V(\theta) \,, \tag{1}$$

where V is the objective function. Then, gradient descent methods produce a sequence of parameter estimates as follows: $\theta_0 \in \mathbb{R}^d$ and for all $n \geq 1$,

$$\theta_{n+1} = \theta_n - \gamma_{n+1} \nabla V(\theta_n) \,,$$

where ∇V denotes the gradient of V and for all $n \geq 1$, $\gamma_n > 0$ is the learning rate. In many cases, it is not possible to compute the exact gradient of the objective function, hence the introduction of vanilla Stochastic Gradient Descent, defined for all $n \geq 1$ by:

$$\theta_{n+1} = \theta_n - \gamma_{n+1} \widehat{\nabla V}(\theta_n) \,,$$

where $\widehat{\nabla V}(\theta_n)$ is an estimator of $\nabla V(\theta_n)$. In deep learning, stochasticity emerges with the use of mini-batches, since it is not feasible to compute gradients based on the entire dataset.

While these algorithms have been extensively studied, both theoretically and practically (see, e.g., Bottou et al., 2018), many questions remain open. In particular, most results are based on the case where the estimator $\widehat{\nabla V}$ is unbiased. Although this assumption is valid in the case of vanilla SGD, it breaks down in many common applications. For example, zeroth-order methods used to optimize black-box functions (Nesterov and Spokoiny, 2017) in generative adversarial networks (Moosavi-Dezfooli et al., 2017; Chen et al., 2017) have access only to noisy biased realizations of the objective functions.

Furthermore, in reinforcement learning algorithms such as Q-learning (Jaakkola et al., 1993), policy gradient (Baxter and Bartlett, 2001), and temporal difference learning (Bhandari et al., 2018; Lakshminarayanan and Szepesvari, 2018; Dalal et al., 2018), gradient estimators are often obtained using a Markov chain with state-dependent transition probability. These estimators are then biased (Sun et al., 2018; Doan et al., 2020). Other examples of biased gradients can be found in the field of generative modeling with Markov Chain Monte Carlo (MCMC) and Sequential Monte Carlo (SMC) (Gloaguen et al., 2022; Cardoso et al., 2023). In particular, the Importance Weighted Autoencoder

[*]Laboratoire de Probabilités, Statistique et Modélisation (LPSM), Sorbonne Université, 75005 Paris, France.
[†]LOPF, Califrais' Machine Learning Lab, Paris, France

(IWAE) proposed by Burda et al. (2016), which is a variant of the standard Variational Autoencoder (VAE) (Kingma and Welling, 2014), yields biased estimators.

In practical applications, vanilla SGD exhibits difficulties to calibrate the step sequences. Therefore, modern variants of SGD employ adaptive steps that use past stochastic gradients or Hessians to avoid saddle points and deal with ill-conditioned problems. The idea of adaptive steps was first proposed in the online learning literature by Auer et al. (2002) and later adopted in stochastic optimization, with the Adagrad algorithm of Duchi et al. (2011). Adagrad aims to normalize the gradient by introducing information about the square root of the inverse of the covariance of the gradient.

We give non-asymptotic convergence guarantees for modern variants of SGD where both the estimators are biased and the steps are adaptive. To our knowledge, existing results consider either adaptive steps but unbiased estimators or biased estimators and non-adaptive steps. Indeed, many standard analyses for SGD (Moulines and Bach, 2011) and SGD with adaptive steps (Duchi et al., 2011) require unbiased gradients to obtain convergence results. More recently, convergence results have been obtained for SGD with biased gradients (Tadić and Doucet, 2011; Karimi et al., 2019; Ajalloeian and Stich, 2020) but non-adaptive steps.

More precisely, our contributions are summarized as follows.

- We provide convergence guarantees for the Biased Adaptive Stochastic Approximation framework, under weak assumptions on the bias and "Expected smoothness". To the best of our knowledge, these are the first convergence results in this framework to incorporate adaptive steps in the biased SA.
- In particular, we establish that Adagrad, RMSProp, and Adam with a biased gradient converge to a critical point for non-convex smooth functions with a convergence rate of $O(\log n/\sqrt{n} + b_n)$, where b_n is related to the bias at iteration n. However, we achieve an improved linear convergence rate with the Polyak-Łojasiewicz (PL) condition.
- We provide a bound on the bias of IWAE, and illustrate our convergence results using this bound in the experimental results and show how the effect of bias can be reduced by appropriate hyperparameter tuning.
- Finally, we present applications with biased gradients where the bias can be controlled and provide the implications of our theorems in these applications, especially Stochastic Bilevel Optimization and Conditional Stochastic Optimization but also self-normalized importance sampling estimators or coordinate sampling.

Organization of the paper. In Section 2, we introduce the setting of the paper and relevant related works. In Section 3, we present the Adaptive Stochastic Approximation framework and the main assumptions for theoretical results. In Section 4, we propose convergence rates for the risk in the PL condition case and the squared norm of gradients without the PL condition in the context of Biased Adaptive Stochastic Approximation. Finally, we extend the analysis to Adagrad, RMSProp, and Adam with biased gradients. We illustrate our results using VAE in Section 5. All proofs are postponed to the appendix.

2 Setting and Related Works

Stochastic Approximation. Stochastic Approximation (SA) methods go far beyond SGD. They consist of sequential algorithms designed to find the zeros of a function when only noisy observations are available. Indeed, Robbins and Monro (1951) introduced the Stochastic Approximation algorithm as an iterative recursive algorithm to solve the following integration equation:

$$h(\theta) = \mathbb{E}_\pi \left[H_\theta(X) \right] = \int_{\mathsf{X}} H_\theta(x)\pi(x)\mathrm{d}x = 0 \ , \tag{2}$$

where h is the mean field function, X is a random variable taking values in a measurable space $(\mathsf{X}, \mathcal{X})$, and $\mathbb{E}_\pi$ is the expectation under the distribution π. In this context, H_θ can be any arbitrary function. If $H_\theta(X)$ is an unbiased estimator of the gradient of the objective function, then $h(\theta) = \nabla V(\theta)$. As a result, the minimization problem (1) is then equivalent to solving problem (2), and we can note that SGD is a specific instance of Stochastic Approximation. Stochastic Approximation methods are then defined as follows:

$$\theta_{n+1} = \theta_n - \gamma_{n+1} H_{\theta_n}\left(X_{n+1}\right) \ ,$$

where the term $H_{\theta_n}(X_{n+1})$ is the n-th stochastic update, also known as the drift term, and is a potentially biased estimator of $\nabla V(\theta_n)$. It depends on a random variable X_{n+1} which takes its values in $(\mathsf{X}, \mathcal{X})$. In machine learning, V typically represents theoretical risk, θ denotes model parameters, and X_{n+1} stands for the data.

Adaptive Stochastic Gradient Descent. SGD can be traced back to Robbins and Monro (1951), and its averaged counterpart was proposed by Polyak and Juditsky (1992). The non-asymptotic results of SGD in both convex and strong convex cases can be found in Moulines and Bach (2011). Ghadimi and Lan (2013) prove the convergence of a random iterate of SGD for nonconvex smooth functions, which was already suggested by the results of Bottou (1991). They show that SGD with constant or decreasing stepsize $\gamma_k = 1/\sqrt{k}$ converges to a stationary point of a non-convex smooth function V at a rate of $O(1/\sqrt{n})$ where n is the number of iterations.

Most adaptive first-order methods, such as Adam (Kingma and Ba, 2015), Adadelta (Zeiler, 2012), RMSProp (Tieleman et al., 2012), and NADA (Dozat, 2016), are based on the blueprint provided by the Adagrad family of algorithms. The first known work on adaptive steps for non-convex stochastic optimization, in the asymptotic case, was presented by Kresoja et al. (2017). Ward et al. (2020) proved that Adagrad converges to a critical point for non-convex objectives at a rate of $O(\log n/\sqrt{n})$ when using a scalar adaptive step. In addition, Zou et al. (2018) extended this proof to multidimensional settings. More recently, Défossez et al. (2020) focused on the convergence rates for Adagrad and Adam. Furthermore, several modified versions of Adam have been proposed, such as AMSGRAD (Zaheer et al., 2018) and YOGI (Reddi et al., 2018).

Biased Stochastic Approximation. The asymptotic results of Biased Stochastic Approximation have been studied by Tadić and Doucet (2011). The non-asymptotic analysis of Biased Stochastic Approximation can be found in the reinforcement learning literature, especially in the context of temporal difference (TD) learning, as explored by Bhandari et al. (2018); Lakshminarayanan and Szepesvari (2018); Dalal et al. (2018). The case of non-convex smooth functions has been studied by Karimi et al. (2019). The authors establish convergence results for the mean field function at a rate of $O(\log n/\sqrt{n} + b)$, where b corresponds to the bias and n to the number of iterations. For strongly convex functions, the convergence of SGD with biased gradients can be found in Ajalloeian and Stich (2020), who consider a constant step size. This analysis applies specifically to the case of Martingale noise.

In Khaled and Richtárik (2020); Demidovich et al. (2024), a novel assumption known as "Expected smoothness" is introduced, which is the weakest assumption compared to the existing literature in the biased SGD setting. The authors provide convergence results in the case of non-convex smooth functions. Convergence results with assumptions on the control of bias and MSE can be found Liu and Tajbakhsh (2023); Dieuleveut et al. (2023). Applications of biased gradients can be found in bilevel optimization Ji et al. (2021); Grazzi et al. (2023); Huang et al. (2021) and conditional stochastic optimization Hu et al. (2020, 2021b). Moreover, biased gradients are also used in various other applications (Hu et al., 2021a; Li and Wai, 2022; Beznosikov et al., 2023; Liu and Tajbakhsh, 2023). Finally, Alacaoglu and Lyu (2023) studied convergence results of biased gradients with Adagrad in the Markov Chain case, focusing on the norm of the gradient of the Moreau envelope while assuming the boundedness of the objective function. Our analysis provides non-asymptotic results in a more general setting, for a wide variety of objective functions and adaptive algorithms and treating both the Martingale and Markov chain cases.

3 Adaptive Stochastic Approximation

3.1 Framework

Consider the optimization problem (1) where the objective function V is assumed to be differentiable. In this paper, we focus on the following Stochastic Approximation (SA) algorithm with adaptive steps: $\theta_0 \in \mathbb{R}^d$ and for all $n \geq 0$,

$$\theta_{n+1} = \theta_n - \gamma_{n+1} A_n H_{\theta_n}(X_{n+1}) \ , \tag{3}$$

where $\gamma_{n+1} > 0$ and A_n is a sequence of symmetric and positive definite matrices. In a context of biased gradient estimates, choosing

$$A_n = \left[\delta I_d + \Big(\frac{1}{n+1}\sum_{k=0}^{n} H_{\theta_k}(X_{k+1}) H_{\theta_k}(X_{k+1})^\top\Big)\right]^{-1/2}$$

can be assimilated to the full Adagrad algorithm (Duchi et al., 2011). However, computing the square root of the inverse becomes expensive in high dimensions, so in practice, Adagrad is often used with diagonal matrices. This approach has been shown to be particularly effective in sparse optimization settings. Denoting by $\mathrm{Diag}(A)$ the matrix

formed with the diagonal terms of A and setting all other terms to 0, Adagrad with diagonal matrices is defined in our context as:

$$A_n = \left[\delta I_d + \mathrm{Diag}(\bar{H}_n(X_{0:n+1}, \theta_{0:n}))\right]^{-1/2} , \tag{4}$$

where

$$\bar{H}_n(X_{0:n+1}, \theta_{0:n}) = \frac{1}{n+1} \sum_{k=0}^{n} H_{\theta_k}(X_{k+1}) H_{\theta_k}(X_{k+1})^\top .$$

In RMSProp (Tieleman et al., 2012), $\bar{H}_n(X_{0:n+1}, \theta_{0:n})$ in (4) is an exponential moving average of the past squared gradients. It is defined by:

$$(1-\rho) \sum_{k=0}^{n} \rho^{n-k} H_{\theta_k}(X_{k+1}) H_{\theta_k}(X_{k+1})^\top ,$$

where ρ is the moving average parameter. Furthermore, when A_n is a recursive estimate of the inverse Hessian, it corresponds to the Stochastic Newton algorithm (Boyer and Godichon-Baggioni, 2023).

3.2 Assumptions

Consider the following assumptions necessary to establish our theoretical results.

H1 There exists a constant $\mu > 0$ such that for all $\theta \in \mathbb{R}^d$,

$$2\mu \left(V(\theta) - V\left(\theta^*\right)\right) \leq \|\nabla V(\theta)\|^2 .$$

H1 corresponds to the Polyak-Łojasiewicz condition, which is weaker than strong convexity and remains satisfied even when the function is non-convex. The PL condition has been extensively studied theoretically (Karimi et al., 2016) and has been verified in many applications, such as over-parameterized deep networks (Du et al., 2019) and Linear Quadratic Regulator models (Fazel et al., 2018).

H2 The objective function V is L-smooth. For all $(\theta, \theta') \in \mathbb{R}^d \times \mathbb{R}^d$,

$$\left\|\nabla V(\theta) - \nabla V\left(\theta'\right)\right\| \leq L \left\|\theta - \theta'\right\| .$$

This assumption is crucial to obtain our convergence rate and is very common (see, e.g., Moulines and Bach, 2011; Bottou et al., 2018). Under this assumption, for all $(\theta, \theta') \in \mathbb{R}^d \times \mathbb{R}^d$,

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

H3 (i) There exist two non-increasing positive sequences $(\lambda_n)_{n\geq 1}$ and $(r_n)_{n\geq 1}$ such that: $\mathbb{E}[\nabla V(\theta_n)^T A_n H_{\theta_n}(X_{n+1})] \geq \lambda_{n+1} \left(\mathbb{E}\left[\|\nabla V(\theta_n)\|^2\right] - r_{n+1}\right)$.

(ii) Expected smoothness: there exist a non-increasing non-negative sequence $(\sigma_n^2)_{n\geq 1}$, and positive constants $\tilde{\sigma}_1, \tilde{\sigma}_2$ such that: $\mathbb{E}[\|H_{\theta_n}(X_{n+1})\|^2] \leq \sigma_n^2 + \tilde{\sigma}_1 \mathbb{E}[\|\nabla V(\theta_n)\|^2] + \tilde{\sigma}_2 \mathbb{E}[V(\theta_n) - V(\theta^*)]$.

In this assumption, for all $n \in \mathbb{N}$, r_{n+1} represents the bias, and λ_{n+1} may depend on the minimum eigenvalue of A_n. In Demidovich et al. (2024, Theorem 2), it has been demonstrated that this assumption is weaker than the alternatives used in the literature on biased SGD setting. We have adapted these assumptions with adaptive steps. It is important to note that the first point of Assumption 3 depends on the application (objective function V) and on the adaptive algorithm (matrix A_n) that we want to use. The purpose of this assumption is to provide a more general framework that covers all possible applications and adaptive algorithms. In the biased SGD setting, if the bias term $\|\mathbb{E}[H_{\theta_n}(X_{n+1}) \mid \mathcal{F}_n] - \nabla V(\theta_n)\|$ is bounded by $\tilde{b}_{n+1}$, we can easily verify the first point of H3 by considering $\lambda_{n+1} = 1/2$ and $r_{n+1} = \tilde{b}_{n+1}^2/2$. We show in Section 4.3 that this assumption is also verified in algorithms such as Adagrad and RMSProp. This assumption can be easily verified in many applications such as self-normalized importance sampling (Agapiou et al., 2017), sequential Monte Carlo (Del Moral et al., 2010; Olsson and Westerborn, 2017), zeroth order methods (Nesterov and Spokoiny, 2017), bilevel optimization (Ji et al., 2021), and conditional stochastic optimization (Hu et al., 2020). The second point of H3 is a weaker assumption compared to bounding the variance of the noise term. Applications where we can verify these assumptions are discussed in Appendix D.

We consider also an additional assumption on A_n. Let $\|A\|$ be the spectral norm of a matrix A.

H4 There exists $(\beta_n)_{n\geq 1}$ such that for all $n \in \mathbb{N}$,

$$\|A_n\| := \lambda_{\max}(A_n) \leq \beta_{n+1}.$$

In our setting, since A_n is assumed to be a symmetric matrix, the spectral norm is equal to the largest eigenvalue. H4 plays a crucial role, as the estimates may diverge when this assumption is not satisfied. Given a sequence $\{\beta_n\}_{n\in\mathbb{N}}$, one way to ensure that H4 is satisfied is to replace the random matrices A_n with

$$\tilde{A}_n = \frac{\min\{\|A_n\|, \beta_{n+1}\}}{\|A_n\|} A_n. \tag{5}$$

It is then clear that $\|\tilde{A}_n\| \leq \beta_{n+1}$. Furthermore, in most cases, especially for Adagrad, RMSProp and Stochastic Newton algorithm, control of $\lambda_{\max}(A_n)$ in H4 is satisfied. For example, in Adagrad and RMSProp, in (4), we have $\lambda_{\max}(A_n) \leq \delta^{-1/2}$. Instead of using a constant regularization term, we can also choose a decreasing regularization term, such as $\delta = \beta_{n+1}^{-2}$.

4 Convergence Results

4.1 Convergence under PL-condition

In this section, we study the convergence rate of SGD with biased gradients and adaptive steps in the convex case. We give below a simplified version of the bound we obtained on the risk and refer to Theorem A.2 in the appendix for a formal statement with explicit constants.

Theorem 4.1. *Let $\theta_n \in \mathbb{R}^d$ be the n-th iterate of the recursion* (3) *and $\gamma_n = C_\gamma n^{-\gamma}, \beta_n = C_\beta n^\beta, \lambda_n = C_\lambda n^{-\lambda}$ with $C_\gamma > 0, C_\beta > 0$, and $C_\lambda > 0$. Assume that $\gamma, \beta, \lambda \geq 0$ and $\gamma + \lambda < 1$. Then, under H1 - H4, we have:*

$$\mathbb{E}\left[V(\theta_n) - V(\theta^*)\right] = O\left(n^{-\gamma+2\beta+\lambda} + r_n\right). \tag{6}$$

The rate obtained is classical and shows the tradeoff between a term coming from the adaptive steps (with a dependence on γ, β, λ) and a term r_n which depends on the control of the bias. To minimize the right hand-side of (6), we would like to have $\beta = \lambda = 0$. However, this would require much stronger assumptions. For example, in the case of Adagrad and RMSProp, the gradients would need to be bounded, which will be discussed later.

We stress that Theorem 4.1 applies to any adaptive algorithm of the form (3), with the only assumption being Assumption 4 on the eigenvalues of A_n. Without any information on these eigenvalues, the choice that $\beta_n \propto n^\beta$ and $\lambda_n \propto n^{-\lambda}$ allows us to remain very general.

To illustrate Theorem 4.1 and the impact of bias, we consider in Figure 1 a simple least squares objective function $V(\theta) = \|A\theta\|^2/2$ in dimension $d = 10$. We artificially add to every gradient a zero-mean Gaussian noise and a bias term $r_n = C_r n^{-r}$ at each iteration n. We use Adagrad with a learning rate $\gamma = 1/2, \beta = 0$ and $\lambda = 0$. Then, the bound of Theorem 4.1 is of the form $O(n^{-1/2} + n^{-r})$. First, note that the impact of a constant bias term ($r_n = 1$) never vanishes. From $r_n = 1$ to $r_n = n^{-1/2}$, the effect of the bias decreases until a threshold is reached where there is no significant improvement. The convergence rate in the case $r_n = n^{-1/2}$ is then the same as in the case without bias, illustrating the fact that in this case the dominating term comes from the learning rate.

Finally, note that non-adaptive SGD is a particular case of Theorem 4.1. Thus, our theorem gives new results also in the non-adaptive case with generic step sizes and biased gradients with decreasing bias.

4.2 Non-convex smooth case

In the non-convex smooth case, the theoretical results are based on a randomized version of Stochastic Approximation as described by Nemirovski et al. (2009); Ghadimi and Lan (2013); Karimi et al. (2019). In classical SA, the update (3) is performed a fixed number of times n, and the quantity of interest is the last parameter θ_n. On the other hand, in Randomized Stochastic Approximation, we introduce a random variable R which takes its values in $\{1, \ldots, n\}$ and the quantity of interest is θ_R. We stress that this procedure is a technical tool for the proofs, in practical applications we will always use classical SA.

The following theorem provides a bound in expectation on the gradient of the objective function V, which is the best results we can have given that no assumption is made about existence of a global minimum of V.

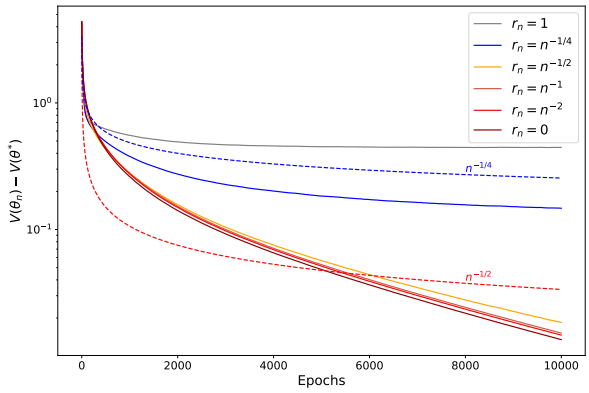

Figure 1: Value of $V(\theta_n) - V(\theta^*)$ with Adagrad for different values of $r_n = n^{-r}$ and a learning rate $\gamma_n = n^{-1/2}$. The dashed curve corresponds to the expected convergence rate $O(n^{-1/4})$ for $r = 1/4$ and $O(n^{-1/2})$ for $r \geq 1/2$.

Theorem 4.2. *Assume that H2-H4 hold. Assume also that for all $k \geq 0$, we have $\gamma_{k+1} \leq \lambda_{k+1}/(\tilde{\sigma}_1 L \beta_{k+1}^2)$. For any $n \geq 1$, let $R \in \{0, \dots, n\}$ be a discrete random variable such that:*

$$\mathbb{P}(R = k) := \frac{w_{k+1}\gamma_{k+1}\lambda_{k+1}}{\sum_{j=0}^{n} w_{j+1}\gamma_{j+1}\lambda_{j+1}} \, .$$

where $w_{k+1} = \prod_{j=1}^{k+1}(1 + \tilde{\sigma}_2 \delta_j)^{-1}$ with $\delta_j = L\gamma_j^2\beta_j^2/2$. Then,

$$\mathbb{E}\left[\|\nabla V(\theta_R)\|^2\right] \leq 2\frac{V_n^* + \alpha_{1,n} + \alpha_{2,n}}{\sum_{j=0}^{n} w_{j+1}\gamma_{j+1}\lambda_{j+1}} \, ,$$

where

$$\alpha_{1,n} = \sum_{k=0}^{n} w_{k+1}\gamma_{k+1}\lambda_{k+1}r_{k+1}, \alpha_{2,n} = \sum_{k=0}^{n} w_{k+1}\delta_{k+1}\sigma_k^2,$$
$$\text{and} \quad V_n^* = \mathbb{E}[V(\theta_0) - w_{n+1}V(\theta^*)].$$

If $\tilde{\sigma}_2 = 0$, Theorem 4.2 recovers the asymptotic rate of convergence obtained by Karimi et al. (2019) with respect to the hyperparameters γ, β, and λ and the bias. We can observe that if $\gamma \leq \lambda + 2\beta$, the condition on $(\gamma_k)_{k\geq 1}$ can be met simply by tuning C_γ. In particular, if $A_n = I_d$, the requirement on the step sizes can be expressed as $\gamma_{k+1} \leq 1/(\tilde{\sigma}_1 L)$.

We give below the convergence rates obtained from Theorem 4.2 under the same assumptions on γ_n, β_n and λ_n as in the PL case.

Corollary 4.3. *Assume that H2-H4 hold. Let $\gamma_n = C_\gamma n^{-\gamma}, \beta_n = C_\beta n^{\beta}, \lambda_n = C_\lambda n^{-\lambda}$ with $C_\gamma > 0, C_\beta > 0$, and $C_\lambda > 0$. Assume that $\gamma, \beta, \lambda \geq 0$ and $\gamma + \lambda < 1$. Then, if $\tilde{\sigma}_2 = 0$, we have:*

$$\mathbb{E}\left[\|\nabla V(\theta_R)\|^2\right] = \begin{cases} O\left(n^{-\gamma+\lambda+2\beta} + b_n\right) & \text{if } \vartheta < 1/2 \, , \\ O\left(n^{\gamma+\lambda-1} + b_n\right) & \text{if } \vartheta > 1/2 \, , \\ O\left(n^{\gamma+\lambda-1}\log n + b_n\right) & \text{if } \vartheta = 1/2 \, , \end{cases}$$

with $\vartheta = \gamma - \beta$. and where the bias term b_n comes from r_n and can be constant or decreasing. In the latter case, writing $r_n = C_r n^{-r}$, we have:

$$b_n = \begin{cases} O\left(n^{-r}\right) & \text{if } r + \lambda + \gamma < 1 \, , \\ O\left(n^{\gamma+\lambda-1}\right) & \text{if } r + \lambda + \gamma > 1 \, , \\ O\left(n^{\gamma+\lambda-1}\log n\right) & \text{if } r + \lambda + \gamma = 1 \, . \end{cases}$$

In practice, the value of r is known in advance while the other parameters can be tuned to achieve the optimal rate of convergence. In any scenario, we can never achieve a bound of $O(1/\sqrt{n} + b_n)$, and the best rate we can achieve is $O(\log n/\sqrt{n} + b_n)$ if and only if $\gamma = 1/2, \beta = 0$ and $\lambda = 0$. In this case, all eigenvalues of A_n must be bounded both from below and above. In the context of decreasing bias, if $r \geq 1/2$, the bias term contributes to the speed of the algorithm. Otherwise, the other term is the leading term of the upper bound. However, in both cases, the best achievable bound is $O(\log n/\sqrt{n})$ if $r \geq 1/2$.

Bounded Gradient Case. Now, we provide the convergence analysis of Randomized Adaptive Stochastic Approximation with a bounded stochastic update. Consider the following additional assumption about the stochastic update.

H5 The stochastic update is bounded, i.e., there exists $M \geq 0$ such that for all $n \in \mathbb{N}$,

$$\left\| H_{\theta_n}(X_{n+1}) \right\| \leq M.$$

It is important to note that under H3, this is equivalent to bounding the stochastic gradient of the objective function. Corollary 4.4 provides a bound on the gradient of the objective function V, which is similar to Theorem 4.2.

Corollary 4.4. *Assume that H2-H5 hold. Let $\gamma_n = C_\gamma n^{-\gamma}, \beta_n = C_\beta n^{\beta}, \lambda_n = C_\lambda n^{-\lambda}$ with $C_\gamma > 0, C_\beta > 0$, and $C_\lambda > 0$. Assume that $\gamma, \beta, \lambda \geq 0$ and $\gamma + \lambda < 1$. For any $n \geq 1$, let $R \in \{0, \dots, n\}$ be a uniformly distributed random variable. Then,*

$$\mathbb{E}\left[\|\nabla V(\theta_R)\|^2 \right] \leq \frac{V_n^* + \alpha'_{1,n} + LM^2 \alpha'_{2,n}/2}{\sqrt{n}},$$

where V_n^* is defined in Theorem 4.2, $\alpha'_{1,n} = \sum_{k=0}^{n} \gamma_{k+1} \lambda_{k+1} r_{k+1}$ and $\alpha'_{2,n} = \sum_{k=0}^{n} \gamma_{k+1}^2 \beta_{k+1}^2$.

Importantly, in Corollary 4.4, there are no assumptions on the step sizes, and we obtain a better bound than in Theorem 4.2.

4.3 Application to Adagrad and RMSProp

In this section, we provide the convergence analysis of Adagrad and RMSProp with a biased gradient estimator.

Remark 4.5. Under H5, for all eigenvalues λ of A_n, the adaptive matrix in Adagrad or RMSProp, it holds that $(M^2 + \delta)^{-1/2} \leq \lambda \leq \delta^{-1/2}$, i.e., H4 is satisfied with $\lambda = 0$ and $\beta = 0$.

Corollary 4.6. *Assume that H2 and H5 hold. Let $\gamma_n = c_\gamma n^{-1/2}$ and A_n denote the adaptive matrix in Adagrad or RMSProp. For any $n \geq 1$, let $R \in \{0, \dots, n\}$ be a uniformly distributed random variable. Suppose that for any $n \geq 1$, there exist positive constants α and C_α such that:*

$$\left\| \mathbb{E}\left[H_{\theta_n}(X_{n+1}) \,|\mathcal{F}_n \right] - \nabla V(\theta_n) \right\| \leq C_\alpha n^{-\alpha}. \tag{7}$$

Then,

$$\mathbb{E}\left[\|\nabla V(\theta_R)\|^2 \right] = O\left(\frac{\log n}{\sqrt{n}} + b_n \right).$$

The bias b_n is explicitly given in Appendix A.

Since we do not have information about the global minimum of the objective function V, Corollary 4.6 establishes the rate of convergence of Adagrad and RMSProp with biased gradient to a critical point for non-convex smooth functions. In the case of an unbiased gradient, we obtain the same bound as in Zou et al. (2018), under the same assumptions:

$$\mathbb{E}\left[\|\nabla V(\theta_R)\|^2 \right] = O\left(\frac{\log n}{\sqrt{n}} \right).$$

If the bias is of the order $O(n^{-1/4})$, the algorithm achieves the same convergence rate as in the case of an unbiased gradient.

4.4 Adam with Biased gradients

Here, we provide the convergence analysis of Adam with a biased gradient estimator. Since Adam uses an exponential moving average of past gradients instead of the current gradient, it changes the analysis of the convergence of this algorithm slightly. The exponential moving average of past squared gradients is defined by:

$$m_n = (1-\rho_1)\sum_{k=0}^{n}\rho_1^{n-k}H_{\theta_k}(X_{k+1}),$$

where ρ_1 is the moving average parameter. Under Assumption 5, we can control the minimum and maximum eigenvalues, as it has the same eigenvalues of A_n as in RMSProp.

Input: Initial point θ_0, maximum number of iterations n, step sizes $\{\gamma_k\}_{k\geq 1}$, momentum parameters $\rho_1, \rho_2 \in [0,1)$ and regularization parameter $\delta \geq 0$.
Set $m_0 = 0, V_0 = 0$.
for $k = 0$ to $n-1$ **do**
  Compute the stochastic update $H_{\theta_k}(X_{k+1})$.
  $m_k = \rho_1 m_{k-1} + (1-\rho_1)H_{\theta_k}(X_{k+1})$
  $V_k = \rho_2 V_{k-1} + (1-\rho_2)H_{\theta_k}(X_{k+1})H_{\theta_k}(X_{k+1})^\top$
  $A_k = \left[\delta I_d + \mathrm{Diag}(V_k)\right]^{-1/2}$
  $\theta_{k+1} = \theta_k - \gamma_{k+1}A_k m_k$
end for
Output: $(\theta_k)_{1\leq k\leq n}$

Algorithm 1: Adam with Biased gradients

Theorem 4.7. *Let $\gamma_n = c_\gamma n^{-1/2}$, A_n denote the adaptive matrix in Adam and $\rho_1 \in [0,1)$. For any $n \geq 1$, let $R \in \{0,\dots,n\}$ be a uniformly distributed random variable. Then, under H2, H3(i), H5,*

$$\mathbb{E}\left[\|\nabla V(\theta_R)\|^2\right] = O\left(\frac{\log n}{\sqrt{n}} + b_n\right),$$

where the term b_n corresponds to the bias which comes from r_n in H3(i). Choosing $r_n = C_r n^{-r}$, we get:

$$b_n = \begin{cases} O\left(n^{-r}\right) & \text{if } r < 1/2, \\ O\left(n^{-1/2}\right) & \text{if } r > 1/2, \\ O\left(n^{-1/2}\log n\right) & \text{if } r = 1/2. \end{cases}$$

If the bias is of the order $O(n^{-1/4})$, we achieve a convergence rate of $O(n^{-1/2}\log n)$, similar to that of Adagrad and RMSProp. It's worth noting that our results are also applicable to SGD momentum by taking $A_n = I_d$.

4.5 Some Applications

In this section, we propose different settings illustrating the range of application of our theoretical results.

- The first examples that we use to illustrate our results are IWAE and bias-reduced IWAE (BR-IWAE). In Appendix B, we provide details on IWAE, BR-IWAE, and several other bias reduction techniques. We also establish a bias control for IWAE in Theorem 5.1 which allows to obtain a convergence analysis.
- We also discuss the implications of our convergence results for Stochastic Bilevel Optimization and Conditional Stochastic Optimization in Appendix C. Specifically, Theorem C.4 and Theorem C.6 establish the convergence results in Stochastic Bilevel Optimization and Conditional Stochastic Optimization, respectively.
- We also give other examples in which the bias of the estimator can be controlled, such as Self-Normalized Importance Sampling (Appendix D.1), Sequential Monte Carlo Methods (Appendix D.2), Policy Gradient (Appendix D.3), Zeroth-Order Gradient (Appendix D.4), and Coordinate Sampling (Appendix D.5).

5 Experiments

In this section, we illustrate our theoretical results in the context of deep VAE. The experiments were conducted using PyTorch (Paszke et al., 2017), and the source code can be found here*. In generative models, the objective is to maximize the marginal likelihood defined as:

$$\log p_\theta(x) = \log \mathbb{E}_{p_\theta(\cdot|x)}\left[\frac{p_\theta(x, Z)}{p_\theta(Z|x)}\right],$$

where $(x, z) \mapsto p_\theta(x, z)$ is the complete likelihood, x are the observations and Z is the latent variable. Under some simple technical assumptions, by Fisher's identity, we have:

$$\nabla_\theta \log p_\theta(x) = \int \nabla_\theta \log p_\theta(x, z) p_\theta(z \mid x) \mathrm{d}z. \tag{8}$$

However, in most cases, the conditional density $z \mapsto p_\theta(z \mid x)$ is intractable and can only be sampled. Variational Autoencoders introduce an additional parameter ϕ and a family of variational distributions $z \mapsto q_\phi(z \mid x)$ to approximate the true posterior distribution. Parameters are estimated by maximizing the Evidence Lower Bound (ELBO):

$$\log p_\theta(x) \geq \mathbb{E}_{q_\phi(\cdot|x)}\left[\log \frac{p_\theta(x, Z)}{q_\phi(Z|x)}\right] =: \mathcal{L}_{\mathrm{ELBO}}(\theta, \phi; x).$$

The Importance Weighted Autoencoder (IWAE) (Burda et al., 2016) is a variant of the VAE that incorporates importance weighting to obtain a tighter ELBO. The IWAE objective can be written as follows:

$$\mathcal{L}_k^{\mathrm{IWAE}}(\theta, \phi; x) = \mathbb{E}_{q_\phi^{\otimes k}(\cdot|x)}\left[\log \frac{1}{k}\sum_{\ell=1}^{k} \frac{p_\theta(x, Z^{(\ell)})}{q_\phi(Z^{(\ell)}|x)}\right],$$

where k corresponds to the number of samples drawn from the variational posterior distribution. The estimator of the gradient of ELBO in IWAE corresponds to the biased estimator of the gradient of the marginal log likelihood $\log p_\theta(x)$. The following theorem provides the bias of this estimator.

Theorem 5.1. *Let $\mathsf{X} \subseteq \mathbb{R}^{d_x}$ and $\mathsf{Z} \subseteq \mathbb{R}^{d_z}$ denote the data space and the latent space, respectively. Assume that for all $\theta \in \Theta \subset \mathbb{R}^d$, $x \in \mathsf{X}$ and $z \in \mathsf{Z}$, there exist M such that:*

$$\left\|\nabla_\theta \log p_\theta(x, z)\right\| \leq M(x),$$

Then, for all $\theta \in \Theta$, $\phi \in \Phi$ and $x \in \mathsf{X}$, there exists a constant $C > 0$ such that:

$$\left\|\mathbb{E}_{q_\phi^{\otimes k}(\cdot|x)}\left[\widehat{\nabla}_\theta \mathcal{L}_k^{IWAE}(\theta, \phi; x) - \nabla_\theta V(\theta)\right]\right\| \leq \frac{C}{k},$$

where $\nabla_\theta V(\theta)$ and $\widehat{\nabla}_\theta \mathcal{L}_k^{IWAE}(\theta, \phi; x)$ are defined in (9).

Since bias has an impact on convergence rates, we propose to use one of the bias reduction techniques, the Biased Reduced Importance Weighted Autoencoder (BR-IWAE) (Cardoso et al., 2022), which is detailed in Appendix B.

Dataset. We conduct our experiments on the CIFAR-10 dataset (Krizhevsky et al., 2009), which is a widely used dataset for image classification tasks. Additional experiments are provided in Appendix E.

Model. We use a Convolutional Neural Network (CNN) architecture with the Rectified Linear Unit (ReLU) activation function for both the encoder and the decoder. The latent space dimension is set to 100. Further details of the model are provided in Appendix E. We estimate the log-likelihood using the VAE, IWAE, and BR-IWAE models, all of which are trained for 100 epochs. Training is conducted using Adagrad, RMSProp and Adam with a decaying learning rate.

For the first experiment, we set a constant bias, i.e., we use $k = 5$ samples in both IWAE and BR-IWAE, while restricting the maximum iteration of the MCMC algorithm to 5 for BR-IWAE. The test losses are presented in Figure 2. We show the negative log-likelihood on the test dataset for VAE, IWAE, and BR-IWAE with Adagrad, RMSProp and

*URL hidden during review process

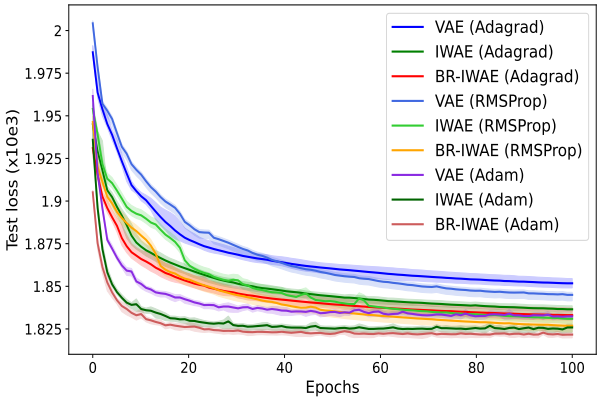

Figure 2: Negative Log-Likelihood on the test set for Different Generative Models with Adagrad, RMSProp, and Adam on the CIFAR-10 Dataset. Bold lines represent the mean over 5 independent runs.

Adam. As expected, we observe that IWAE outperforms VAE, while BR-IWAE outperforms IWAE by reducing bias in both cases.

To illustrate our results, we choose to incorporate a time-dependent bias that decreases by choosing a bias of order $O(n^{-\alpha})$ at iteration n as in (7). The bias of the estimator of the gradient in IWAE is of the order $O(1/k)$, where k is the number of importance weights. Therefore, choosing the bias of order $O(n^{-\alpha})$ is equivalent to using n^{α} samples at iteration n, to estimate the gradient. This procedure is detailed in Appendix B. We vary α only for IWAE for computational efficiency and plot the following quantities.

- In Figures 3 and 4, the gradient squared norm $\|\nabla V(\theta_n)\|^2$ to illustrate the convergence rate.
- In Figure 5, the Negative Log-Likelihood along iterations.

Figures 3 and 4 illustrate our results, while the other figures are meant to confirm the behavior of test loss with different values of α. All figures are plotted on a logarithmic scale for better visualization. It is important to note that all figures are in respect to epochs, whereas here, n represents the iteration (number of updates of the gradient).

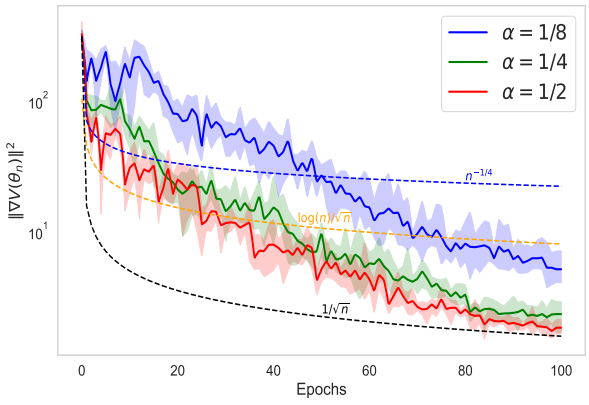

Figure 3: Value of $\|\nabla V(\theta_n)\|^2$ in IWAE with Adagrad. Bold lines represent the mean over 5 independent runs.

Note that the dashed curve corresponds to the expected convergence rate $O(n^{-1/4})$ for $\alpha = 1/8$ and $O(\log n/\sqrt{n})$ for $\alpha = 1/4$ and for $\alpha = 1/2$. We can clearly observe that for each of the cases, fast convergence is achieved when n is sufficiently large. There are several possible explanations for this rapid convergence with a decreasing time-dependent bias. First, we may be able to improve the upper bound by obtaining for instance a better bound for the bias.

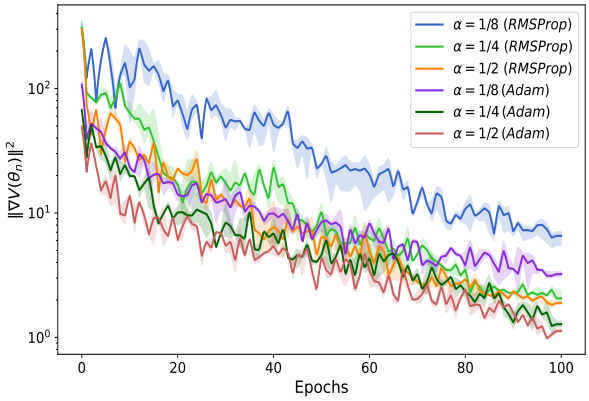

Figure 4: Value of $\|\nabla V(\theta_n)\|^2$ in IWAE with RMSProp and Adam. Bold lines represent the mean over 5 independent runs.

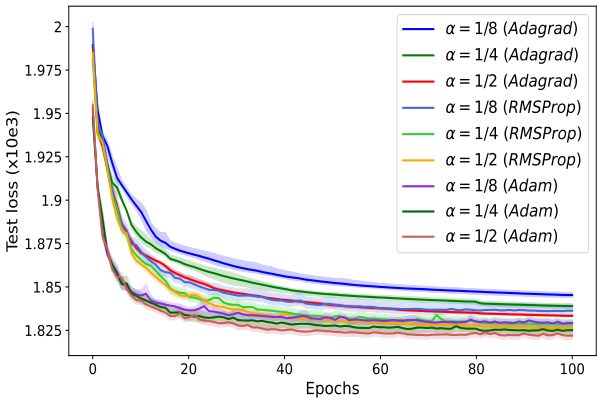

Figure 5: Negative Log-Likelihood on the test set on the CIFAR-10 Dataset for IWAE with Adagrad, RMSProp, and Adam. Bold lines represent the mean over 5 independent runs.

Our experiments show similar results for Adagrad, RMSProp and Adam in terms of convergence rates, although Adam performs slightly better due to the incorporation of momentum. We consistently observe the impact of bias, although it tends to be relatively small, as the bias correction terms may help mitigate bias in the moving averages.

It is clear that with a larger α, convergence in both squared gradient norm and negative log-likelihood is faster. However, beyond a certain threshold for α, we observe that the rate of convergence does not change significantly. Since choosing a larger α induces an additional computational cost, it is crucial to select an appropriate value of α, that achieves fast convergence without being too computationally costly.

6 Discussion

This paper provides a non-asymptotic analysis of Biased Adaptive Stochastic Approximation with and without the PL condition in the non-convex smooth setting. We derive a convergence rate of $O(\log n/\sqrt{n} + b_n)$ for non-convex smooth functions, where b_n corresponds to the time-dependent decreasing bias, and an improved linear convergence rate with the Polyak-Łojasiewicz (PL) condition. We also establish that Adagrad, RMSProp, and Adam with biased gradients converge to critical points for non-convex smooth functions. Our results provide insights on hyper-parameters tuning to achieve fast convergence and reduce computational time.

Impact Statements

This paper presents work aimed at advancing the field of Machine Learning. Our work may have several potential societal consequences, but we do not believe any of them require specific highlighting in this context.

References

Agapiou, S., Papaspiliopoulos, O., Sanz-Alonso, D., and Stuart, A. M. (2017). Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, pages 405–431.

Ajalloeian, A. and Stich, S. U. (2020). On the Convergence of SGD with Biased Gradients. *arXiv preprint arXiv:2008.00051*.

Alacaoglu, A. and Lyu, H. (2023). Convergence of first-order methods for constrained nonconvex optimization with dependent data. In *International Conference on Machine Learning*, pages 458–489. PMLR.

Auer, P., Cesa-Bianchi, N., and Gentile, C. (2002). Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64(1):48–75.

Baxter, J. and Bartlett, P. L. (2001). Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350.

Beznosikov, A., Horváth, S., Richtárik, P., and Safaryan, M. (2023). On biased compression for distributed learning. *Journal of Machine Learning Research*, 24(276):1–50.

Bhandari, J., Russo, D., and Singal, R. (2018). A finite time analysis of temporal difference learning with linear function approximation. In *Conference On Learning Theory*, pages 1691–1692. PMLR.

Bottou, L. (1991). *Une approche théorique de l'apprentissage connexioniste; applications à la reconnaissance de la parole*. PhD thesis, Paris 11.

Bottou, L., Curtis, F. E., and Nocedal, J. (2018). Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311.

Boyer, C. and Godichon-Baggioni, A. (2023). On the asymptotic rate of convergence of stochastic newton algorithms and their weighted averaged versions. *Computational Optimization and Applications*, 84(3):921–972.

Burda, Y., Grosse, R., and Salakhutdinov, R. (2016). Importance weighted autoencoders. In *International Conference on Learning Representations*.

Cardoso, G., El Idrissi, Y. J., Le Corff, S., Moulines, É., and Olsson, J. (2023). State and parameter learning with PaRIS particle Gibbs. In *International Conference on Machine Learning*, pages 3625–3675. PMLR.

Cardoso, G., Samsonov, S., Thin, A., Moulines, E., and Olsson, J. (2022). BR-SNIS: bias reduced self-normalized importance sampling. *Advances in Neural Information Processing Systems*, 35:716–729.

Chen, C., Shen, L., Zou, F., and Liu, W. (2022). Towards practical adam: Non-convexity, convergence theory, and mini-batch acceleration. *Journal of Machine Learning Research*, 23(229):1–47.

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. (2017). Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26.

Chen, T., Sun, Y., and Yin, W. (2021). Closing the gap: Tighter analysis of alternating stochastic gradient methods for bilevel problems. *Advances in Neural Information Processing Systems*, 34:25294–25307.

Dalal, G., Szörényi, B., Thoppe, G., and Mannor, S. (2018). Finite sample analyses for td (0) with function approximation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Défossez, A., Bottou, L., Bach, F., and Usunier, N. (2020). A simple convergence proof of Adam and Adagrad. *arXiv preprint arXiv:2003.02395*.

Del Moral, P., Doucet, A., and Singh, S. S. (2010). A backward particle interpretation of Feynman-Kac formulae. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44(5):947–975.

Demidovich, Y., Malinovsky, G., Sokolov, I., and Richtárik, P. (2024). A guide through the zoo of biased sgd. *Advances in Neural Information Processing Systems*, 36.

Dieuleveut, A., Fort, G., Moulines, E., and Wai, H.-T. (2023). Stochastic approximation beyond gradient for signal processing and machine learning. *IEEE Transactions on Signal Processing*.

Doan, T. T., Nguyen, L. M., Pham, N. H., and Romberg, J. (2020). Finite-time analysis of stochastic gradient descent under markov randomness. *arXiv preprint arXiv:2003.10973*.

Dozat, T. (2016). Incorporating Nesterov momentum into Adam.

Du, S., Lee, J., Li, H., Wang, L., and Zhai, X. (2019). Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pages 1675–1685. PMLR.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7).

Fazel, M., Ge, R., Kakade, S., and Mesbahi, M. (2018). Global convergence of policy gradient methods for the linear quadratic regulator. In *International conference on machine learning*, pages 1467–1476. PMLR.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

Franceschi, L., Frasconi, P., Salzo, S., Grazzi, R., and Pontil, M. (2018). Bilevel programming for hyperparameter optimization and meta-learning. In *International conference on machine learning*, pages 1568–1577. PMLR.

Ghadimi, S. and Lan, G. (2013). Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368.

Ghadimi, S. and Wang, M. (2018). Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*.

Gloaguen, P., Le Corff, S., and Olsson, J. (2022). A pseudo-marginal sequential Monte Carlo online smoothing algorithm. *Bernoulli*, 28(4):2606–2633.

Godichon-Baggioni, A. and Tarrago, P. (2023). Non asymptotic analysis of adaptive stochastic gradient algorithms and applications. *arXiv preprint arXiv:2303.01370*.

Grazzi, R., Pontil, M., and Salzo, S. (2023). Bilevel optimization with a lower-level contraction: Optimal sample complexity without warm-start. *Journal of Machine Learning Research*, 24(167):1–37.

Hong, M., Wai, H.-T., Wang, Z., and Yang, Z. (2023). A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic. *SIAM Journal on Optimization*, 33(1):147–180.

Hu, B., Seiler, P., and Lessard, L. (2021a). Analysis of biased stochastic gradient descent using sequential semidefinite programs. *Mathematical programming*, 187:383–408.

Hu, Y., Chen, X., and He, N. (2021b). On the bias-variance-cost tradeoff of stochastic optimization. *Advances in Neural Information Processing Systems*, 34:22119–22131.

Hu, Y., Zhang, S., Chen, X., and He, N. (2020). Biased stochastic first-order methods for conditional stochastic optimization and applications in meta learning. *Advances in Neural Information Processing Systems*, 33:2759–2770.

Huang, F., Li, J., and Gao, S. (2021). Biadam: Fast adaptive bilevel optimization methods. *arXiv preprint arXiv:2106.11396*.

Jaakkola, T., Jordan, M., and Singh, S. (1993). Convergence of stochastic iterative dynamic programming algorithms. *Advances in Neural Information Processing Systems*, 6.

Ji, K., Yang, J., and Liang, Y. (2021). Bilevel optimization: Convergence analysis and enhanced design. In *International conference on machine learning*, pages 4882–4892. PMLR.

Karimi, B., Miasojedow, B., Moulines, E., and Wai, H.-T. (2019). Non-asymptotic analysis of biased stochastic approximation scheme. In *Conference on Learning Theory*, pages 1944–1974. PMLR.

Karimi, H., Nutini, J., and Schmidt, M. (2016). Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I 16*, pages 795–811. Springer.

Khaled, A. and Richtárik, P. (2020). Better theory for sgd in the nonconvex world. *arXiv preprint arXiv:2002.03329*.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *International Conference on Learning Representations*.

Kresoja, M., Lužanin, Z., and Stojkovska, I. (2017). Adaptive stochastic approximation algorithm. *Numerical Algorithms*, 76(4):917–937.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Lakshminarayanan, C. and Szepesvari, C. (2018). Linear stochastic approximation: How far does constant step-size and iterate averaging go? In *International Conference on Artificial Intelligence and Statistics*, pages 1347–1355. PMLR.

Leluc, R. and Portier, F. (2022). Sgd with coordinate sampling: Theory and practice. *The Journal of Machine Learning Research*, 23(1):15470–15516.

Li, Q. and Wai, H.-T. (2022). State dependent performative prediction with stochastic approximation. In *International Conference on Artificial Intelligence and Statistics*, pages 3164–3186. PMLR.

Liu, H., Simonyan, K., and Yang, Y. (2018). Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*.

Liu, Y. and Tajbakhsh, S. D. (2023). Adaptive stochastic optimization algorithms for problems with biased oracles. *arXiv preprint arXiv:2306.07810*.

McLeish, D. (2011). A general method for debiasing a monte carlo estimator. *Monte Carlo methods and applications*, 17(4):301–315.

Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., and Frossard, P. (2017). Universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1765–1773.

Moulines, E. and Bach, F. (2011). Non-asymptotic analysis of stochastic approximation algorithms for machine learning. *Advances in Neural Information Processing Systems*, 24.

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. (2009). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609.

Nesterov, Y. and Spokoiny, V. (2017). Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17:527–566.

Nowozin, S. (2018). Debiasing evidence approximations: On importance-weighted autoencoders and jackknife variational inference. In *International Conference on Learning Representations*.

Nutini, J., Schmidt, M., Laradji, I., Friedlander, M., and Koepke, H. (2015). Coordinate descent converges faster with the gauss-southwell rule than random selection. In *International Conference on Machine Learning*, pages 1632–1641. PMLR.

Olsson, J. and Westerborn, J. (2017). Efficient particle-based online smoothing in general hidden Markov models: the paris algorithm.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in PyTorch.

Polyak, B. T. and Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855.

Reddi, S. J., Kale, S., and Kumar, S. (2018). On the convergence of adam and beyond. In *International Conference on Learning Representations*.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407.

Sun, T., Sun, Y., and Yin, W. (2018). On Markov chain gradient descent. *Advances in Neural Information Processing Systems*, 31.

Tadić, V. B. and Doucet, A. (2011). Asymptotic bias of stochastic gradient search. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 722–727. IEEE.

Teh, Y., Newman, D., and Welling, M. (2006). A collapsed variational bayesian inference algorithm for latent dirichlet allocation. *Advances in neural information processing systems*, 19.

Tieleman, T., Hinton, G., et al. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2):26–31.

Tong, Q., Liang, G., and Bi, J. (2022). Calibrating the adaptive learning rate to improve convergence of adam. *Neurocomputing*, 481:333–356.

Ward, R., Wu, X., and Bottou, L. (2020). Adagrad stepsizes: Sharp convergence over nonconvex landscapes. *The Journal of Machine Learning Research*, 21(1):9047–9076.

Wu, X., Sun, J., Hu, Z., Li, J., Zhang, A., and Huang, H. (2024). Federated conditional stochastic optimization. *Advances in Neural Information Processing Systems*, 36.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Zaheer, M., Reddi, S., Sachan, D., Kale, S., and Kumar, S. (2018). Adaptive methods for nonconvex optimization. *Advances in neural information processing systems*, 31.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Zou, F., Shen, L., Jie, Z., Sun, J., and Liu, W. (2018). Weighted adagrad with unified momentum. *arXiv preprint arXiv:1808.03408*.

A Convergence Proofs

A.1 Proof of Theorem 4.1

We first establish a technical lemma which is essential for the proof.

Lemma A.1. *Let* $(\delta_n)_{n\geq 0}$, $(\gamma_n)_{n\geq 1}$, $(\eta_n)_{n\geq 1}$, *and* $(v_n)_{n\geq 1}$ *be some positive sequences satisfying the following assumptions.*

- *The sequence δ_n follows the recursive relation:*

$$\delta_n \leq (1 - 2\omega\gamma_n + \eta_n\gamma_n)\,\delta_{n-1} + v_n\gamma_n,$$

with $\delta_0 \geq 0$ and $\omega > 0$.

- *Let $n_0 = \inf\{n \geq 1 : \eta_n \leq \omega\}$, then for all $n \geq n_0 + 1$, we assume that $\omega\gamma_n \leq 1$.*

Then, for all $n \in \mathbb{N}$,

$$\delta_n \leq \exp\left(-\omega \sum_{k=n/2}^{n} \gamma_k\right) \exp\left(2 \sum_{k=1}^{n} \eta_k\gamma_k\right)\left(\delta_0 + 2 \max_{1\leq k\leq n} \frac{v_k}{\eta_k}\right) + \frac{1}{\omega} \max_{n/2\leq k\leq n} v_k.$$

The proof is given in Godichon-Baggioni and Tarrago (2023).

Theorem A.2. *Let $\theta_n \in \mathbb{R}^d$ be the n-th iterate of recursion* (3)*. Under H1-H 4,*

$$\mathbb{E}\left[V(\theta_n) - V(\theta^*)\right] \leq \left(\mathbb{E}\left[V(\theta_0) - V(\theta^*)\right] + 2 \max_{1\leq k\leq n} \frac{\lambda_{k+1} v_k}{\beta_{k+1}^2 \gamma_{k+1}}\right) \exp\left(-\frac{\mu}{2} \sum_{k=n/2}^{n} \lambda_{k+1}\gamma_{k+1}\right)$$
$$\times \exp\left(2 \sum_{k=1}^{n} C_k \beta_{k+1}^2 \gamma_{k+1}^2\right) + \frac{2}{\mu} \max_{n/2\leq k\leq n} v_k,$$

where

$$C_k = \max\left\{1, \frac{\mu^2 \lambda_{k+1}^2}{(2\tilde{\sigma}_2 L + 4\tilde{\sigma}_1 L^2)\beta_{k+1}^2}\right\} \quad \text{and} \quad v_k = r_{k+1} + \frac{L\sigma_k^2}{2} \frac{\beta_{k+1}^2}{\lambda_{k+1}} \gamma_{k+1}.$$

with the convention $C_k = 1$ if $\tilde{\sigma}_1 = \tilde{\sigma}_2 = 0$.

Proof. As V is L smooth (Assumption H2) and using the recursion (3) of Adaptive SA, we obtain:

$$\begin{aligned} V(\theta_{n+1}) &\leq V(\theta_n) + \langle \nabla V(\theta_n) \mid \theta_{n+1} - \theta_n \rangle + \frac{L}{2} \|\theta_{n+1} - \theta_n\|^2 \\ &\leq V(\theta_n) - \gamma_{n+1} \langle \nabla V(\theta_n) \mid A_n H_{\theta_n}(X_{n+1}) \rangle + \frac{L\gamma_{n+1}^2}{2} \|A_n\|^2 \left\|H_{\theta_n}(X_{n+1})\right\|^2 \end{aligned}$$

Writing $V_n = V(\theta_n) - V(\theta^*)$, we get

$$V_{n+1} \leq V_n - \gamma_{n+1} \langle \nabla V(\theta_n) \mid A_n H_{\theta_n}(X_{n+1}) \rangle + \frac{L}{2} \gamma_{n+1}^2 \beta_{n+1}^2 \left\|H_{\theta_n}(X_{n+1})\right\|^2.$$

Then, using H3,

$$\begin{aligned} \mathbb{E}[V_{n+1}] &\leq \mathbb{E}[V_n] - \gamma_{n+1} \mathbb{E}\left[\langle \nabla V(\theta_n) \mid A_n H_{\theta_n}(X_{n+1}) \rangle\right] + \frac{L}{2} \gamma_{n+1}^2 \beta_{n+1}^2 \left(\tilde{\sigma}_2 \mathbb{E}[V_n] + \tilde{\sigma}_1 \mathbb{E}[\|\nabla V(\theta_n)\|^2] + \sigma_n^2\right) \\ &\leq \left(1 + \frac{\tilde{\sigma}_2 L}{2} \beta_{n+1}^2 \gamma_{n+1}^2\right) \mathbb{E}[V_n] + \gamma_{n+1}\lambda_{n+1} r_{n+1} - \gamma_{n+1} \left(\lambda_{n+1} - \frac{\tilde{\sigma}_1 L}{2} \gamma_{n+1} \beta_{n+1}^2\right) \mathbb{E}[\|\nabla V(\theta_n)\|^2] \\ &\quad + \frac{L\sigma_n^2}{2} \gamma_{n+1}^2 \beta_{n+1}^2. \end{aligned}$$

Furthermore, since V satisfies the Polyak-Łojasiewicz condition (H1) and since $\|\nabla V(\theta_n)\|^2 \leq 2LV_n$ (H2),

$$\mathbb{E}[V_{n+1}] \leq \left(1 - \mu\lambda_{n+1}\gamma_{n+1} + \left(\frac{\tilde{\sigma}_2 L}{2} + \tilde{\sigma}_1 L^2\right) \beta_{n+1}^2 \gamma_{n+1}^2\right) \mathbb{E}[V_n] + \gamma_{n+1}\lambda_{n+1} r_{n+1} + \frac{L\sigma_n^2}{2} \gamma_{n+1}^2 \beta_{n+1}^2$$

By choosing $\bar{\gamma}_{n+1} = \lambda_{n+1}\gamma_{n+1}$, we get:

$$\mathbb{E}[V_{n+1}] \leq \left(1 - \mu\bar{\gamma}_{n+1} + \left(\frac{\tilde{\sigma}_2 L}{2} + \tilde{\sigma}_1 L^2\right) \frac{\beta_{n+1}^2}{\lambda_{n+1}^2} \bar{\gamma}_{n+1}^2\right) \mathbb{E}[V_n] + \bar{\gamma}_{n+1} r_{n+1} + \frac{L\sigma_n^2}{2} \frac{\beta_{n+1}^2}{\lambda_{n+1}} \bar{\gamma}_{n+1}\gamma_{n+1}.$$

In order to satisfy the assumptions of Lemma A.1, consider $C_n = \max\left\{1, (\mu^2\lambda_{n+1}^2)/((2\tilde{\sigma}_2 L + 4\tilde{\sigma}_1 L^2)\beta_{n+1}^2)\right\}$, and since $C_n \geq 1$, we have:

$$\mathbb{E}\left[V_{n+1}\right] \leq \left(1 - \mu\bar{\gamma}_{n+1} + \frac{C_n\beta_{n+1}^2}{\lambda_{n+1}^2}\bar{\gamma}_{n+1}^2\right)\mathbb{E}\left[V_n\right] + \bar{\gamma}_{n+1}r_{n+1} + \frac{L\sigma_n^2}{2}\frac{\beta_{n+1}^2}{\lambda_{n+1}}\bar{\gamma}_{n+1}\gamma_{n+1}.$$

Now, using lemma A.1 by choosing:

$$\delta_n = \mathbb{E}\left[V_n\right], \quad \eta_n = \frac{C_n\beta_{n+1}^2}{\lambda_{n+1}^2}\bar{\gamma}_{n+1}, \quad \omega = \frac{\mu}{2}, \quad v_n = r_{n+1} + \frac{L\sigma_n^2}{2}\frac{\beta_{n+1}^2}{\lambda_{n+1}}\gamma_{n+1},$$

we have:

$$\mathbb{E}\left[V\left(\theta_n\right) - V(\theta^*)\right] \leq \left(\mathbb{E}\left[V\left(\theta_0\right) - V(\theta^*)\right] + 2\max_{1\leq k\leq n}\frac{v_k\lambda_{k+1}^2}{\beta_{k+1}^2\bar{\gamma}_{k+1}}\right)\mathrm{e}^{-\frac{\mu}{2}\sum_{k=n/2}^n\bar{\gamma}_{k+1}}\mathrm{e}^{2\sum_{k=1}^n C_k\beta_{k+1}^2\bar{\gamma}_{k+1}^2/\lambda_{k+1}^2} + \frac{2}{\mu}\max_{n/2\leq k\leq n}\left\{v_k\right\},$$

which concludes the proof by choosing $\bar{\gamma}_{n+1} = \lambda_{n+1}\gamma_{n+1}$. □

A.2 Proof of Theorem 4.2

By H2, V is L-smooth and using the recursion (3) of Adaptive SA together with a Taylor expansion, we obtain:

$$V\left(\theta_{k+1}\right) \leq V\left(\theta_k\right) + \left\langle \nabla V\left(\theta_k\right) \mid \theta_{k+1} - \theta_k \right\rangle + \frac{L}{2}\left\|\theta_{k+1} - \theta_k\right\|^2,$$

which yields

$$V\left(\theta_{k+1}\right) \leq V\left(\theta_k\right) - \gamma_{k+1}\left\langle \nabla V\left(\theta_k\right) \mid A_k H_{\theta_k}\left(X_{k+1}\right)\right\rangle + \delta_{k+1}\left\|H_{\theta_k}\left(X_{k+1}\right)\right\|^2,$$

with $\delta_{k+1} = L\gamma_{k+1}^2\beta_{k+1}^2/2$. Using Assumptions H3 and H4,

$$\begin{aligned}\mathbb{E}[V(\theta_{k+1})] \leq \mathbb{E}[V(\theta_k)] - \gamma_{k+1}\lambda_{k+1}\mathbb{E}[\|\nabla V(\theta_k)\|^2] + \gamma_{k+1}\lambda_{k+1}r_{k+1} + \delta_{k+1}\sigma_k^2 \\ + \delta_{k+1}\left(\tilde{\sigma}_1\mathbb{E}[\|\nabla V(\theta_k)\|^2] + \tilde{\sigma}_2\mathbb{E}\left[V\left(\theta_k\right) - V\left(\theta^*\right)\right]\right).\end{aligned}$$

Therefore,

$$\begin{aligned}\gamma_{k+1}\left(\lambda_{k+1} - \frac{L\tilde{\sigma}_1}{2}\gamma_{k+1}\beta_{k+1}^2\right)\mathbb{E}[\|\nabla V\left(\theta_k\right)\|^2] \leq \left(1 + \tilde{\sigma}_2\delta_{k+1}\right)\left(\mathbb{E}\left[V\left(\theta_k\right)\right] - V\left(\theta^*\right)\right) - \left(\mathbb{E}\left[V\left(\theta_{k+1}\right)\right] - V\left(\theta^*\right)\right) \\ + \gamma_{k+1}\lambda_{k+1}r_{k+1} + \delta_{k+1}\sigma_k^2.\end{aligned}$$

Let us now consider the sequence of weights w_k defined by $w_0 = 1$ and $w_k = \prod_{j=1}^k\left(1 + \tilde{\sigma}_2\delta_j\right)^{-1}$. Then,

$$\begin{aligned}w_{k+1}\gamma_{k+1}\left(\lambda_{k+1} - \frac{L\tilde{\sigma}_1}{2}\gamma_{k+1}\beta_{k+1}^2\right)\mathbb{E}[\|\nabla V\left(\theta_k\right)\|^2] \leq w_k\left(\mathbb{E}[V\left(\theta_k\right)] - V\left(\theta^*\right)\right) - w_{k+1}\left(\mathbb{E}\left[V\left(\theta_{k+1}\right)\right] - V\left(\theta^*\right)\right) \\ + w_{k+1}\gamma_{k+1}\lambda_{k+1}r_{k+1} + w_{k+1}\delta_{k+1}\sigma_k^2.\end{aligned}$$

In the sequel, let us now denote $V_n = [V\left(\theta_n\right)] - V\left(\theta^*\right)$, so that

$$\begin{aligned}\sum_{k=0}^n w_{k+1}\gamma_{k+1}\lambda_{k+1}\left(1 - \frac{L\tilde{\sigma}_1\gamma_{k+1}\beta_{k+1}^2}{2\lambda_{k+1}}\right)\mathbb{E}\left[\|\nabla V\left(\theta_k\right)\|^2\right] \leq w_n\mathbb{E}\left[V_n\right] - w_{n+1}\mathbb{E}\left[V_{n+1}\right] + \frac{1}{2}\sum_{k=0}^n w_{k+1}\gamma_{k+1}\lambda_{k+1}r_{k+1} \\ + \sum_{k=0}^n w_{k+1}\delta_{k+1}\sigma_k^2.\end{aligned}$$

Then, given that $\gamma_{k+1} \leq \lambda_{k+1}/(L\tilde{\sigma}_1\beta_{k+1}^2)$, we have

$$\frac{1}{2}\mathbb{E}\left[\sum_{k=0}^{n} w_{k+1}\gamma_{k+1}\lambda_{k+1}\left\|\nabla V\left(\theta_k\right)\right\|^2\right] \leq w_0\mathbb{E}\left[V_0\right] - w_{n+1}\mathbb{E}\left[V_{n+1}\right] + \frac{1}{2}\sum_{k=0}^{n} w_{k+1}\gamma_{k+1}\lambda_{k+1}r_{k+1} + \sum_{k=0}^{n} w_{k+1}\delta_{k+1}\sigma_k^2 \ .$$

Consequently, by definition of the discrete random variable R,

$$\begin{aligned}\mathbb{E}\left[\left\|\nabla V\left(\theta_R\right)\right\|^2\right] &= \sum_{k=0}^{n} w_{k+1}\frac{\gamma_{k+1}\lambda_{k+1}\mathbb{E}\left[\left\|\nabla V\left(\theta_k\right)\right\|^2\right]}{\sum_{j=0}^{n} w_{j+1}\gamma_{j+1}\lambda_{j+1}} \\ &\leq 2\frac{\mathbb{E}\left[V_0\right] - w_{n+1}\mathbb{E}\left[V_{n+1}\right] + \sum_{k=0}^{n} w_{k+1}\gamma_{k+1}r_{k+1} + \sum_{k=0}^{n} w_{k+1}\delta_{k+1}\sigma_k^2}{\sum_{j=0}^{n} w_{j+1}\gamma_{j+1}\lambda_{j+1}},\end{aligned}$$

which concludes the proof by noting that $V_{n+1} \geq V\left(\theta^*\right)$.

A.3 Proof of Corollary 4.3

The proof is a direct consequence of the fact that for a sufficiently large n:

$$\sum_{k=1}^{n}\frac{1}{k^s} = \begin{cases} \mathcal{O}\left(n^{-s+1}\right) & \text{if } 0 \leq s < 1 \ , \\ \mathcal{O}\left(1\right) & \text{if } s > 1 \ , \\ \mathcal{O}\left(\log n\right) & \text{if } s = 1 \ . \end{cases}$$

A.4 Proof of Corollary 4.4

By Assumption H2, V is L-smooth and using recursion (3) of Adaptive SA same as Theorem 4.2, we obtain:

$$\begin{aligned}V\left(\theta_{k+1}\right) &\leq V\left(\theta_k\right) + \left\langle \nabla V\left(\theta_k\right) \mid \theta_{k+1} - \theta_k \right\rangle + \frac{L}{2}\left\|\theta_{k+1} - \theta_k\right\|^2 \\ &\leq V\left(\theta_k\right) - \gamma_{k+1}\left\langle \nabla V\left(\theta_k\right) \mid A_k H_{\theta_k}\left(X_{k+1}\right)\right\rangle + \frac{L\gamma_{k+1}^2}{2}\left\|A_k\right\|^2\left\|H_{\theta_k}\left(X_{k+1}\right)\right\|^2,\end{aligned}$$

which, using H5 yields:

$$V\left(\theta_{k+1}\right) \leq V\left(\theta_k\right) - \gamma_{k+1}\left\langle \nabla V\left(\theta_k\right) \mid A_k H_{\theta_k}\left(X_{k+1}\right)\right\rangle + \frac{L}{2}\gamma_{k+1}^2\beta_{k+1}^2 M^2.$$

Using H3,

$$\mathbb{E}[V(\theta_{k+1})|\mathcal{F}_k] \leq V(\theta_k) - \gamma_{k+1}\lambda_{n+1}\|\nabla V(\theta_k)\|^2 + \gamma_{k+1}\lambda_{k+1}r_{k+1} + \frac{LM^2}{2}\gamma_{k+1}^2\beta_{k+1}^2 \ .$$

Therefore,

$$\gamma_{k+1}\lambda_{k+1}\left\|\nabla V\left(\theta_k\right)\right\|^2 \leq V\left(\theta_k\right) - \mathbb{E}\left[V\left(\theta_{k+1}\right)|\mathcal{F}_k\right] + \gamma_{k+1}\lambda_{k+1}r_{k+1} + \frac{LM^2}{2}\gamma_{k+1}^2\beta_{k+1}^2,$$

and

$$\sum_{k=0}^{n}\gamma_{k+1}\lambda_{k+1}\mathbb{E}\left[\left\|\nabla V\left(\theta_k\right)\right\|^2\right] \leq \mathbb{E}\left[V\left(\theta_0\right) - V\left(\theta_{n+1}\right)\right] + \sum_{k=0}^{n}\gamma_{k+1}\lambda_{k+1}r_{k+1} + \frac{LM^2}{2}\sum_{k=0}^{n}\gamma_{k+1}^2\beta_{k+1}^2.$$

Consequently, by definition of the discrete random variable R,

$$\begin{aligned}\mathbb{E}\left[\left\|\nabla V\left(\theta_R\right)\right\|^2\right] &= \frac{1}{n}\sum_{k=0}^{n}\mathbb{E}\left[\left\|\nabla V\left(\theta_k\right)\right\|^2\right] \leq \sum_{k=0}^{n}\frac{\gamma_{k+1}\lambda_{k+1}}{\sqrt{n}}\mathbb{E}\left[\left\|\nabla V\left(\theta_k\right)\right\|^2\right] \\ &\leq \frac{V_{0,n} + \sum_{k=0}^{n}\gamma_{k+1}\lambda_{k+1}r_{k+1} + LM^2\sum_{k=0}^{n}\gamma_{k+1}^2\beta_{k+1}^2/2}{\sqrt{n}} \ ,\end{aligned}$$

where $V_{0,n} = \mathbb{E}[V(\theta_0) - V(\theta_{n+1})]$, which conclude the proof by noting that $V(\theta_{n+1}) \geq V(\theta^*)$.

A.5 Proof of Corollary 4.6

Here, we consider the case where the regularization is non-increasing, i.e. where $\delta = \beta_{n+1}$. The constant case is strictly analogous.

Adagrad

- **Lower bound for the smallest eigenvalue of A_n.** By assumption H5, we have:

$$\left\| \frac{1}{n+1} \sum_{k=0}^{n} H_{\theta_k}(X_{k+1}) H_{\theta_k}(X_{k+1})^\top \right\| \leq M^2.$$

  This implies that:

$$\lambda_{\min}(A_n) = \lambda_{\max}\left(\beta_{n+1}^{-2} I_d + \mathrm{Diag}\left(\frac{1}{n+1} \sum_{k=0}^{n} H_{\theta_k}(X_{k+1}) H_{\theta_k}(X_{k+1})^\top \right) \right)^{-1/2} \geq (\beta_1^{-2} + M^2)^{-1/2}.$$

- **Upper bound for the largest eigenvalue of A_n.**

$$\lambda_{\max}(A_n) = \lambda_{\min}\left(\beta_{n+1}^{-2} I_d + \mathrm{Diag}\left(\frac{1}{n+1} \sum_{k=0}^{n} H_{\theta_k}(X_{k+1}) H_{\theta_k}(X_{k+1})^\top \right) \right)^{-1/2} \leq \beta_{n+1}.$$

Therefore, by setting $\lambda_{n+1} = (\beta_1^{-2} + M^2)^{-1/2}$ and $\beta_n = C_\beta n^\beta$, we have $\lambda = 0$ and one can arbitrarily choose β (one can take $\beta = 0$ for the constant regularization case).

RMSProp

- **Lower bound for the smallest eigenvalue of A_n.** By assumption H5, we have:

$$\|V_n\| \leq (1-\rho) \sum_{k=1}^{n} \rho^{n-k} \left\| H_{\theta_k}(X_{k+1}) \right\|^2 \leq M^2 (1-\rho) \sum_{k=1}^{n} \rho^{n-k} \leq M^2,$$

  where we used the fact that $\sum_{k=1}^{n} \rho^{n-k} \leq (1-\rho)^{-1}$. This implies that:

$$\lambda_{\min}(A_n) = \lambda_{\max}\left(\beta_{n+1}^{-2} I_d + \mathrm{Diag}\left(V_n \right) \right)^{-1/2} \geq (\beta_1^{-2} + M^2)^{-1/2}.$$

- **Upper bound for the largest eigenvalue of A_n.** Note that

$$\lambda_{\max}(A_n) = \lambda_{\min}\left(\beta_{n+1}^{-2} I_d + \mathrm{Diag}\left(V_n \right) \right)^{-1/2} \leq \beta_{n+1}.$$

Verifying Assumption 3(i) for Adagrad. Using the tower property, we have:

$$\mathbb{E}\left[\left\langle \nabla V(\theta_n) \middle| A_n H_{\theta_n}(X_{n+1}) \right\rangle \right] = \mathbb{E}\left[\mathbb{E}\left[\left\langle \nabla V(\theta_n) \middle| A_n H_{\theta_n}(X_{n+1}) \right\rangle \middle| \mathcal{F}_n \right] \right],$$

where $(\mathcal{F}_n)_{n \geq 0}$ represents the filtration generated by the random variables $(\theta_0, \{X_k\}_{k \leq n})$. Let $\tilde{A}_n$ be an adaptive $\mathcal{F}_n$-measurable matrix. Then,

$$\mathbb{E}\left[\left\langle \nabla V(\theta_n) \middle| A_n H_{\theta_n}(X_{n+1}) \right\rangle \middle| \mathcal{F}_n \right] = \underbrace{\left\langle \nabla V(\theta_n) \middle| \tilde{A}_n \mathbb{E}\left[H_{\theta_n}(X_{n+1}) \middle| \mathcal{F}_n \right] \right\rangle}_{\text{Treated as in SGD but with } \lambda_{\min}(\tilde{A}_n)}$$
$$+ \underbrace{\mathbb{E}\left[\left\langle \nabla V(\theta_n) \middle| (A_n - \tilde{A}_n) H_{\theta_n}(X_{n+1}) \right\rangle \middle| \mathcal{F}_n \right]}_{\text{Control error between } A_n \text{ and } \tilde{A}_n}.$$

We only verify Assumption H3(i) for Adagrad algorithm since it is analogous to RMSProp. Consider A_n given by:

$$A_n = \left(\mathrm{diag}\left(\beta_{n+1}^{-2} I_d + \frac{1}{n+1} \sum_{k=0}^{n} H_{\theta_k}(X_{k+1}) H_{\theta_k}(X_{k+1})^\top \right) \right)^{-1/2}.$$

First, writing

$$\tilde{A}_n = \left(\operatorname{diag} \left(\beta_{n+1}^{-2} I_d + \frac{1}{n+1} \sum_{k=0}^{n-1} H_{\theta_k}(X_{k+1}) H_{\theta_k}(X_{k+1})^\top \right) \right)^{-1/2}$$

and denoting by $A[i]$ the i-th element of the diagonal of a matrix A, we have

$$A_n[i] - \tilde{A}_n[i] = u_n^{-1/2} \left(v_n^{1/2} - u_n^{1/2} \right) v_n^{-1/2} \leq 0,$$

where $u_n = \beta_{n+1}^{-2} + \sum_{k=0}^{n} \left(H_{\theta_k}(X_{k+1})[i] \right)^2 / (n+1)$ and $v_n = \beta_{n+1}^{-2} + \sum_{k=0}^{n-1} \left(H_{\theta_k}(X_{k+1})[i] \right)^2 / (n+1)$. Then, since $u_n \geq v_n$,

$$A_n[i] - \tilde{A}_n[i] = \frac{v_n - u_n}{\sqrt{u_n v_n} \left(\sqrt{u_n} + \sqrt{v_n} \right)} \geq -\frac{1}{n+1} \left(H_{\theta_n}(X_{n+1})[i] \right)^2 \frac{1}{2 {v_n}^{3/2}} \geq -\frac{\beta_{n+1}^3}{n+1} \left(H_{\theta_n}(X_{n+1})[i] \right)^2 .$$

Since the bias of $H_{\theta_n}(X_{n+1})$ is bounded by $\tilde{b}_n := C_\alpha n^{-\alpha}$,

$$\begin{aligned}
&\mathbb{E}\left[\langle \nabla V(\theta_n) | A_n H_{\theta_n}(X_{n+1}) \rangle | \mathcal{F}_n \right] \\
&= \left\langle \nabla V(\theta_n) | \tilde{A}_n \mathbb{E}\left[H_{\theta_n}(X_{n+1}) | \mathcal{F}_n \right] \right\rangle + \mathbb{E}\left[\left\langle \nabla V(\theta_n) | (A_n - \tilde{A}_n) H_{\theta_n}(X_{n+1}) \right\rangle | \mathcal{F}_n \right] \\
&\geq \lambda_{\min}\left(\tilde{A}_n \right) \|\nabla V(\theta_n)\|^2 - \lambda_{\max}\left(\tilde{A}_n \right) \|\nabla V(\theta_n)\| \tilde{b}_n - \|\nabla V(\theta_n)\| \frac{\beta_{n+1}^3}{n+1} \mathbf{E}\left[\left\| H_{\theta_n}(X_{n+1}) \right\|^3 | \mathcal{F}_n \right].
\end{aligned}$$

As $H_{\theta_n}(X_{n+1})$ and the gradient of V are uniformly bounded by M, $\lambda_{\min}(\tilde{A}_n) \geq (\beta_1^{-2} + M^2)^{-1/2}$, so that

$$\mathbb{E}\left[\langle \nabla V(\theta_n) | A_n H_{\theta_n}(X_{n+1}) \rangle | \mathcal{F}_n \right] \geq \frac{1}{\sqrt{\beta_1^{-2} + M^2}} \|\nabla V(\theta_n)\|^2 - \beta_{n+1} M \tilde{b}_n - M^4 \frac{\beta_{n+1}^3}{n+1},$$

and Assumption H3(i) is satisfied with $\lambda_{n+1} = (\beta_1^{-2} + M^2)^{-1/2}$ and $r_{n+1} = M \beta_{n+1}^2 \tilde{b}_n^2 / \lambda_{n+1} + M^4 \beta_{n+1}^3 / (n+1)$.

A.6 Proof of Theorem 4.7

Proof. The proof of this theorem is inspired by Reddi et al. (2018) and Tong et al. (2022), considering biased gradient estimators and decreasing step sizes. For simplicity, we use $\hat{V}_k = \max(\hat{V}_{k-1}, V_k)$ instead of V_k, which is known as AMSGRAD (Reddi et al., 2018). The operation $\max(D_1, D_2)$ for diagonal matrices D_1 and D_2 is defined as the matrix formed by taking the maximum between the diagonal elements of D_1 and D_2. Let $\tilde{\theta}_{k+1} = \theta_{k+1} + \kappa(\theta_{k+1} - \theta_k)$, for $k \geq 1, \kappa \in [0, 1)$ and $m_k = \rho_1 m_{k-1} + (1 - \rho_1) g_k$ with $g_k = H_{\theta_k}(X_{k+1})$. Using the recursion of Adam, we have:

$$\begin{aligned}
\tilde{\theta}_{k+1} - \tilde{\theta}_k = (1+\kappa)\theta_{k+1} - (1+2\kappa)\theta_k + \kappa\theta_{k-1} &= (1+\kappa)(\theta_{k+1} - \theta_k) - \kappa(\theta_k - \theta_{k-1}) \\
&= -(1+\kappa)\gamma_{k+1} A_k m_k + \kappa \gamma_k A_{k-1} m_{k-1}.
\end{aligned}$$

Choosing $\kappa = \rho_1 / (1 - \rho_1)$, we can rewrite it as:

$$\tilde{\theta}_{k+1} - \tilde{\theta}_k = \kappa \left(\gamma_k A_{k-1} - \gamma_{k+1} A_k \right) m_{k-1} - \gamma_{k+1} A_k g_k.$$

By Assumption H2, V is L-smooth, using the recursion of Adam together with a Taylor expansion with $\tilde{\theta}_k$, we obtain:

$$\begin{aligned}
V\left(\tilde{\theta}_{k+1} \right) &\leq V\left(\tilde{\theta}_k \right) + \left\langle \nabla V\left(\tilde{\theta}_k \right) \mid \tilde{\theta}_{k+1} - \tilde{\theta}_k \right\rangle + \frac{L}{2} \left\| \tilde{\theta}_{k+1} - \tilde{\theta}_k \right\|^2 \\
&\leq V\left(\tilde{\theta}_k \right) - \gamma_{k+1} \left\langle \nabla V\left(\tilde{\theta}_k \right) \mid A_k g_k \right\rangle + \kappa \left\langle \nabla V\left(\tilde{\theta}_k \right) \mid \left(\gamma_k A_{k-1} - \gamma_{k+1} A_k \right) m_{k-1} \right\rangle + L \gamma_{k+1}^2 \|A_k g_k\|^2 \\
&\quad + L \kappa^2 \left\| \left(\gamma_k A_{k-1} - \gamma_{k+1} A_k \right) m_{k-1} \right\|^2 \\
&\leq V\left(\tilde{\theta}_k \right) + T_{1,k} + T_{2,k} + T_{3,k} + T_{4,k},
\end{aligned}$$

where

$$\begin{aligned}
T_{1,k} &= -\gamma_{k+1} \langle \nabla V(\theta_k) \mid A_k g_k \rangle + L \gamma_{k+1}^2 \|A_k g_k\|^2 , \\
T_{2,k} &= -\gamma_{k+1} \left\langle \nabla V\left(\tilde{\theta}_k \right) - \nabla V(\theta_k) \mid A_k g_k \right\rangle , \\
T_{3,k} &= \kappa \left\langle \nabla V\left(\tilde{\theta}_k \right) \mid \left(\gamma_k A_{k-1} - \gamma_{k+1} A_k \right) m_{k-1} \right\rangle , \\
T_{4,k} &= L \kappa^2 \left\| \left(\gamma_k A_{k-1} - \gamma_{k+1} A_k \right) m_{k-1} \right\|^2 .
\end{aligned}$$

Note first that

$$\sum_{k=1}^{n} \mathbb{E}\left[T_{1,k}\right] = -\sum_{k=1}^{n} \gamma_{k+1} \mathbb{E}\left[\langle \nabla V\left(\theta_k\right) \mid A_k g_k \rangle\right] + L \sum_{k=1}^{n} \gamma_{k+1}^2 \mathbb{E}\left[\|A_k g_k\|^2\right]$$
$$\leq -C_\lambda \sum_{k=1}^{n} \gamma_{k+1} \mathbb{E}\left[\|\nabla V\left(\theta_k\right)\|^2\right] + C_\lambda \sum_{k=1}^{n} \gamma_{k+1} r_{k+1} + L \sum_{k=1}^{n} \gamma_{k+1}^2 \mathbb{E}\left[\|A_k g_k\|^2\right],$$

where $C_\lambda = (\delta + M^2)^{-1/2}$.

For the second term, using the inequality $xy \leq x^2/2 + y^2/2$ for all x, y, and the smoothness of V, we get:

$$\begin{aligned}
\sum_{k=1}^{n} \mathbb{E}\left[T_{2,k}\right] &= -\sum_{k=1}^{n} \mathbb{E}\left[\left\langle \nabla V\left(\tilde{\theta}_k\right) - \nabla V\left(\theta_k\right) \mid \gamma_{k+1} A_k g_k \right\rangle\right] \\
&\leq \frac{1}{2} \sum_{k=1}^{n} \mathbb{E}\left[\left\| \nabla V\left(\tilde{\theta}_k\right) - \nabla V\left(\theta_k\right) \right\|^2\right] + \frac{1}{2} \sum_{k=1}^{n} \mathbb{E}\left[\|\gamma_{k+1} A_k g_k\|^2\right] \\
&\leq \frac{L^2}{2} \sum_{k=1}^{n} \mathbb{E}\left[\left\|\tilde{\theta}_k - \theta_k\right\|^2\right] + \sum_{k=1}^{n} \frac{\gamma_{k+1}^2}{2} \mathbb{E}\left[\|A_k g_k\|^2\right] \\
&\leq \frac{\kappa^2 L^2}{2} \sum_{k=1}^{n} \mathbb{E}\left[\|\theta_k - \theta_{k-1}\|^2\right] + \sum_{k=1}^{n} \frac{\gamma_{k+1}^2}{2} \mathbb{E}\left[\|A_k g_k\|^2\right] \\
&\leq \frac{\kappa^2 L^2}{2} \sum_{k=1}^{n} \gamma_k^2 \mathbb{E}\left[\|A_{k-1} m_{k-1}\|^2\right] + \sum_{k=1}^{n} \frac{\gamma_{k+1}^2}{2} \mathbb{E}\left[\|A_k g_k\|^2\right].
\end{aligned}$$

For the third term, using the boundedness of the gradient of V and the fact that $\|m_k\| \leq M$ by Lemma A.3, we have:

$$\begin{aligned}
\sum_{k=1}^{n} \mathbb{E}\left[T_{3,k}\right] &= \kappa \sum_{k=1}^{n} \mathbb{E}\left[\left\langle \nabla V\left(\tilde{\theta}_k\right) \mid \left(\gamma_k A_{k-1} - \gamma_{k+1} A_k\right) m_{k-1} \right\rangle\right] \\
&\leq \kappa M^2 \sum_{i=1}^{d} \sum_{k=1}^{n} \mathbb{E}\left[\gamma_k A_{k-1}[i] - \gamma_{k+1} A_k[i]\right] \\
&\leq \kappa M^2 \sum_{i=1}^{d} \mathbb{E}\left[\gamma_1 A_0[i] - \gamma_{n+1} A_n[i]\right] \leq \kappa M^2 d C_\gamma,
\end{aligned}$$

where in the second inequality, we used the fact that γ_k and A_k are decreasing since we use $\hat{V}_k = \max(\hat{V}_{k-1}, V_k)$. For the last term, using the boundedness of the gradient of V yields:

$$\begin{aligned}
\sum_{k=1}^{n} \mathbb{E}\left[T_{4,k}\right] &= L\kappa^2 \sum_{k=1}^{n} \mathbb{E}\left[\left\|\left(\gamma_k A_{k-1} - \gamma_{k+1} A_k\right) m_{k-1}\right\|^2\right] \\
&\leq L\kappa^2 M^2 \sum_{i=1}^{d} \sum_{k=1}^{n} \mathbb{E}\left[\left(\gamma_k A_{k-1}[i] - \gamma_{k+1} A_k[i]\right)^2\right] \\
&\leq L\kappa^2 M^2 \sum_{i=1}^{d} \sum_{k=1}^{n} \mathbb{E}\left[\left(\gamma_k A_{k-1}[i]\right)^2 - \left(\gamma_{k+1} A_k[i]\right)^2\right] \\
&\leq L\kappa^2 M^2 d C_\gamma^2,
\end{aligned}$$

where we used the inequality $(x - y)^2 \leq x^2 - y^2$ when $x \geq y$ in the second last inequality.

Combining all these terms, we finally obtain:

$$C_\lambda \sum_{k=1}^{n} \gamma_{k+1} \mathbb{E}\left[\|\nabla V\left(\theta_k\right)\|^2\right] \leq V^* + C_\lambda \sum_{k=1}^{n} \gamma_{k+1} r_{k+1} + L \sum_{k=1}^{n} \gamma_{k+1}^2 \mathbb{E}\left[\|A_k g_k\|^2\right] + \sum_{k=1}^{n} \frac{\gamma_{k+1}^2}{2} \mathbb{E}\left[\|A_k g_k\|^2\right]$$
$$+ \kappa M^2 d C_\gamma + L\kappa^2 M^2 d C_\gamma^2 + \frac{\kappa^2 L^2}{2} \sum_{k=1}^{n} \gamma_k^2 \mathbb{E}\left[\|A_{k-1} m_{k-1}\|^2\right]$$

where $V^* = \mathbb{E}[V(\theta_0) - V(\theta^*)] \geq \mathbb{E}[V(\theta_0) - V(\tilde{\theta}_{n+1})]$. Choosing $\gamma_n = n^{-1/2}$ and using Lemma A.3 and Chen et al. (2022, Lemma 24) yields

$$\sum_{k=1}^{n} \gamma_{k+1}^2 \mathbb{E}\left[\|A_k m_k\|^2\right] \leq (1-\rho_1) \sum_{k=1}^{n} \gamma_{k+1}^2 \mathbb{E}\left[\|A_k g_k\|^2\right] \leq (1-\rho_1) d C_\gamma^2 \log\left(1 + \frac{nM^2}{\delta}\right) = O\left(d \log n\right).$$

Therefore, by dividing both sides by $C_\lambda n^{-1/2}$, we obtain

$$\frac{1}{n} \sum_{k=1}^{n} \mathbb{E}\left[\|\nabla V(\theta_k)\|^2\right] = O\left(\frac{1}{\sqrt{n}} + \frac{d \log n}{\sqrt{n}} + \frac{d}{\sqrt{n}} + b_n\right),$$

which concludes the proof. □

Lemma A.3. *Let $\gamma_{k+1} \leq \gamma_k$ for all $k \geq 0$ and let A_k be the adaptive matrix defined by $(\delta I_d + \hat{V}_k)^{-1/2}$, where $\hat{V}_k = \max(\hat{V}_{k-1}, V_k)$, similar to AMSGRAD. Assume that $\rho_1 \in [0, 1)$. Then, for all $k \geq 0$:*

$$\|m_k\|^2 \leq M^2 \quad \text{and} \quad \sum_{k=1}^{n} \gamma_{k+1}^2 \mathbb{E}\left[\|A_k m_k\|^2\right] \leq (1-\rho_1) \sum_{k=1}^{n} \gamma_{k+1}^2 \|A_k g_k\|^2.$$

Proof. For the first inequality, we have:

$$\|m_k\| = \left\|(1-\rho_1) \sum_{\ell=1}^{k} \rho_1^{k-\ell} g_\ell\right\| \leq (1-\rho_1) \sum_{\ell=1}^{k} \rho_1^{k-\ell} \|g_\ell\| \leq M(1-\rho_1) \sum_{\ell \geq 0} \rho_1^{\ell} \leq M,$$

where we used the fact that $\sum_{\ell \geq 0} \rho_1^\ell = 1/(1-\rho_1)$. For the second inequality, using the fact that γ_k and A_k are decreasing since we use $\hat{V}_k = \max(\hat{V}_{k-1}, V_k)$, we can write:

$$\begin{aligned}
\sum_{k=1}^{n} \gamma_{k+1}^2 \|A_k m_k\|^2 = \sum_{k=1}^{n} \gamma_{k+1}^2 \left\|A_k (1-\rho_1) \sum_{\ell=1}^{k} \rho_1^{k-\ell} g_\ell\right\|^2 &\leq (1-\rho_1)^2 \sum_{k=1}^{n} \gamma_{k+1}^2 \sum_{\ell=1}^{k} \rho_1^{k-\ell} \|A_\ell g_\ell\|^2 \\
&\leq (1-\rho_1)^2 \sum_{k=1}^{n} \sum_{\ell=1}^{k} \rho_1^{k-\ell} \gamma_{\ell+1}^2 \|A_\ell g_\ell\|^2 \\
&\leq (1-\rho_1)^2 \sum_{\ell=1}^{n} \sum_{k=\ell}^{n} \rho_1^{k-\ell} \gamma_{\ell+1}^2 \|A_\ell g_\ell\|^2,
\end{aligned}$$

which concludes the proof. □

A.7 The Impact of regularization parameter δ in Adam

In our case, we have a dependence on δ in the logarithm, which is common for adaptive algorithms. The regularization parameter δ, originally introduced to avoid the zero denominator issue when V_k approaches 0, is often overlooked. However, it has been empirically observed that the performance of adaptive methods can be sensitive to the choice of this parameter, especially when a very small δ is used, which has resulted in performance issues in some applications.

In practice, δ is typically chosen as 10^{-8}. In our convergence rate analysis, even though the logarithm of δ^{-1} is small, it still impacts the convergence rate. A larger δ will lead to a better convergence rate, while a smaller δ will preserve stronger adaptivity. We need to find a better compromise between the convergence rate and the adaptivity to choose δ. In (Zaheer et al., 2018; Reddi et al., 2018; Tong et al., 2022), it was shown that by choosing δ between 10^{-3} and 10^{-1}, better results were obtained in some applications of deep learning.

Furthermore, several modified versions of Adam have been proposed, such as AMSGRAD (Zaheer et al., 2018) and YOGI (Reddi et al., 2018) with the discussion of the regularization parameter δ. The authors of Tong et al. (2022) proposed a new modified version of Adam called SADAM to represent the calibrated ADAM using the softplus function. In this algorithm, they define $\hat{V}_k = \text{softplus}\left(\sqrt{V_k}\right)$ while other terms remain unchanged. Since we have:

$$\hat{V}_k = \text{softplus}\left(\sqrt{V_k}\right) = \frac{1}{b} \log\left(1 + e^{b\sqrt{V_k}}\right) \approx \frac{1}{b} \log\left(e^{b\sqrt{V_k}}\right) = \sqrt{V_k},$$

where b is the parameter to control for achieving a better convergence rate. In this case, we have $\lambda_{\max}(A_k) \leq b/\log 2$, which is similar to $\delta^{-1/2}$ in Adagrad and Adam. Additionally, they demonstrate that $b \approx 50$ appears to be a good choice based on the empirical observations.

B IWAE / BR-IWAE

B.1 Importance Weighted Autoencoder (IWAE)

In this section, we elaborate on the IWAE procedure within our framework to illustrate its convergence rate. First, let's recall some basics of IWAE. The IWAE objective function is defined as:

$$\mathcal{L}_k^{\mathrm{IWAE}}(\theta, \phi; x) = \mathbb{E}_{q_\phi^{\otimes k}(\cdot|x)}\left[\log \frac{1}{k}\sum_{\ell=1}^{k} \frac{p_\theta(x, Z^{(\ell)})}{q_\phi(Z^{(\ell)}|x)}\right],$$

where k corresponds to the number of samples drawn from the encoder's approximate posterior distribution. Denoting V as the objective function, i.e., $V(\theta) = \log p_\theta(x)$, the gradient of V and the estimator of the gradient of the ELBO of the IWAE objective are given by:

$$\begin{aligned} \nabla_\theta V(\theta) &= \nabla_\theta \log p_\theta(x) = \mathbb{E}_{p_\theta(\cdot|x)}\left[\nabla_\theta \log p_\theta(x, z)\right], \\ \widehat{\nabla}_\theta \mathcal{L}_k^{\mathrm{IWAE}}(\theta, \phi; x) &= \sum_{\ell=1}^{k} \frac{w^{(\ell)}}{\sum_{\ell=1}^{k} w^{(\ell)}} \nabla_\theta \log p_\theta(x, z^{(\ell)}), \end{aligned} \tag{9}$$

where $w^{(\ell)} = \frac{p_\theta(x, z^{(\ell)})}{q_\phi(z^{(\ell)}|x)}$ the unnormalized importance weights.

B.1.1 Proof of Theorem 5.1

Proof. The proof is adapted from Agapiou et al. (2017, Theorem 2.1). By definition,

$$\widehat{\nabla}_\theta \mathcal{L}_k^{\mathrm{IWAE}}(\theta, \phi; x) - \nabla_\theta V(\theta) = \frac{\sum_{\ell=1}^{k} w^{(\ell)} \left(\nabla_\theta \log p_\theta(x, z^{(\ell)}) - \mathbb{E}_{p_\theta(\cdot|x)}\left[\nabla_\theta \log p_\theta(x, z)\right]\right)}{\sum_{\ell=1}^{k} w^{(\ell)}}.$$

Writing $\tilde{H}(x, z^{(\ell)}) = \nabla_\theta \log p_\theta(x, z^{(\ell)}) - \mathbb{E}_{p_\theta(\cdot|x)}\left[\nabla_\theta \log p_\theta(x, z)\right]$, yields

$$\hat{\nabla}_\theta \mathcal{L}_k^{\mathrm{IWAE}}(\theta, \phi; x) - \nabla_\theta V(\theta) = \frac{\sum_{\ell=1}^{k} w^{(\ell)} \tilde{H}(x, z^{(\ell)})}{\sum_{\ell=1}^{k} w^{(\ell)}}.$$

Since $\mathbb{E}_{q_\phi(\cdot|x)}[w\tilde{H}(x, z)] = 0$, we have:

$$\widehat{\nabla}_\theta \mathcal{L}_k^{\mathrm{IWAE}}(\theta, \phi; x) - \nabla_\theta V(\theta) = \frac{\frac{1}{k}\sum_{\ell=1}^{k} w^{(\ell)} \tilde{H}(x, z^{(\ell)}) - \mathbb{E}_{q_\phi(\cdot|x)}\left[w\tilde{H}(x, z)\right]}{\frac{1}{k}\sum_{\ell=1}^{k} w^{(\ell)}}.$$

As $\sum_{\ell=1}^{k} w^{(\ell)} \tilde{H}(x, z^{(\ell)})/k$ is an unbiased estimator of $\mathbb{E}_{q_\phi(\cdot|x)}\left[w\tilde{H}(x, z)\right]$,

$$\begin{aligned} &\mathbb{E}_{q_\phi^{\otimes k}(\cdot|x)}\left[\hat{\nabla}_\theta \mathcal{L}_k^{\mathrm{IWAE}}(\theta, \phi; x) - \nabla_\theta V(\theta)\right] \\ &\qquad = \mathbb{E}_{q_\phi^{\otimes k}(\cdot|x)}\left[\left(\frac{1}{\frac{1}{k}\sum_{\ell=1}^{k} w^{(\ell)}} - \frac{1}{\mathbb{E}_{q_\phi(\cdot|x)}[w]}\right)\left(\frac{1}{k}\sum_{\ell=1}^{k} w^{(\ell)} \tilde{H}(x, z^{(\ell)}) - \mathbb{E}_{q_\phi(\cdot|x)}\left[w\tilde{H}(x, z)\right]\right)\right], \end{aligned}$$

so that

$$\begin{aligned} &\mathbb{E}_{q_\phi^{\otimes k}(\cdot|x)}\left[\hat{\nabla}_\theta \mathcal{L}_k^{\mathrm{IWAE}}(\theta, \phi; x) - \nabla_\theta V(\theta)\right] \\ &\qquad = \mathbb{E}_{q_\phi^{\otimes k}(\cdot|x)}\left[\frac{\left(\frac{1}{k}\sum_{\ell=1}^{k} w^{(\ell)} \tilde{H}(x, z^{(\ell)}) - \mathbb{E}_{q_\phi(\cdot|x)}\left[w\tilde{H}(x, z)\right]\right)\left(\mathbb{E}_{q_\phi(\cdot|x)}[w] - \frac{1}{k}\sum_{\ell=1}^{k} w^{(\ell)}\right)}{\mathbb{E}_{q_\phi(\cdot|x)}[w]\,\frac{1}{k}\sum_{\ell=1}^{k} w^{(\ell)}}\right]. \end{aligned}$$

Therefore,

$$\left\|\mathbb{E}_{q_\phi^{\otimes k}(\cdot|x)}\left[\hat{\nabla}_\theta \mathcal{L}_k^{\mathrm{IWAE}}(\theta, \phi; x) - \nabla_\theta V(\theta)\right]\right\| \leq A_1 + A_2,$$

where

$$A_1 = \left\| \mathbb{E}_{q_\phi^{\otimes k}(\cdot|x)} \left[\left(\hat{\nabla}_\theta \mathcal{L}_k^{\mathrm{IWAE}}(\theta, \phi; x) - \nabla_\theta V(\theta) \right) \mathbb{1}_{\left\{ \frac{2}{k} \sum_{\ell=1}^k w^{(\ell)} > \mathbb{E}_{q_\phi(\cdot|x)}[w] \right\}} \right] \right\|,$$
$$A_2 = \left\| \mathbb{E}_{q_\phi^{\otimes k}(\cdot|x)} \left[\left(\widehat{\nabla}_\theta \mathcal{L}_k^{\mathrm{IWAE}}(\theta, \phi; x) - \nabla_\theta V(\theta) \right) \mathbb{1}_{\left\{ \frac{2}{k} \sum_{\ell=1}^k w^{(\ell)} \leq \mathbb{E}_{q_\phi(\cdot|x)}[w] \right\}} \right] \right\|.$$

Note that

$$\begin{aligned}
A_1 &\leq \left\| \mathbb{E}_{q_\phi^{\otimes k}(\cdot|x)} \left[\frac{2}{\mathbb{E}_{q_\phi(\cdot|x)}[w]^2} \left(\frac{1}{k} \sum_{\ell=1}^k w^{(\ell)} \tilde{H}(x, z^{(\ell)}) - \mathbb{E}_{q_\phi(\cdot|x)} \left[w \tilde{H}(x, z) \right] \right) \left(\mathbb{E}_{q_\phi(\cdot|x)}[w] - \frac{1}{k} \sum_{\ell=1}^k w^{(\ell)} \right) \right] \right\| \\
&\leq \frac{2}{\mathbb{E}_{q_\phi(\cdot|x)}[w]^2} \mathbb{E}_{q_\phi^{\otimes k}(\cdot|x)} \left[\left\| \frac{1}{k} \sum_{\ell=1}^k w^{(\ell)} \tilde{H}(x, z^{(\ell)}) - \mathbb{E}_{q_\phi(\cdot|x)} \left[w \tilde{H}(x, z) \right] \right\| \left\| \frac{1}{k} \sum_{\ell=1}^k w^{(\ell)} - \mathbb{E}_{q_\phi(\cdot|x)}[w] \right\| \right] \\
&\leq \frac{2}{\mathbb{E}_{q_\phi(\cdot|x)}[w]^2} \mathbb{E}_{q_\phi^{\otimes k}(\cdot|x)} \left[\left\| \frac{1}{k} \sum_{\ell=1}^k w^{(\ell)} \tilde{H}(x, z^{(\ell)}) - \mathbb{E}_{q_\phi(\cdot|x)} \left[w \tilde{H}(x, z) \right] \right\|^2 \right]^{1/2} \\
&\qquad \times \mathbb{E}_{q_\phi^{\otimes k}(\cdot|x)} \left[\left(\frac{1}{k} \sum_{\ell=1}^k w^{(\ell)} - \mathbb{E}_{q_\phi(\cdot|x)}[w] \right)^2 \right]^{1/2},
\end{aligned}$$

where we used Cauchy-Schwarz inequality in the last inequality. On the other hand,

$$\mathbb{E}_{q_\phi^{\otimes k}(\cdot|x)} \left[\left(\frac{1}{k} \sum_{\ell=1}^k w^{(\ell)} - \mathbb{E}_{q_\phi(\cdot|x)}[w] \right)^2 \right] = \mathbb{V} \left(\frac{1}{k} \sum_{\ell=1}^k w^{(\ell)} \right) \leq \frac{\mathbb{E}_{q_\phi(\cdot|x)} \left[w^2 \right]}{k},$$

and

$$\begin{aligned}
&\mathbb{E}_{q_\phi^{\otimes k}(\cdot|x)} \left[\left\| \frac{1}{k} \sum_{\ell=1}^k w^{(\ell)} \tilde{H}(x, z^{(\ell)}) - \mathbb{E}_{q_\phi(\cdot|x)} \left[w \tilde{H}(x, z) \right] \right\|^2 \right] \\
&\qquad = \mathrm{Tr} \left(\mathbb{V} \left(\frac{1}{k} \sum_{\ell=1}^k w^{(\ell)} \tilde{H}(x, z^{(\ell)}) \right) \right) \leq 4dM^2 \frac{\mathbb{E}_{q_\phi(\cdot|x)} \left[w^2 \right]}{k}.
\end{aligned}$$

Finally, we deduce that

$$A_1 \leq \frac{2}{\mathbb{E}_{q_\phi(\cdot|x)}[w]^2} \frac{1}{\sqrt{k}} \mathbb{E}_{q_\phi(\cdot|x)} \left[w^2 \right]^{1/2} \frac{2\sqrt{d}M}{\sqrt{k}} \mathbb{E}_{q_\phi(\cdot|x)} \left[w^2 \right]^{1/2} = \frac{\mathbb{E}_{q_\phi(\cdot|x)} \left[w^2 \right]}{\mathbb{E}_{q_\phi(\cdot|x)}[w]^2} \frac{4\sqrt{d}M}{k}.$$

Using the assumption on the boundedness of $\left\| \nabla_\theta \log p_\theta(x, z) \right\|$ and the Markov inequality, we obtain:

$$\begin{aligned}
A_2 &\leq 2M\mathbb{P} \left(2 \frac{1}{k} \sum_{\ell=1}^k w^{(\ell)} \leq \mathbb{E}_{q_\phi(\cdot|x)}[w] \right) \\
&\leq 2M\mathbb{P} \left(2 \left(\frac{1}{k} \sum_{\ell=1}^k w^{(\ell)} - \mathbb{E}_{q_\phi(\cdot|x)}[w] \right) \leq -\mathbb{E}_{q_\phi(\cdot|x)}[w] \right) \\
&\leq 2M\mathbb{P} \left(\left| \frac{1}{k} \sum_{\ell=1}^k w^{(\ell)} - \mathbb{E}_{q_\phi(\cdot|x)}[w] \right| \geq \frac{\mathbb{E}_{q_\phi(\cdot|x)}[w]}{2} \right) \leq \frac{\mathbb{E}_{q_\phi(\cdot|x)} \left[w^2 \right]}{\mathbb{E}_{q_\phi(\cdot|x)}[w]^2} \frac{8M}{k},
\end{aligned}$$

which concludes the proof. □

Input: Initial point θ_0, maximum number of iterations n, step sizes $\{\gamma_k\}_{k\geq 1}$ and a hyperparameter $\alpha \geq 0$ to control the bias and MSE.
for $k = 0$ to $n-1$ **do**
Compute the stochastic update $\nabla_{\theta,\phi}\mathcal{L}_{k^\alpha}^{\text{IWAE}}(\theta_k, \phi_k; X_{k+1})$ using k^α samples from the variational posterior distribution and adaptive steps A_k.
Set $\theta_{k+1} = \theta_k - \gamma_{k+1}A_k\nabla_\theta\mathcal{L}_{k^\alpha}^{\text{IWAE}}(\theta_k, \phi_k; X_{k+1})$ and $\phi_{k+1} = \phi_k - \gamma_{k+1}A_k\nabla_\phi\mathcal{L}_{k^\alpha}^{IWAE}(\theta_k, \phi_k; X_{k+1})$.
end for
Output: $(\theta_k)_{1\leq k\leq n}$

Algorithm 2: Adaptive Stochastic Approximation for IWAE

B.2 BR-IWAE

In this section, we provide additional details on the Biased Reduced Importance Weighted Autoencoder (BR-IWAE). In IWAE, instead of estimating the gradient of the ELBO with respect to θ via the Monte Carlo method, we estimate the gradient of the true objective function $\mathbb{E}_{p_\theta(\cdot|x)}[\nabla_\theta \log p_\theta(x,z)]$ using the BR-SNIS estimator (Cardoso et al., 2022). This estimator aims to reduce the bias of self-normalized importance sampling estimators without increasing the variance.

Input: Maximum number of iterations $t_{\max}$ for MCMC and number of samples k from the variational distribution $q_\phi(\cdot \mid x)$.
Initialization: Draw $\tilde{z}_0$ from the variational distribution $q_\phi(\cdot \mid x)$.
for $t = 0$ to $t_{\max} - 1$ **do**
Draw $I_{t+1} \in \{1, \ldots, k\}$ uniformly at random and set $z_{t+1}^{I_{t+1}} = \tilde{z}_t$.
Draw $z_{t+1}^{1:k\setminus\{I_{t+1}\}}$ independently from the variational distribution $q_\phi(\cdot \mid x)$.
Compute the unnormalized importance weights:

$$w_{t+1}^{(\ell)} = \frac{p_\theta(x, z_{t+1}^{(\ell)})}{q_\phi(z_{t+1}^{(\ell)}|x)} \quad \forall \ell \in \{1, \ldots, k\}.$$

Normalize importance weights:

$$\omega_{t+1}^{(\ell)} = \frac{w_{t+1}^{(\ell)}}{\sum_{\ell=1}^{N} w_{t+1}^{(\ell)}} \quad \forall \ell \in \{1, \ldots, k\}.$$

Select $\tilde{z}_{t+1}$ from the set $z_{t+1}^{1:k}$ by choosing z_{t+1}^{ℓ} with probability $\omega_{t+1}^{(\ell)}$.
end for
Output: $\left(z_t^{1:k}\right)_{1\leq t\leq t_{\max}}$ and $\left(\omega_t^{1:k}\right)_{1\leq t\leq t_{\max}}$.

Algorithm 3: BR-IWAE Gradient Estimator

The BR-SNIS estimator of $\mathbb{E}_{p_\theta(\cdot|x)}[\nabla_\theta \log p_\theta(x,z)]$ is given by:

$$\widehat{\nabla}_\theta \log p_\theta(x, z_{t_0:t_{\max}}^{1:k}) = \frac{1}{t_{\max} - t_0} \sum_{t=t_0+1}^{t_{\max}} \sum_{\ell=1}^{k} \omega_t^{(\ell)} \nabla_\theta \log p_\theta(x, z_t^\ell),$$

where t_0 corresponds to a burn-in period. By Cardoso et al. (2022, Theorem 4) the bias of this estimator decreases exponentially with t_0. The BR-IWAE algorithm proceeds in two steps, which are repeated during optimization:

- Update the parameter ϕ as in the IWAE algorithm, that is, for all $n \geq 1$:

$$\phi_{n+1} = \phi_n - \gamma_{n+1}A_n\nabla_\phi\mathcal{L}_k^{\text{IWAE}}(\theta_n, \phi_n; X_{n+1}).$$

- Update the parameter θ by estimating (8) using BR-SNIS as detailed in Algorithm 3:

$$\phi_{n+1} = \phi_n - \gamma_{n+1}A_n\widehat{\nabla}_\theta \log p_\theta(X_{n+1}, z_{t_0:t_{\max}}^{1:k}).$$

B.3 Some Other Techniques for Reducing Bias

In the previous section, we discussed one technique for reducing bias, BR-IWAE. Here, we provide an overview of some other bias reduction techniques within our context. First, the jackknife bias-corrected estimator (Nowozin, 2018)

is defined as:

$$\mathcal{L}^{\text{Jackknife}}(\theta, \phi; x) = k\mathcal{L}_k^{\text{IWAE}}(\theta, \phi; x) - (k-1)\mathcal{L}_{k-1}^{\text{IWAE}}(\theta, \phi; x),$$

which achieves a reduced bias of $O(k^{-2})$. This can also be generalized to have a bias of order $O(k^{-m})$ for some $m \geq 1$ by considering:

$$\mathcal{L}_{k,m}^{\text{Jackknife}} = \sum_{j=0}^{m} c(k, m, j)\mathcal{L}_{k-j}^{\text{IWAE}},$$

where the coefficients $c(k, m, j)$ are given as

$$c(k, m, j) = (-1)^j \frac{(k-j)^m}{(m-j)!\, j!}.$$

The Delta method Variational Inference (DVI) (Teh et al., 2006) is defined by:

$$\mathcal{L}_k^{DVI} = \mathbb{E}_{q_\phi^{\otimes k}(\cdot|x)}\left[\log \frac{1}{k}\sum_{\ell=1}^{k} w^{(\ell)} + \frac{\bar{s}_k^2}{2k\bar{w}_k}\right],$$

where

$$w^{(\ell)} = \frac{p_\theta(x, z^{(\ell)})}{q_\phi(z^{(\ell)}|x)}, \quad \bar{w}_k = \frac{1}{k}\sum_{\ell=1}^{k} w^{(\ell)} \quad \text{and} \quad \bar{s}_k^2 = \frac{1}{k-1}\sum_{\ell=1}^{k}(w^{(\ell)} - \bar{w}_k)^2.$$

The Monte Carlo estimator of the Delta method Variational Inference objective achieves a reduced bias of $O(k^{-2})$. Some other techniques for reducing bias include the iterated bootstrap for bias correction, the debiasing lemma (McLeish, 2011), and Multi-Level Monte Carlo and its variants (Hu et al., 2021b).

C Implications of Our Theorem in Bilevel Optimization and Conditional Stochastic Optimization

C.1 Stochastic Bilevel Optimization

We consider the Stochastic Bilevel Optimization problem given by:

$$\min_{\theta \in \mathbb{R}^d} V(\theta) = \mathbb{E}_\xi\left[f(\theta, \phi^*(\theta); \xi)\right] \quad \text{(upper-level)} \tag{10}$$

subject to

$$\phi^*(\theta) \in \underset{\phi \in \mathbb{R}^q}{\operatorname{argmin}} \mathbb{E}_\zeta\left[g(\theta, \phi; \zeta)\right] \quad \text{(lower-level)}$$

where the upper and inner level functions f and g are both jointly continuously differentiable and ξ and ζ are random variables. The goal of equation (10) is to minimize the objective function V with respect to θ, where $\phi^*(\theta)$ is obtained by solving the lower-level minimization problem. This bilevel problem involves many machine learning problems with a hierarchical structure, which include hyper-parameter optimization Franceschi et al. (2018), metalearning Finn et al. (2017), policy optimization Hong et al. (2023) and neural network architecture search Liu et al. (2018). The gradient of the objective function V is given by:

$$\nabla V(\theta) = \nabla_\theta f(\theta, \phi^*(\theta)) - \nabla_{\theta\phi} g(\theta, \phi^*(\theta))v^*,$$

where v^* is the solution of the following linear system:

$$\nabla_\phi^2 g(\theta, \phi^*(\theta))v = \nabla_\phi f(\theta, \phi^*(\theta)).$$

Instead of computing v^*, the solution of the linear system above, Ji et al. (2021); Chen et al. (2021) proposes a method to estimate v^*. This estimation introduces bias in the gradient of the objective function.

H6 For all $\theta \in \mathbb{R}^d$, $g(\theta, \phi)$ is strongly convex with respect to ϕ with parameter $\mu_g > 0$.

H7 (Regularity Lipschitz condition) Assume that $f, \nabla f, \nabla g, \nabla^2 g$ are respectively Lipschitz continuous with Lipschitz constants $\ell_{f,0}$, $\ell_{f,1}$, $\ell_{g,1}$ and $\ell_{g,2}$.

Assumptions H6 and H7 are the same assumptions used in Chen et al. (2021) to obtain the convergence results with SGD. Furthermore, these two assumptions ensure that the first- and second-order derivatives of f and g, as well as the solution mapping $\phi^*(\theta)$, are well-behaved.

Proposition C.1. *(Chen et al. (2021, Lemma 2.2)) Under Assumption 6, we have:*

$$\nabla V(\theta) = \nabla_\theta f\left(\theta, \phi^*(\theta)\right) - \nabla^2_{\theta\phi} g\left(\theta, \phi^*(\theta)\right)\left[\nabla^2_\phi g\left(\theta, \phi^*(\theta)\right)\right]^{-1} \nabla_\phi f\left(\theta, \phi^*(\theta)\right).$$

Due to the dependence of the minimizer of the lower-level problem $\phi^*(\theta)$, obtaining an unbiased estimate of $\nabla V(\theta)$ is challenging. To address this, we replace $\phi^*(\theta)$ in the gradient with y and define

$$\bar{\nabla}_\theta f(\theta, \phi) := \nabla_\theta f(\theta, \phi) - \nabla^2_{\theta\phi} g(\theta, \phi)\left[\nabla^2_\phi g(\theta, \phi)\right]^{-1} \nabla_\phi f(\theta, \phi).$$

Furthermore, by estimating $\left[\nabla^2_\phi g(\theta, \phi)\right]^{-1}$, we define the stochastic update H_k (Chen et al., 2021) as follows:

$$H_k = \nabla_\theta f\left(\theta_k, \phi_{k+1}; \xi_k\right) - \nabla^2_{\theta\phi} g\left(\theta_k, \phi; \zeta_k^{(0)}\right)\left[\frac{N}{\ell_{g,1}} \prod_{i=1}^{N'}\left(I - \frac{1}{\ell_{g,1}} \nabla^2_\phi g\left(\theta_k, \phi_{k+1}; \zeta_k^{(i)}\right)\right)\right] \nabla_\phi f\left(\theta_k, \phi_{k+1}; \xi_k\right),$$

where N' is drawn from $\{1, \ldots, N\}$ uniformly at random and $\left\{\zeta^{(1)}, \ldots, \zeta^{(N')}\right\}$ are i.i.d. samples.

Input: Initial points θ_0, ϕ_0, maximum number of iterations for the upper-level n and for the lower-level T and step sizes $\{\gamma_k, \tilde{\gamma}_k\}_{k\geq 1}$.
for $k = 0$ to $n - 1$ **do**
  Set $\phi_{k,0} = \phi_k$.
  for $t = 0$ to $T - 1$ **do**
    $\phi_{k,t+1} = \phi_{k,t} - \tilde{\gamma}_{k+1} \nabla_\phi g\left(\theta_k, \phi_{k,t}; \zeta_{k,t}\right)$
  end for
  Set $\phi_{k+1} = \phi_{k,T}$.
  Compute the stochastic update H_k and adaptive matrix A_k.
  $\theta_{k+1} = \theta_k - \gamma_{k+1} A_k H_k$
end for
Output: $(\theta_k, \phi_k)_{1\leq k\leq n}$

Algorithm 4: Algorithm for the Stochastic Bilevel Optimization

In Algorithm 4, we perform T steps of SGD on the lower-level variable ϕ_k before updating the upper-level variable θ_k using adaptive methods such as Adagrad, RMSProp, or Adam.

Lemma C.2. *(Ghadimi and Wang (2018, Lemma 2.2)) Under Assumptions H6 and H7, for all $(\theta, \theta') \in \left(\mathbb{R}^d\right)^2$, we have:*

$$\left\|\nabla V\left(\theta\right) - \nabla V\left(\theta'\right)\right\| \leq L_V \left\|\theta - \theta'\right\|,$$

with the constant L_V is given by

$$L_V = \ell_{f,1} + \frac{\ell_{g,1}\left(\ell_{f,1} + L_f\right)}{\mu_g} + \frac{\ell_{f,0}}{\mu_g}\left(\ell_{g,2} + \frac{\ell_{g,1}\ell_{g,2}}{\mu_g}\right),$$

and L_f is defined as $L_f = \ell_{f,1} + \frac{\ell_{g,1}\ell_{f,1}}{\mu_g} + \frac{\ell_{f,0}}{\mu_g}\left(\ell_{g,2} + \frac{\ell_{g,1}\ell_{g,2}}{\mu_g}\right)$.

Lemma C.3. *Under Assumptions H6 and H7, the following inequalities hold:*

$$\left\|\nabla V\left(\theta_k\right) - \mathbb{E}\left[H_k \mid \mathcal{F}_k\right]\right\|^2 \leq 2L_f^2 \left\|\phi_{k+1} - \phi^*\left(\theta_k\right)\right\|^2 + 2\tilde{b}_k^2,$$

$$\left\|\bar{\nabla}_\theta f(\theta, \phi)\right\| \leq \ell_{f,0} + \frac{\ell_{g,1}\ell_{f,0}}{\mu_g},$$

where $L_f = \ell_{f,1} + \frac{\ell_{g,1}\ell_{f,1}}{\mu_g} + \frac{\ell_{f,0}}{\mu_g}\left(\ell_{g,2} + \frac{\ell_{g,1}\ell_{g,2}}{\mu_g}\right)$ and $\tilde{b}_k = \ell_{g,1}\ell_{f,1}\frac{1}{\mu_g}\left(1 - \frac{\mu_g}{\ell_{g,1}}\right)^N$.

Proof. For the bias term, we have:

$$\begin{aligned}\|\nabla F(\theta_k) - \mathbb{E}[H_k \mid \mathcal{F}_k]\|^2 &= \left\|\bar{\nabla} f(\theta_k, \phi^*(\theta_k)) - \bar{\nabla} f(\theta_k, \phi_{k+1}) + \bar{\nabla} f(\theta_k, \phi_{k+1}) - \mathbb{E}[H_k \mid \mathcal{F}_k]\right\|^2 \\ &\leq 2\left\|\bar{\nabla} f(\theta_k, \phi^*(\theta_k)) - \bar{\nabla} f(\theta_k, \phi_{k+1})\right\|^2 + 2\left\|\bar{\nabla} f(\theta_k, \phi_{k+1}) - \mathbb{E}[H_k \mid \mathcal{F}_k]\right\|^2 \\ &\leq 2L_f^2\left\|\phi_{k+1} - \phi^*(\theta_k)\right\|^2 + 2\tilde{b}_k^2,\end{aligned}$$

where we used Ghadimi and Wang (2018, Lemma 2.2) for the first term and Hong et al. (2023, Lemma 11) for the second term.

For the second inequality, we have:

$$\begin{aligned}\left\|\bar{\nabla}_\theta f(\theta,\phi)\right\| &= \left\|\nabla_\theta f(\theta,\phi) - \nabla^2_{\theta\phi} g(\theta,\phi)\left[\nabla^2_\phi g(\theta,\phi)\right]^{-1}\nabla_\phi f(\theta,\phi)\right\| \\ &\leq \|\nabla_\theta f(\theta,\phi)\| + \left\|\nabla^2_{\theta\phi} g(\theta,\phi)\right\|\left\|\left[\nabla^2_\phi g(\theta,\phi)\right]^{-1}\right\|\left\|\nabla_\phi f(\theta,\phi)\right\| \\ &\leq \ell_{f,0} + \frac{\ell_{g,1}\ell_{f,0}}{\mu_g}.\end{aligned}$$

□

Theorem C.4. *Consider the Bilevel Optimization problem defined in Algorithm 4. Let $\gamma_n = c_\gamma n^{-1/2}$ and $\tilde{\gamma}_n = c_{\tilde{\gamma}} n^{-1/2}/T$. For any $n \geq 1$, let $R \in \{0, \ldots, n\}$ be a uniformly distributed random variable. Assume the boundedness of the variance of the estimators of ∇f, ∇g, and $\nabla^2 g$. Then, under Assumptions H6 and H7, we have:*

$$\mathbb{E}\left[\|\nabla V(\theta_R)\|^2\right] = O\left(\frac{\log n}{\sqrt{n}} + b_n\right).$$

Proof. By using Lemma C.3, V is smooth and Lemma C.3, the bias and the gradient of V are bounded. Using our Corollary 4.6, we obtain:

$$\mathbb{E}\left[\|\nabla V(\theta_R)\|^2\right] = O\left(\frac{\log n}{\sqrt{n}} + b_n\right),$$

where

$$b_n = O\left(\frac{\sum_{k=0}^n \gamma_{k+1}\tilde{b}_k^2 + \sum_{k=0}^n \gamma_{k+1}\|\phi_{k+1} - \phi^*(\theta_k)\|^2}{\sqrt{n}}\right).$$

Then, with Ghadimi and Wang (2018, Lemma 2.3) and Chen et al. (2021, Lemma 3), we derive:

$$b_n = O\left(\frac{\sum_{k=0}^n \gamma_{k+1}\tilde{b}_k^2}{\sqrt{n}} + \frac{1}{\sqrt{n}}\right).$$

□

C.2 Conditional Stochastic Optimization

We now consider a class of Conditional Stochastic Optimization:

$$\min_{\theta\in\mathbb{R}^d} V(\theta) := \mathbb{E}_\xi\left[f_\xi\left(\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta,\xi)\right]\right)\right], \tag{11}$$

where $f_\xi(\cdot) : \mathbb{R}^q \to \mathbb{R}$ depends on the random vector ξ and $g_\eta(\cdot,\xi) : \mathbb{R}^d \to \mathbb{R}^q$ is a vector-valued function dependent on both random vectors ξ and η. The inner expectation is taken with respect to the conditional distribution of η given ξ. Given certain conditions on the regularity of these functions, the gradient of V as defined in (11) can be expressed as:

$$\nabla V(\theta) = \mathbb{E}_\xi\left[\left(\mathbb{E}_{\eta|\xi}\left[\nabla g_\eta(\theta,\xi)\right]\right)^\top \nabla f_\xi\left(\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta,\xi)\right]\right)\right].$$

Constructing an unbiased stochastic estimator of this gradient can be both costly and, in some cases, impractical. Instead, we opt for a biased estimator of $\nabla V(\theta)$, using just one sample ξ and m i.i.d. samples $\{\eta_j\}_{j=1}^m$ from the conditional distribution of η given ξ:

$$\widehat{\nabla} V(\theta;\xi,\{\eta_j\}_{j=1}^m) := \left(\frac{1}{m}\sum_{j=1}^m \nabla g_{\eta_j}(\theta,\xi)\right)^\top \nabla f_\xi\left(\frac{1}{m}\sum_{j=1}^m g_{\eta_j}(\theta,\xi)\right).$$

H8 For all ξ and η, assume that $f_\xi(\cdot)$, $\nabla f_\xi(\cdot)$, $g_\eta(\cdot,\xi)$, and $\nabla g_\eta(\cdot,\xi)$ are respectively Lipschitz continuous with Lipschitz constants $\ell_{f,0}$, $\ell_{f,1}$, $\ell_{g,0}$ and $\ell_{g,1}$.

H9 For all θ and ξ, we assume that $\mathbb{E}_{\eta|\xi}\left[\left\|g_\eta(\theta,\xi)-\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta,\xi)\right]\right\|^2\right]\le\sigma_g^2$.

Lemma C.5. *(Hu et al. (2020, Lemma 2.2)) Under Assumptions H8 and H9, the following holds:*

$$\left\|\mathbb{E}\left[\widehat{\nabla}V(\theta;\xi,\{\eta_j\}_{j=1}^m)\right]-\nabla V(\theta)\right\|^2\le\frac{\ell_{g,0}^2\ell_{f,1}^2\sigma_g^2}{m}.$$

Theorem C.6. *Consider the Conditional Stochastic Optimization problem defined in* (11)*. Let $\gamma_n=c_\gamma n^{-1/2}$, A_n denote the adaptive matrix in Adam and $\rho_1\in[0,1)$. For any $n\ge1$, let $R\in\{0,\dots,n\}$ be a uniformly distributed random variable. Then, under Assumptions H8 and H9, we have:*

$$\mathbb{E}\left[\|\nabla V(\theta_R)\|^2\right]=O\left(\frac{\log n}{\sqrt{n}}+b_n\right),$$

where b_n is defined by writing m_k as the number of conditional samples at iteration k:

$$b_n=O\left(\frac{\sum_{k=0}^n\frac{m_k}{\sqrt{k}}}{\sqrt{n}}\right).$$

Proof. **Smoothness of V:**

$$\begin{aligned}
\left\|\nabla V(\theta)-\nabla V(\theta')\right\| &= \left\|\mathbb{E}_\xi\left[\left(\mathbb{E}_{\eta|\xi}\left[\nabla g_\eta(\theta,\xi)\right]\right)^T\nabla f_\xi\left(\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta,\xi)\right]\right)\right]-\mathbb{E}_\xi\left[\left(\mathbb{E}_{\eta|\xi}\left[\nabla g_\eta(\theta',\xi)\right]\right)^T\nabla f_\xi\left(\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta',\xi)\right]\right)\right]\right\| \\
&\le \left\|\mathbb{E}_\xi\left[\left(\mathbb{E}_{\eta|\xi}\left[\nabla g_\eta(\theta,\xi)\right]\right)^T\nabla f_\xi\left(\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta,\xi)\right]\right)\right]-\mathbb{E}_\xi\left[\left(\mathbb{E}_{\eta|\xi}\left[\nabla g_\eta(\theta',\xi)\right]\right)^T\nabla f_\xi\left(\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta,\xi)\right]\right)\right]\right\| \\
&+ \left\|\mathbb{E}_\xi\left[\left(\mathbb{E}_{\eta|\xi}\left[\nabla g_\eta(\theta',\xi)\right]\right)^T\nabla f_\xi\left(\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta,\xi)\right]\right)\right]-\mathbb{E}_\xi\left[\left(\mathbb{E}_{\eta|\xi}\left[\nabla g_\eta(\theta',\xi)\right]\right)^T\nabla f_\xi\left(\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta',\xi)\right]\right)\right]\right\| \\
&\le \left\|\mathbb{E}_\xi\left[\left(\mathbb{E}_{\eta|\xi}\left[\nabla g_\eta(\theta,\xi)\right]-\mathbb{E}_{\eta|\xi}\left[\nabla g_\eta(\theta',\xi)\right]\right)^T\nabla f_\xi\left(\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta,\xi)\right]\right)\right]\right\| \\
&+ \left\|\mathbb{E}_\xi\left[\left(\mathbb{E}_{\eta|\xi}\left[\nabla g_\eta(\theta',\xi)\right]\right)^T\left(\nabla f_\xi\left(\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta,\xi)\right]\right)-\nabla f_\xi\left(\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta',\xi)\right]\right)\right)\right]\right\| \\
&\le \mathbb{E}_\xi\left[\left\|\mathbb{E}_{\eta|\xi}\left[\nabla g_\eta(\theta,\xi)\right]-\mathbb{E}_{\eta|\xi}\left[\nabla g_\eta(\theta',\xi)\right]\right\|\left\|\nabla f_\xi\left(\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta,\xi)\right]\right)\right\|\right] \\
&+ \mathbb{E}_\xi\left[\left\|\mathbb{E}_{\eta|\xi}\left[\nabla g_\eta(\theta',\xi)\right]\right\|\left\|\nabla f_\xi\left(\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta,\xi)\right]\right)-\nabla f_\xi\left(\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta',\xi)\right]\right)\right\|\right] \\
&\le \ell_{g,1}\ell_{f,0}\left\|\theta-\theta'\right\|+\ell_{g,0}\ell_{f,1}\mathbb{E}_\xi\left[\left\|\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta,\xi)\right]-\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta',\xi)\right]\right\|\right] \\
&\le \ell_{g,1}\ell_{f,0}\left\|\theta-\theta'\right\|+\ell_{g,0}^2\ell_{f,1}\left\|\theta-\theta'\right\|.
\end{aligned}$$

Boundness of ∇V:

$$\begin{aligned}
\|\nabla V(\theta)\| &= \left\|\mathbb{E}_\xi\left[\left(\mathbb{E}_{\eta|\xi}\left[\nabla g_\eta(\theta,\xi)\right]\right)^T\nabla f_\xi\left(\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta,\xi)\right]\right)\right]\right\| \\
&\le \mathbb{E}_\xi\left[\left\|\mathbb{E}_{\eta|\xi}\left[\nabla g_\eta(\theta,\xi)\right]\right\|\left\|\nabla f_\xi\left(\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta,\xi)\right]\right)\right\|\right]\le\ell_{g,0}\ell_{f,0}
\end{aligned}$$

We conclude the proof using Lemma C.5. □

These results can also be extended to the Federated Conditional Stochastic Optimization problem (Wu et al., 2024), which is defined by:

$$\min_{\theta\in\mathbb{R}^d}V(\theta)=\frac{1}{L}\sum_{\ell=1}^L\mathbb{E}_{\xi_\ell}\left[f_{\xi_\ell}^\ell\left(\mathbb{E}_{\eta_l|\xi_\ell}\left[g_{\eta_\ell}^\ell(\theta,\xi_\ell)\right]\right)\right],$$

where $\mathbb{E}_{\xi_\ell} f^\ell_{\xi_\ell}(\cdot) : \mathbb{R}^q \to \mathbb{R}$ is the outer-layer function on the ℓ-th device with the randomness ξ_ℓ, and $\mathbb{E}_{\eta_\ell|\xi_\ell} g^\ell_{\eta_\ell}(\cdot, \xi_\ell) : \mathbb{R}^d \to \mathbb{R}^q$ is the inner-layer function on the ℓ-th device with respect to the conditional distribution of η_ℓ given ξ_ℓ. If the functions $f^\ell_{\xi_\ell}(\cdot)$ and $g^\ell_{\eta_\ell}(\cdot, \xi_\ell)$ for all L devices verify Assumptions H8 and H9, we obtain the same convergence rate.

The following Table 1 provides a comprehensive summary of the key points, including the verification of our assumptions and the convergence results obtained in both Stochastic Bilevel Optimization and Conditional Stochastic Optimization.

Table 1: Stochastic Bilevel Optimization and Conditional Stochastic Optimization with our biased adaptive SA framework.

Applications	Stochastic Bilevel Optimization	Conditional Stochastic Optimization
Problem	$\min_{\theta\in\mathbb{R}^d} V(\theta) = \mathbb{E}_\xi\left[f(\theta, \phi^*(\theta); \xi)\right]$ subject to $\phi^*(\theta) \in \underset{\phi\in\mathbb{R}^q}{\operatorname{argmin}} \mathbb{E}_\zeta\left[g(\theta, \phi; \zeta)\right]$	$\min_{\theta\in\mathbb{R}^d} V(\theta) = \mathbb{E}_\xi\left[f_\xi\left(\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta, \xi)\right]\right)\right]$
Gradient	$\nabla_\theta f(\theta, \phi^*(\theta)) - \nabla_{\theta\phi} g(\theta, \phi^*(\theta)) v^*$	$\mathbb{E}_\xi\left[\left(\mathbb{E}_{\eta|\xi}\left[\nabla g_\eta(\theta, \xi)\right]\right)^T \nabla f_\xi\left(\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta, \xi)\right]\right)\right]$
Lipchitz Constant (A2)	$\ell_{f,1} + \frac{\ell_{g,1}(\ell_{f,1}+L_f)}{\mu_g} + \frac{\ell_{f,0}}{\mu_g}\left(\ell_{g,2} + \frac{\ell_{g,1}\ell_{g,2}}{\mu_g}\right)$	$\ell_{g,1}\ell_{f,0} + \ell^2_{g,0}\ell_{f,1}$
Bias bound (A3)	$\ell_{g,1}\ell_{f,1}\frac{1}{\mu_g}\left(1 - \frac{\mu_g}{\ell_{g,1}}\right)^N$	$\frac{\ell^2_{g,0}\ell^2_{f,1}\sigma^2_g}{m}$
Gradient Bound (A5)	$\ell_{f,0} + \frac{\ell_{g,1}\ell_{f,0}}{\mu_g}$	$\ell_{g,0}\ell_{f,0}$
Convergence	$O\left(\frac{\log n}{\sqrt{n}} + b_n\right)$	$O\left(\frac{\log n}{\sqrt{n}} + b_n\right)$

D Some Other Examples of Biased Gradients with Control on Bias

In this section, we explore examples of applications using biased gradient estimators while having control over the bias.

D.1 Self-Normalized Importance Sampling

Let π be a probability measure on a measurable space $(\mathsf{X}, \mathcal{X})$. The objective is to estimate $\pi(f) = \mathbb{E}_\pi[f(X)]$ for a measurable function $f : \mathcal{X} \to \mathbb{R}^d$ such that $\pi(|f|) < \infty$. Assume that $\pi(\mathrm{d}x) \propto w(x)\lambda(\mathrm{d}x)$, where w is a positive weight function and λ is a proposal probability distribution, and that $\lambda(w) = \int w(x)\lambda(\mathrm{d}x) < \infty$. For a function $f : \mathcal{X} \to \mathbb{R}^d$ such that $\pi(|f|) < \infty$, the identity

$$\pi(f) = \frac{\lambda(\omega f)}{\lambda(\omega)}, \tag{12}$$

leads to the Self-Normalized Importance Sampling (SNIS) estimator:

$$\Pi_N f\left(X^{1:N}\right) = \sum_{i=1}^N \omega^i_N f\left(X^i\right), \quad \omega^i_N = \frac{w\left(X^i\right)}{\sum_{\ell=1}^N w\left(X^\ell\right)},$$

where $X^{1:N} = \left(X^1, \ldots, X^N\right)$ are independent draws from λ and the ω^i_N are called the normalized weights. Agapiou et al. (2017) shows that the bias of SNIS estimator can be expressed as:

$$\left\|\mathbb{E}\left[\Pi_N f\left(X^{1:N}\right) - \pi(f)\right]\right\| \leq \frac{12}{N}\frac{\lambda\left(\omega^2\right)}{\lambda(\omega)^2}.$$

This particular type of estimator can be found in the Importance Weighted Autoencoder (IWAE) framework (Burda et al., 2016), as illustrated in B. The estimator of the gradient of ELBO in IWAE corresponds to the biased SNIS estimator of the gradient of the marginal log likelihood $\log p_\theta(x)$. Consequently, based on these results, both the bias and Mean Squared Error (MSE) are of order $O(1/k)$, where k corresponds to the number of samples drawn from the variational posterior distribution.

D.2 Sequential Monte Carlo Methods

We focus here in the task of estimating the parameters, denoted as θ, in Hidden Markov Models. In this context, the hidden Markov chain is denoted by $(X_t)_{t\geq 0}$. The distribution of X_0 has density χ with respect to the Lebesgue measure μ and for all $t \geq 0$, the conditional distribution of X_{t+1} given $X_{0:t}$ has density $m_\theta(X_t, \cdot)$. It is assumed that this state is partially observed through an observation process $(Y_t)_{0\leq t\leq T}$. The observations $Y_{0:t}$ are assumed to be independent conditionally on $X_{0:t}$ and, for all $0 \leq t \leq T$, the distribution of Y_t given $X_{0:t}$ depends on X_t only and has density $g_\theta(X_t, \cdot)$ with respect to the Lebesgue measure. The joint distribution of hidden states and observations is given by

$$p_\theta(x_{0:T}, y_{0:T}) = \chi(x_0) g_\theta(x_0, y_0) \prod_{t=0}^{T-1} m_\theta(x_t, x_{t+1}) g_\theta(x_{t+1}, y_{t+1}).$$

Our objective is to maximize the likelihood of the model:

$$p_\theta(y_{0:T}) = \int p_\theta(x_{0:T}, y_{0:T}) \, \mathrm{d}x_{0:T}.$$

To use a gradient-based method for this maximization problem, we need to compute the gradient of the objective function. Under simple technical assumptions, by Fisher's identity,

$$\begin{aligned}
\nabla_\theta \log p_\theta(y_{0:T}) &= \int \nabla_\theta \log p_\theta(x_{0:T}, y_{0:T}) p_\theta(x_{0:T}|y_{0:T}) \mathrm{d}x_{0:T} \\
&= \mathbb{E}_{x_{0:T}\sim p_\theta(.|y_{0:T})} \left[\nabla_\theta \log p_\theta(x_{0:T}, y_{0:T})\right] \\
&= \mathbb{E}_{x_{0:T}\sim p_\theta(.|y_{0:T})} \left[\sum_{t=0}^{T-1} s_{t,\theta}(x_t, x_{t+1})\right],
\end{aligned}$$

where $s_{t,\theta}(x, x') = \nabla_\theta \log\{m_\theta(x, x') \, g_\theta(x, y_{t+1})\}$ for $t > 0$ and by convention $s_{0,\theta}(x, x') = \nabla_\theta \log g_\theta(x, y_0)$. Given that the gradient of the log-likelihood represents the smoothed expectation of an additive functional, one may opt for Online Smoothing algorithms to mitigate computational costs. The estimation of the gradient $\nabla_\theta \log p_\theta(y_{0:T})$ is given by:

$$H_\theta(y_{0:T}) = \sum_{i=1}^{N} \frac{\omega_T^i}{\Omega_T} \tau_{T,\theta}^i,$$

where $\{\tau_{T,\theta}^i\}_{i=1}^N$ are particle approximations obtained using particles $\{\left(\xi_T^i, \omega_T^i\right)\}_{i=1}^N$ targeting the filtering distribution ϕ_T, i.e. the conditional distribution of x_T given $y_{0:T}$. In the Forward-only implementation of FFBSm (Del Moral et al., 2010), the particle approximations $\{\tau_{T,\theta}^i\}_{i=1}^N$ are computed using the following formula, with an initialization of $\tau_0^i = 0$ for all $i \in [\![1, N]\!]$:

$$\tau_{t+1,\theta}^i = \sum_{j=1}^{N} \frac{\omega_t^j m_\theta(\xi_t^j, \xi_{t+1}^i)}{\sum_{\ell=1} \omega_t^\ell m_\theta(\xi_t^\ell, \xi_{t+1}^i)} \left\{\tau_{t,\theta}^j + s_{t,\theta}(\xi_t^j, \xi_{t+1}^i)\right\}, \quad t \in \mathbb{N}.$$

The estimator of the gradient $H_\theta(y_{0:T})$ computed by the Forward-only implementation of FFBSm is biased. The bias and MSE of this estimator are of order $O(1/N)$ (Del Moral et al., 2010), where N corresponds to the number of particles used to estimate it. Using alternative recursion methods to compute $\{\tau_{T,\theta}^i\}_{i=1}^N$ results in different algorithms, such as the particle-based rapid incremental smoother (PARIS) (Olsson and Westerborn, 2017) and its pseudo-marginal extension Gloaguen et al. (2022) and Parisian particle Gibbs (PPG) (Cardoso et al., 2023). In such cases, one can also control the bias and MSE of the estimator.

D.3 Policy Gradient for Average Reward over Infinite Horizon

Consider a finite Markov Decision Process (MDP) denoted as $(\mathcal{S}, \mathcal{A}, R, P)$, where $\mathcal{S}$ represents the state space, $\mathcal{A}$ denotes the action space, $R : \mathcal{S}\times\mathcal{A} \to [0, R_{\max}]$ is a reward function, and P is the transition model. The agent's decision-making process is characterized by a parametric family of policies $\{\pi_\theta\}_{\theta\in\mathbb{R}^d}$, employing the soft-max parameterization. The reward function is given by:

$$V(\theta) := \mathbb{E}_{(S,A)\sim\nu_\theta}[\mathrm{R}(S, A)] = \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \nu_\theta(s, a)\mathrm{R}(s, a),$$

where v_θ represents the unique stationary distribution of the state-action Markov Chain sequence $\{(S_t, A_t)\}_{t\geq 1}$ generated by the policy π_θ. Let $\lambda \in (0,1)$ be a discount factor and T be sufficiently large, the estimator of the gradient of the objective function V is given by:

$$H_\theta(S_{1:T}, A_{1:T}) = \mathrm{R}(S_T, A_T) \sum_{i=0}^{T-1} \lambda^i \nabla \log \pi_\theta(A_{T-i}; S_{T-i}),$$

where $(S_{1:T}, A_{1:T}) := (S_1, A_1, \ldots, S_T, A_T)$ is a realization of state-action sequence generated by the policy π_θ. It's important to note that this gradient estimator is biased, and the bias is of order $O(1-\lambda)$ (Karimi et al., 2019).

D.4 Zeroth-Order Gradient

Consider the problem of minimizing the objective function V. The zeroth-order gradient method is particularly valuable in scenarios where direct access to the gradient of the objective function is challenging or computationally expensive. The zeroth-order gradient oracle obtained by Gaussian smoothing (Nesterov and Spokoiny, 2017) is given by:

$$H_\theta(X) = \frac{V(\theta + \tau X) - V(\theta)}{\tau} X, \tag{13}$$

where $\tau > 0$ is a smoothing parameter and $X \sim \mathcal{N}(0, I_d)$ a random Gaussian vector. Nesterov and Spokoiny (2017, Lemma 3) provide the bias of this estimator:

$$\|\mathbb{E}[H_\theta(X)] - \nabla V(\theta)\| \leq \frac{\tau}{2} L (d+3)^{3/2}. \tag{14}$$

The application of these zeroth-order gradient methods can be found in generative adversarial networks (Moosavi-Dezfooli et al., 2017; Chen et al., 2017).

D.5 Compressed Stochastic Approximation: Coordinate Sampling

The coordinate descent method is based on the iteration:

$$\theta_{n+1} = \theta_n - \gamma_{n+1} H_{\theta_n}(X_{n+1})_{j_n} e_{j_n},$$

where $\{e_1, \ldots, e_d\}$ is the canonical basis of $\mathbb{R}^d$ and $H_{\theta_n}(X_{n+1})_j$ is the j-th coordinate of the gradient. The randomized coordinate selection rule chooses j_n uniformly from the set $\{1, 2, \ldots, d\}$. Alternatively, the Gauss-Southwell selection rule (Nutini et al., 2015) uses:

$$j_{n+1} := \underset{j\in\{1,\ldots,d\}}{\operatorname{argmax}} |H_{\theta_n}(X_{n+1})_j|.$$

This corresponds to a greedy selection procedure since at each iteration we choose the coordinate with the largest directional derivative. Another approach to choosing j_n is Coordinate Sampling (Leluc and Portier, 2022), a variant of the stochastic gradient descent algorithm that incorporates a selection step by sampling to perform random coordinate descent. The distribution of ζ_{n+1}, which selects the coordinate, is characterized by the probability weights vector $(w_n^{(1)}, \ldots, w_n^{(d)})$ defined as:

$$w_n^{(j)} = \mathbb{P}(\zeta_{n+1} = j | \mathcal{F}_n), \quad j \in \{1, \ldots, d\}.$$

This distribution of ζ_{n+1} is referred to as the coordinate sampling policy. The Stochastic Coordinate Gradient Descent algorithm is defined by:

$$\theta_{n+1} = \theta_n - \gamma_{n+1} D(\zeta_{n+1}) H_{\theta_n}(X_{n+1}),$$

where $D(k) = e_k e_k^\top \in \mathbb{R}^{d\times d}$ has its entries equal to 0 except for the (k,k) entry, which is 1. Observe that the distribution of the random matrix $D(\zeta_{n+1})$ is fully characterized by the matrix $D_n = \mathbb{E}[D(\zeta_{n+1})|\mathcal{F}_n] = \mathrm{Diag}(w_n^{(1)}, \ldots, w_n^{(d)})$. In this context, A_n represents a diagonal matrix D_n where the diagonal terms characterize the probability weights for sampling each coordinate. These weights typically depend on preceding iterations and even on current gradients. In this case, we always have $\beta_{n+1} \leq 1$ and to control the minimum eigenvalue, we only require a lower bound on the probability weights. This method can be easily extended to incorporate biased gradients and adaptive steps by introducing $\bar{A}_n = D_n A_n$, where A_n represents the adaptive matrix as before, and D_n is the matrix of probability weights.

E Additional Experimental Details

E.1 Experiment with a Synthetic Time-Dependent Bias

In this setup, we consider a simple least squares objective function $V(\theta) = \|A\theta\|^2/2$ in dimension $d = 10$, where A is a positive matrix ensuring convexity. We introduce zero-mean Gaussian noise with variance $\sigma^2 = 0.01$ to every gradient and artificially include the bias term r_n at each iteration. We explore different values of $r_n \in \{1, n^{-1/4}, n^{-1/2}, n^{-1}, n^{-2}, 0\}$, where $r_n = 1$ corresponds to constant bias, $r_n = 0$ for an unbiased gradient, and the others exhibit decreasing bias.

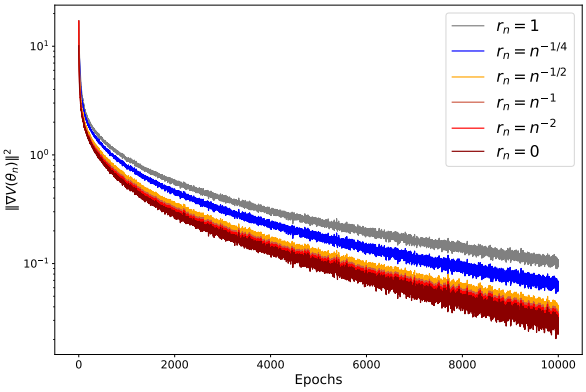

Figure 6: Value of $\|\nabla V(\theta_n)\|^2$ with Adagrad for different values of r_n.

In Figure 6, we observe the convergence rate of the squared norm of the gradient. Similar to Figure 1, we notice the impact of bias on the squared norm of the gradient. When $r \geq 1/2$, we observe nearly the same convergence rate as in the case of an unbiased gradient.

E.2 Additional Experiments in IWAE

In this section, we provide detailed information about the experiments on CIFAR-10. We also conduct additional experiments on the FashionMNIST dataset. For all experiments, we use Adagrad, RMSProp, and Adam with a learning rate decay given by $\gamma_n = C_\gamma/\sqrt{n}$, where $C_\gamma = 0.01$ for Adagrad and $C_\gamma = 0.001$ for RMSProp and Adam. The momentum parameters are set to $\rho_1 = 0.9$ and $\rho_2 = 0.999$, and the regularization parameter δ is fixed at 5×10^{-2}. The impact of this regularization parameter will be illustrated later.

Datasets. We conduct our experiments on two datasets: FashionMNIST (Xiao et al., 2017) and CIFAR-10. The FashionMNIST dataset is a variant of MNIST and consists of 28x28 pixel images of various fashion items, with 60,000 images in the training set and 10,000 images in the test set. CIFAR-10 consists of 32x32 pixel images categorized into 10 different classes. The dataset is divided into 60,000 images in the training set and 10,000 images in the test set.

Models. For FashionMNIST, we use a fully connected neural network with a single hidden layer consisting of 400 hidden units and ReLU activation functions for both the encoder and the decoder. The latent space dimension is set to 20. We use 256 images per iteration (235 iterations per epoch). For CIFAR-10 and CIFAR-100, we use a Convolutional Neural Network (CNN) architecture with 3 Convolutional layers and 2 fully connected layers with ReLU activation functions. The latent space dimension is set to 100. For both datasets, we use 256 images per iteration (196 iterations per epoch).

We estimate the log-likelihood using the VAE, IWAE, and BR-IWAE models, all of which are trained for 100 epochs. Training is conducted using the SGD, SGD with momentum, Adagrad, RMSProp, and Adam algorithms with a decaying learning rate, as mentioned before. For SGD, we employ the clipping method to clip the gradients to prevent excessively large steps.

For this experiment, we set $k = 5$ samples in both IWAE and BR-IWAE, while restricting the maximum iteration of the MCMC algorithm to 5 and the burn-in period to 2 for BR-IWAE. For comparison, we estimate the Negative Log-Likelihood using these three models with SGD, SGD with momentum, Adagrad, RMSProp, and Adam, and the results are presented in Table 2. Similar to the case of CIFAR-10, we observe that IWAE outperforms VAE, while

BR-IWAE outperforms IWAE by reducing bias in all cases. The adaptive methods surpass SGD, and momentum further improves their performances. Consequently, Adam excels among all algorithms due to its adaptive steps and momentum.

Table 2: Comparison of Negative Log-Likelihood on the FashionMNIST Test Set (Lower is Better).

ALGORITHM	VAE	IWAE	BR-IWAE
SGD	247.2	244.9	244.0
SGD WITH MOMENTUM	244.6	240.2	238.4
ADAGRAD	245.8	241.4	240.5
RMSPROP	242.6	239.3	237.8
ADAM	240.3	237.8	236.1

Similarly, as we did in the case of CIFAR-10, we incorporate a time-dependent bias that decreases by choosing a bias of order $O(n^{-\alpha})$ at iteration n. We vary the value of α for both FashionMNIST and CIFAR-100.

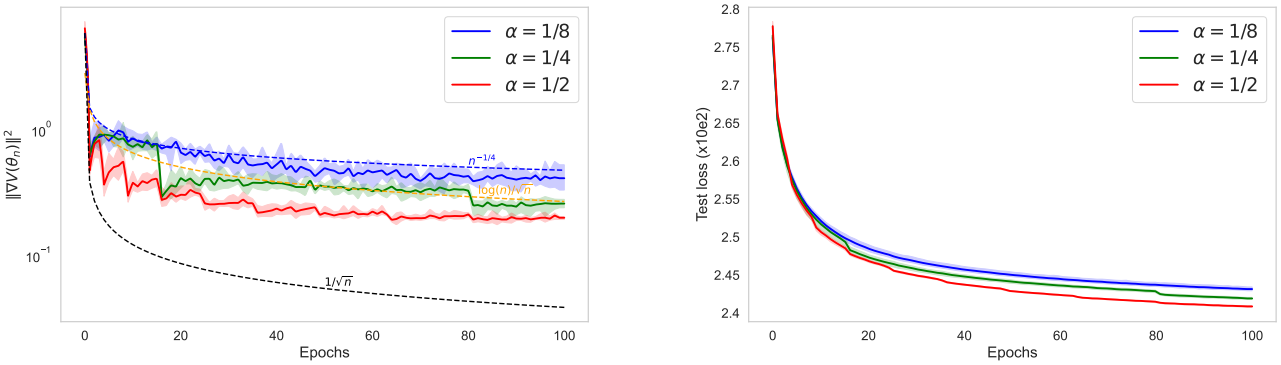

Figure 7: IWAE on the FashionMNIST Dataset with Adagrad for different values of α. Bold lines represent the mean over 5 independent runs.

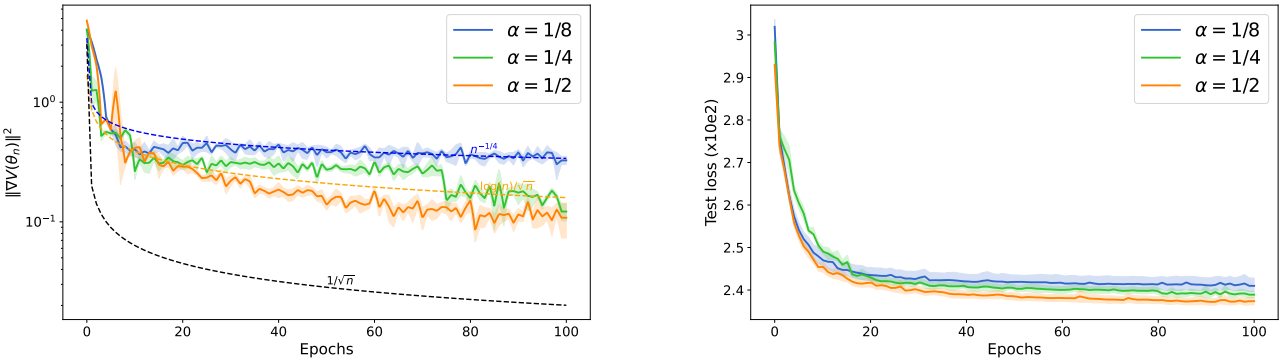

Figure 8: IWAE on the FashionMNIST Dataset with RMSProp for different values of α. Bold lines represent the mean over 5 independent runs.

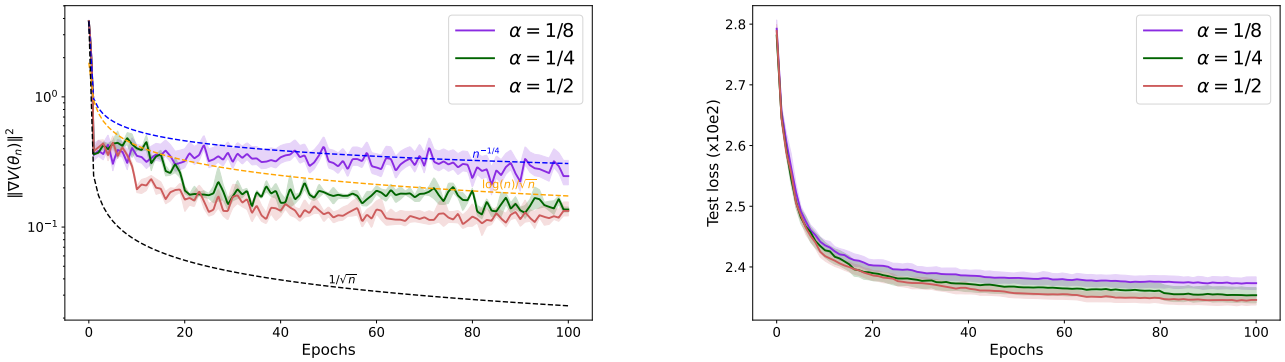

Figure 9: IWAE on the FashionMNIST Dataset with Adam for different values of α. Bold lines represent the mean over 5 independent runs.

All figures are plotted on a logarithmic scale for better visualization and with respect to the number of epochs. The dashed curve corresponds to the expected convergence rate $O(n^{-1/4})$ for $\alpha = 1/8$, and $O(\log n/\sqrt{n})$ for $\alpha = 1/4$, as well as for $\alpha = 1/2$, just as in the case of CIFAR-10. We can clearly observe that for all cases, convergence is achieved when n is sufficiently large. In the case of the FashionMNIST dataset, the bound seems tight, and the convergence rate of $O(n^{-1/2})$ does not seem to be possible to reach, in contrast to the case of CIFAR-10 where the curves corresponding to $\alpha = 1/4$ and $\alpha = 1/2$ approach the $O(n^{-1/2})$ convergence rate. For all figures, with a larger α, the convergence in both the squared gradient norm and negative log-likelihood occurs more rapidly.

The effect of C_γ.

Figure 10 illustrates the convergence in both the squared gradient norm and the negative log-likelihood for $C_\gamma = 0.001$ and $C_\gamma = 0.01$ in Adagrad. In the case of the squared gradient norm, we have only plotted the results for $C_\gamma = 0.001$ for better visualization, and the plot for $C_\gamma = 0.01$ was already presented in Figure 3. It is clear that when C_γ is set to 0.001, the convergence of the negative log-likelihood is slower. Similarly, the convergence in the squared gradient norm for $C_\gamma = 0.001$ achieves convergence, but it is slower compared to the case of $C_\gamma = 0.01$.

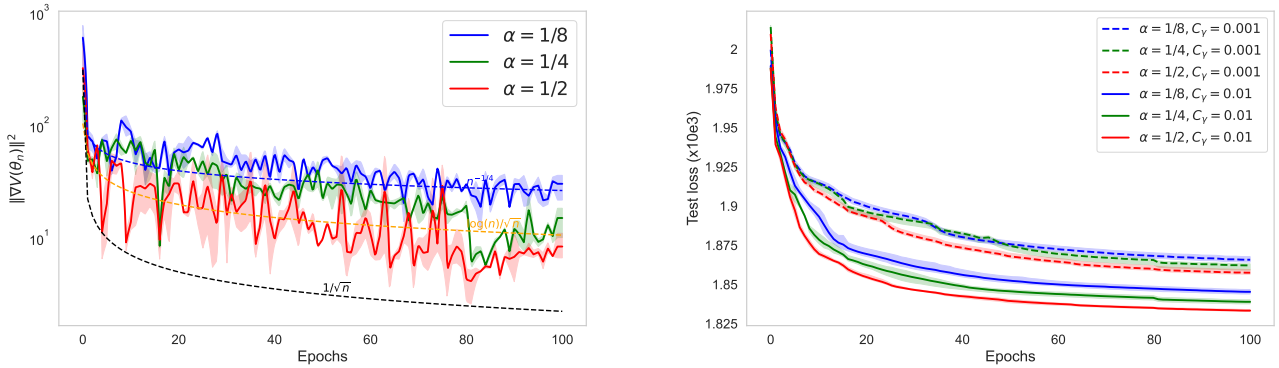

Figure 10: IWAE on the CIFAR-10 Dataset with Adagrad for different values of α and C_γ. Bold lines represent the mean over 5 independent runs.

The Impact of regularization parameter δ.

In Section A.7, we discussed the impact of the regularization parameter δ in Adam. It has been empirically observed that the performance of adaptive methods can be sensitive to the choice of this parameter. Here, we illustrate the impact of this regularization parameter in IWAE. To achieve this, we plot the test loss for different sets of values for $\delta \in \{10^{-8}, 10^{-5}, 10^{-3}, 10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$ in Figure 11.

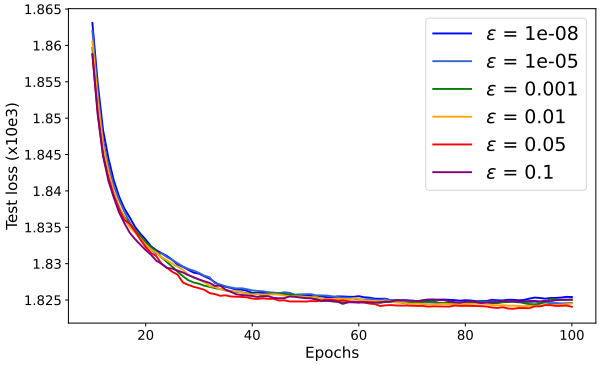

Figure 11: IWAE on the CIFAR-10 Dataset with Adam for different values of δ. Lines represent the mean over 5 independent runs.

As shown in (Zaheer et al., 2018; Reddi et al., 2018; Tong et al., 2022), we observe the same impact in IWAE and get better results with $\delta = 5 \times 10^{-2}$.

The Impact of Bias over Time.

Our experiments illustrate the negative log-likelihood with respect to epochs, and we observed that a higher value of α leads to faster convergence. The key point to consider when tuning α is that while convergence may be faster in terms of iterations, it may lead to higher computational costs. To illustrate this, we set a fixed time limit of 1000 seconds and tested different values of α, plotting the test loss as a function of time in Figure 12. It is clear that with $\alpha = 1/8$, the convergence is always slower, whereas choosing $\alpha = 1/4$ achieves faster convergence than $\alpha = 1/2$. While the difference may seem small here, with more complex models, the disparity becomes significant. Therefore, it is essential to tune the value of α to attain fast convergence and reduce computational time.

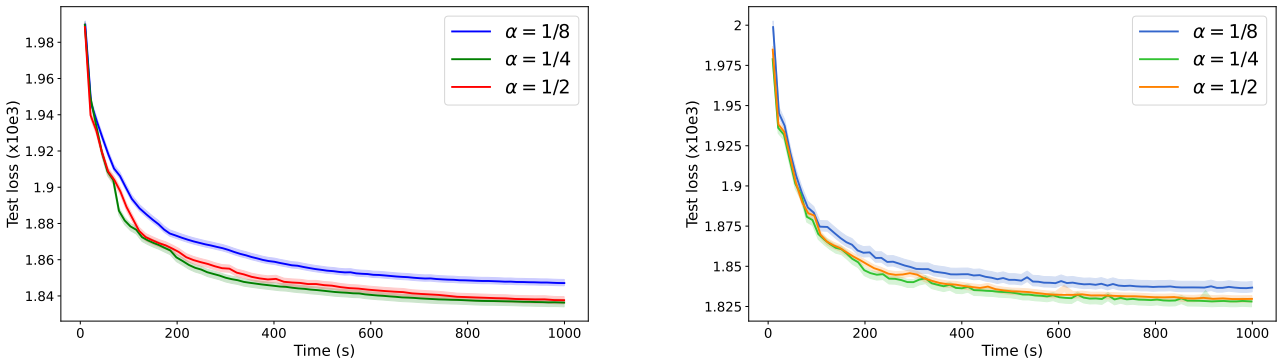

Figure 12: Negative Log-Likelihood on the test set of the CIFAR-10 Dataset for IWAE with Adagrad (on the left) RMSProp (on the right) for Different Values of α over time (in seconds). Bold lines represent the mean over 5 independent runs.

In this paper, all simulations were conducted using the Nvidia Tesla T4 GPU. The total computing hours required for the results presented in this paper are estimated to be around 100 to 200 hours of GPU usage.