Non-asymptotic Analysis of Biased Adaptive Stochastic Approximation

Sobihan Surendran[1,2] [*] **Adeline Fermanian**[2], **Antoine Godichon-Baggioni**[1], **Sylvain Le Corff**[1]
[1]Sorbonne Université, CNRS,
Laboratoire de Probabilités, Statistique et Modélisation, Paris, France
[2]LOPF, Califrais' Machine Learning Lab, Paris, France

Abstract

Stochastic Gradient Descent (SGD) with adaptive steps is widely used to train deep neural networks and generative models. Most theoretical results assume that it is possible to obtain 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 AMSGRAD, an exponential moving average variant of 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) and applications to several learning frameworks that illustrate our convergence results and show how the effect of bias can be reduced by appropriate hyperparameter tuning.

1 Introduction

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

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

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 $n \in \mathbb{N}$,

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

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 $n \in \mathbb{N}$ by:

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

where $\widehat{\nabla V}(\theta_n)$ is an estimator of $\nabla V(\theta_n)$. For example, in deep learning, stochasticity emerges with the use of mini-batches. While these algorithms have been extensively studied, both theoretically and practically, see, e.g., [10], 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

[*]Corresponding author: `sobihan.surendran@sorbonne-universite.fr`

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-0411

to optimize black-box functions [61] in generative adversarial networks [58, 16] have access only to noisy biased realizations of the objective functions.

Furthermore, in reinforcement learning algorithms such as Q-learning [42], policy gradient [5], and temporal difference learning [8, 52, 18], gradient estimators are often obtained using a Markov chain with state-dependent transition probability. These estimators are then biased [69, 23]. 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) [34, 13]. In particular, the Importance Weighted Autoencoder (IWAE) proposed by [12], which is an extension of the standard Variational Autoencoder (VAE) [48], yields biased estimators. Finally, this is also the case in Bilevel Optimization [43, 36, 41] and Conditional Stochastic Optimization [40, 39].

Moreover, in practical applications, vanilla SGD shows difficulties in calibrating 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 [4] and later adopted in stochastic optimization, with the Adagrad algorithm of [27].

In this paper, 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 [27, 77, 67, 74, 19], or biased estimators with non-adaptive steps [70, 44, 2, 22, 21].

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. To the best of our knowledge, these are the first convergence results to incorporate adaptive steps in biased Stochastic Approximation.
- In particular, we establish that Adagrad, RMSProp, and AMSGRAD, an exponential moving average variant of Adam, with a biased gradient, converge to a critical point for non-convex smooth functions with a convergence rate of $\mathcal{O}(\log n/\sqrt{n} + b_n)$, 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.
- Finally, we show how our theoretical results apply to several applications with biased gradients. In particular, we show that our hypotheses hold for Stochastic Bilevel Optimization and Conditional Stochastic Optimization, but also for Self-Normalized Importance Sampling estimators or Coordinate Sampling. We also propose a first non-asymptotic bound on the bias of IWAE, which allows us to illustrate through several experiments the effect of bias on the convergence of the optimization, and to show how this effect can be reduced by an appropriate choice of hyperparameters.

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. In Section 4, we present our principal results, i.e., convergence rates for the risk when the PL condition is assumed, and on the gradient norm without this hypothesis. We illustrate our results 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, [68] 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

https://doi.org/10.52202/079017-0411

a result, the minimization problem (1) is then equivalent to solving problem (2), and we can note that SGD is a specific instance of SA. SA 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 $H_{\theta_n}\left(X_{n+1}\right)$ 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 the theoretical risk, θ the model parameters, and X_{n+1} the data.

Adaptive Stochastic Gradient Descent. SGD can be traced back to [68], and its averaged counterpart was proposed by [66]. The non-asymptotic analysis of SGD in both convex and strong convex cases can be found in [59]. [32] prove the convergence of a random iterate of SGD for nonconvex smooth functions, which was already suggested by the results of [9]. They show that SGD with constant or decreasing stepsize $\gamma_n = 1/\sqrt{n}$ converges to a stationary point of a non-convex smooth function V at a rate of $\mathcal{O}(1/\sqrt{n})$ where n is the number of iterations.

Most adaptive first-order methods, such as Adam [47], Adadelta [78], RMSProp [72], and NADA [25], 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 [50]. [74] proved that Adagrad converges to a critical point for non-convex objectives at a rate of $\mathcal{O}(\log n/\sqrt{n})$ when using a scalar adaptive step. In addition, [79] extended this proof to multidimensional settings. More recently, [19] focused on the convergence rates for Adagrad and Adam. Furthermore, several modified versions of Adam have been proposed, such as AMSGRAD [77] and YOGI [67].

Biased Stochastic Approximation. The asymptotic results of Biased SA have been studied by [70]. The non-asymptotic analysis can be found in the reinforcement learning literature, especially in the context of temporal difference (TD) learning, as explored by [8, 52, 18]. The case of non-convex smooth functions has been studied by [44]. The authors establish convergence results for the mean field function at a rate of $\mathcal{O}(\log n/\sqrt{n} + b)$, 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 [2], specifically addressing the case of Martingale noise with a constant step size.

[46, 21] introduce a novel assumption, known as "Expected Smoothness", which is the weakest assumption compared to the existing literature on biased SGD that we extend to cover the adaptive case. 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 in [56, 22]. Applications of biased gradients can be found in Bilevel Optimization [43, 36, 41] and Conditional Stochastic Optimization [40, 39]. Moreover, biased gradients are also used in various other applications [38, 54, 6, 56]. Finally, [3] 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 SA algorithm with adaptive steps: $\theta_0 \in \mathbb{R}^d$ and for all $n \in \mathbb{N}$,

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

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 [27]. 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 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:

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

where

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

In RMSProp [72], $\bar{H}_n(X_{1:n+1}, \theta_{0:n})$ in (4) is an exponential moving average of the past squared gradients, defined by:

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

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 [11].

3.2 Assumptions

Consider the following assumptions.

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

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

H1 corresponds to the Polyak-Łojasiewicz condition, which is weaker than strong convexity and remains satisfied even when the function is non-convex. It ensures uniqueness of the minimizer θ^*. The PL condition has been extensively studied theoretically [45] and has been verified empirically in many applications, such as over-parameterized deep networks [26] and Linear Quadratic Regulator models [29].

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

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

This assumption is crucial to obtain our convergence rate and is very common see, e.g., [59, 10]. Under this assumption, for all $(\theta, \theta') \in \mathbb{R}^d \times \mathbb{R}^d$,

$$V(\theta) \leq V(\theta') + \langle \nabla V(\theta'), \theta - \theta' \rangle + \frac{L}{2} \|\theta - \theta'\|^2 . \tag{5}$$

H3 (i) Biased Gradients: There exist two non-increasing positive sequences $(\lambda_n)_{n\geq 1}$ and $(r_n)_{n\geq 1}$ such that for all $n \in \mathbb{N}$,

$$\mathbb{E}\big[\langle \nabla V(\theta_n), A_n H_{\theta_n}(X_{n+1}) \rangle\big] \geq \lambda_{n+1} \left(\mathbb{E}\left[\|\nabla V(\theta_n)\|^2\right] - r_{n+1}\right) .$$

(ii) Expected Smoothness: there exists a non-increasing non-negative sequence $(\sigma_n^2)_{n\geq 1}$, and positive constants $\tilde{\sigma}_1$, $\tilde{\sigma}_2$ such that for all $n \in \mathbb{N}$,

$$\mathbb{E}\big[\|H_{\theta_n}(X_{n+1})\|^2\big] \leq \sigma_n^2 + \tilde{\sigma}_1 \mathbb{E}\big[\|\nabla V(\theta_n)\|^2\big] + \tilde{\sigma}_2 \mathbb{E}\big[V(\theta_n) - V(\theta^*)\big] .$$

In this assumption, r_{n+1} represents an additive bias term, generally of the order of the square of the bias, and λ_{n+1} may depend on the minimum eigenvalue of A_n. In [21, Theorem 2], it has been demonstrated that this assumption is weaker than the alternatives used in the literature on biased SGD. We have extended these assumptions to the adaptive case. It is important to note that the first point of H3 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

https://doi.org/10.52202/079017-0411

H3 by considering $\lambda_{n+1} = 1/2$ and $r_{n+1} = \tilde{b}_{n+1}^2$. We show in Section 4.3 that this assumption is also verified in algorithms such as Adagrad and RMSProp. 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 finally consider 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 $(\beta_n)_{n\geq 1}$, one way to ensure that H4 is satisfied is to replace the random matrices A_n with

$$\tilde{A}_n = \frac{\min\{\|A_n\|, \beta_{n+1}\}}{\|A_n\|} A_n \ . \tag{6}$$

It is then clear that $\|\tilde{A}_n\| \leq \beta_{n+1}$. Furthermore, in most cases, especially for Adagrad, RMSProp and Stochastic Newton, control of $\lambda_{\max}(A_n)$ in H4 is satisfied. For example, in Adagrad and RMSProp, in (4), we have $\lambda_{\max}(A_n) \leq \delta^{-1/2}$.

4 Convergence Results

4.1 Convergence under the PL condition

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

Theorem 4.1. *Assume that H1 - H4 hold. 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,*

$$\mathbb{E}\left[V(\theta_n) - V(\theta^*)\right] = \mathcal{O}\left(n^{-\gamma+2\beta+\lambda} + r_n\right) . \tag{7}$$

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 (7), we would like to have $\beta = \lambda = 0$. For example, it is verified in the case of Adagrad and RMSProp if the gradients are bounded, as will be discussed later.

We stress that Theorem 4.1 applies to any adaptive algorithm of the form (3), with the only assumption being H4. 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, which can even be seen as a worst-case scenario. 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 Convergence without the PL condition

In the non-convex smooth case, theoretical results are generally based on a randomized version of SA, as described in [60, 32, 44]. Instead of considering the final parameter θ_n, we introduce a random variable R, which takes its values in $\{1, \ldots, n\}$, and the quantity of interest becomes θ_R. Note that this procedure is a technical tool, in practical applications we use classical SA. The following theorem provides a bound in expectation on the gradient of the objective function V, which is the best we can have given that no assumption is made about the existence of a global minimum of V.

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

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

where

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

If $\tilde{\sigma}_2 = 0$, Theorem 4.2 recovers the asymptotic convergence rate obtained by [44] with respect to the hyperparameters γ, β, and λ, and to the bias. We can observe that if $\gamma \le \lambda + 2\beta$, the condition on $(\gamma_k)_{k\ge1}$ 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 $\gamma_{k+1} \le 1/(\tilde{\sigma}_1 L)$.

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 Theorem 4.1.

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 $\gamma, \beta, \lambda \ge 0$ and $\gamma + \lambda < 1$. Then, if $\tilde{\sigma}_2 = 0$, we have:*

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

where the bias term b_n can be constant or decreasing. In the latter case, writing $r_n = C_r n^{-r}$, we have:

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

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 $\mathcal{O}(1/\sqrt{n} + b_n)$, and the best rate we can reach is $\mathcal{O}(\log n/\sqrt{n} + b_n)$ when $\gamma = 1/2, \beta = 0$, and $\lambda = 0$. In this case, all eigenvalues of A_n must be bounded from both below and above. Note that we could also have obtained such a rate by taking $\lambda_n = n^{-1/2}$ and $\beta_n = n^{-1/2}$ while keeping γ_n constant. However, the assumption that $\beta_n = n^{-1/2}$ is too strong (fast decay of the eigenvalues of A_n), hence our choice of $\beta_n = C_\beta n^\beta$. Finally, for a decreasing bias, if $r \ge 1/2$, the bias term contributes to the convergence rate of the algorithm. Otherwise, the other term is the leading term of the upper bound. In both cases, the best achievable bound is $\mathcal{O}(\log n/\sqrt{n})$ if $r \ge 1/2$.

Bounded Gradient Case. Now, we analyze the convergence of Randomized Adaptive Stochastic Approximation when the stochastic updates are bounded, as given by the following assumption.

H5 There exists $M \ge 0$ such that for all $n \in \mathbb{N}$, $\|H_{\theta_n}(X_{n+1})\| \le M$.

Boundedness of the stochastic gradient of the objective function is a classical assumption in adaptive stochastic optimization [67, 74, 19, 73].

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 $\gamma, \beta, \lambda \ge 0$ and $\gamma + \lambda < 1$. For any $n \ge 1$, let $R \in \{0, \dots, n\}$ be a uniformly distributed random variable. Then,*

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

where $\alpha'_{1,n} = \sum_{k=0}^{n} \gamma_{k+1}\lambda_{k+1}r_{k+1}$, $\alpha'_{2,n} = \sum_{k=0}^{n} \gamma_{k+1}^2\beta_{k+1}^2$, and $V^* = \mathbb{E}[V(\theta_0) - V(\theta^*)]$.

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

We give a convergence analysis of Adagrad and RMSProp with a biased gradient estimator. First, note that, under H5, for all eigenvalues λ of A_n, the adaptive matrix in Adagrad or RMSProp, it holds that $(M^2 + \delta)^{-1/2} \le \lambda \le \delta^{-1/2}$, i.e., H4 is satisfied with $\lambda = 0$ and $\beta = 0$.

Corollary 4.5. *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 $R \in \{0, \ldots, n\}$ be a uniformly distributed random variable. Suppose that for any $n \geq 1$, there exist positive constants α and C_α such that:*

$$\left\| \mathbb{E}\left[H_{\theta_n}\left(X_{n+1}\right) | \mathcal{F}_n\right] - \nabla V\left(\theta_n\right) \right\| \leq C_\alpha n^{-\alpha} \ . \tag{8}$$

Then,

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

where the bias b_n is explicitly given in Appendix A.5.

In the case of an unbiased gradient, we obtain the same bound of $\mathcal{O}(\log n/\sqrt{n})$ as in [74, 79, 19] under the same assumptions. If the bias is of the order $\mathcal{O}(n^{-1/4})$, the algorithm achieves the same convergence rate as in the case of an unbiased gradient.

4.4 AMSGRAD with Biased Gradients

Finally, we show the convergence of AMSGRAD [67] with a biased gradient estimator. At each iteration, AMSGRAD uses an exponential moving average of past gradients instead of the current gradient as in Equation (3), which is detailed in Algorithm 1. The key difference between Adam and AMSGRAD lies in their handling of the second moment estimate. Specifically, AMSGRAD uses the updated term $\hat{V}_k = \max(\hat{V}_{k-1}, \mathrm{Diag}(V_k))$ instead of directly using V_k, with the maximum taken coordinate-wise. This approach is crucial, as it ensures that the eigenvalues of A_n decrease at each iteration. The following theorem provides a bound in expectation on the gradient of the objective function V using randomized iterations with AMSGRAD.

Algorithm 1 AMSGRAD with Biased Gradients

```
Input: Initial point $\theta_0$, 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 $m_0 = 0, V_0 = 0$ and $\hat{V}_0 = 0$
for $k = 0$ to $n - 1$ do
    Compute the stochastic update $H_{\theta_k}\left(X_{k+1}\right)$
    $m_k = \rho_1 m_{k-1} + (1 - \rho_1) H_{\theta_k}(X_{k+1})$
    $V_k = \rho_2 V_{k-1} + (1 - \rho_2) H_{\theta_k}(X_{k+1}) H_{\theta_k}(X_{k+1})^\top$
    $\hat{V}_k = \max\left(\hat{V}_{k-1}, \mathrm{Diag}(V_k)\right)$
    $A_k = \left[\delta I_d + \hat{V}_k\right]^{-1/2}$
    $\theta_{k+1} = \theta_k - \gamma_{k+1} A_k m_k$
end for
Output: $(\theta_k)_{0\leq k\leq n}$
```

Theorem 4.6. *Assume that H2, H3 (i), and H5 hold. Let $\gamma_n = c_\gamma n^{-1/2}$, A_n denote the adaptive matrix of AMSGRAD in Algorithm 1, and $\rho_1, \rho_2 \in [0, 1)$. For any $n \geq 1$, let $R \in \{0, \ldots, n\}$ be a uniformly distributed random variable. Then,*

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

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

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

If the bias is of the order $\mathcal{O}(n^{-1/4})$, we achieve a convergence rate of $\mathcal{O}(\log n/\sqrt{n})$, which is the same as that of an unbiased gradient [19] and similar to that of Adagrad and RMSProp. It is worth noting that our results are also applicable to SGD momentum by taking $A_n = I_d$ in Algorithm 1.

4.5 Convergence Results in i.i.d. and Markov Chain cases

For illustrative purposes, in this subsection we give the form of the bias of the gradient estimator, denoted by $\tilde{b}_n$, in two simple scenarios, i.e., when $\{X_n, n \in \mathbb{N}\}$ is either an i.i.d. sequence or a Markov chain. For Adagrad, RMSProp, and AMSGRAD, bounding the bias of the gradient estimator is a sufficient condition for verifying H3(i), which in turn enables us to derive convergence results in each scenario.

I.i.d. case. Assume that $\{X_n, n \in \mathbb{N}\}$ are i.i.d. random variables. If the mean field function $h(\theta_n) = \mathbb{E}\left[H_{\theta_n}(X_{n+1}) \mid \mathcal{F}_n\right]$ aligns with the true gradient, then the estimator is unbiased. Otherwise, the bias of the gradient estimator is

$$\tilde{b}_{n+1} = \|h(\theta_n) - \nabla V(\theta_n)\| \ .$$

Markov Chain case. Assume now that $\{X_n, n \in \mathbb{N}\}$ is a Markov Chain. The bias consists of two parts: the difference between the mean field function and the true gradient, and a term due to the Markov chain dynamics. For all $T \geq 0$, we define the stochastic update as follows:

$$H_{\theta_k}(X_{k+1}) = \frac{1}{T}\sum_{i=1}^{T} H_{\theta_k}\left(X_{k+1}^{(i)}\right),$$

where $X_{k+1}^{(i)}$ represents the i-th sample generated at iteration $k+1$. This multi-sample estimator is commonly used in applications such as Reinforcement Learning, Markov Chain Monte Carlo, and Sequential Monte Carlo methods, effectively reducing the variance of the gradient estimator. The mixing time τ_{mix} of a Markov chain with stationary distribution π and transition kernel P is characterized as:

$$\tau_{\text{mix}} := \inf\left\{t \ ; \ \sup_x D_{\text{TV}}(P^t(x,\cdot),\pi) \leq \frac{1}{4}\right\},$$

where D_{TV} denotes the total variation distance. For an ergodic Markov chain with stationary distribution π, the bias of this gradient estimator when using T samples per step is

$$\tilde{b}_{n+1} = \|h(\theta_n) - \nabla V(\theta_n)\| + M\sqrt{\tau_{\text{mix}}/T} \ ,$$

where $h(\theta) = \int H_\theta(x)\pi(dx)$. If the general optimization problem reduces to the following stochastic optimization problem with Markov noise, as considered in most of the literature [28, 24, 7]:

$$\min_{\theta\in\mathbb{R}^d} V(\theta) := \mathbb{E}_{x\sim\pi}[f(\theta;x)],$$

where $\theta \mapsto f(\theta;x)$ is a loss function, and π is some stationary data distribution of the Markov Chain and $H_{\theta_k}(X_{k+1}^{(i)}) = \nabla f(\theta_k; X_{k+1}^{(i)})$, then $\tilde{b}_{n+1} = M\sqrt{\tau_{\text{mix}}/T}$, similar to SGD with Markov Noise [24].

5 Applications and Experiments

5.1 Bilevel and Conditional Stochastic Optimization

We can now apply our theoretical results in various settings where biased gradients are involved. In particular, they apply to the fields of Stochastic Bilevel Optimization and Conditional Stochastic Optimization. Stochastic Bilevel Optimization consists of minimizing an objective function V with respect to θ, where V is itself a function of $\phi^*(\theta)$ and $\phi^*(\theta)$ is obtained by solving another minimization problem. Conditional Stochastic Optimization focuses on optimizing the expected value of a function that contains a nested conditional expectation on a random variable η. We provide in Table 1 a summary of the assumptions satisfied in these settings, which allow to apply the results of Section 4 and to obtain a $\mathcal{O}(\log n/\sqrt{n} + b_n)$ convergence rate in both cases, and explicit forms for b_n. To our knowledge, these are the first convergence rates obtained in these settings.

We refer to Appendix D for other examples in which the bias of the estimator can be controlled, in particular 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).

5.2 Experiments with IWAE and BR-IWAE

In this section, we illustrate our theoretical results in the context of deep VAE. The experiments were conducted using PyTorch [65], and the source code can be found here[2]. In generative models,

[2] https://github.com/SobihanSurendran/Adaptive-SA

https://doi.org/10.52202/079017-0411

Table 1: Bilevel and Conditional Stochastic Optimization with our Biased Adaptive SA framework.

Applications	Stochastic Bilevel Optimization	Conditional Stochastic Optimization
Problem	$\min_{\theta \in \mathbb{R}^d} \mathbb{E}_\xi \left[f(\theta, \phi^*(\theta); \xi) \right]$ s.t. $\phi^*(\theta) \in \operatorname{argmin}_{\phi \in \mathbb{R}^q} \mathbb{E}_\zeta \left[g(\theta, \phi; \zeta) \right]$	$\min_{\theta \in \mathbb{R}^d} \mathbb{E}_\xi \left[f_\xi \left(\mathbb{E}_{\eta \mid \xi} \left[g_\eta(\theta, \xi) \right] \right) \right]$
Lipchitz Constant H2	Lemma C.2	Lemma C.6
Bias Control H3	Lemma C.3	Lemma C.5
Gradient Bound H5	Lemma C.3	Lemma C.6
Convergence	Theorem C.4	Theorem C.7

the objective is to maximize the marginal likelihood $\log p_\theta(x)$, which is the marginalization of $(x, z) \mapsto p_\theta(x, z)$, where x represents the observations and z is the latent variable. Under some simple technical assumptions, by Fisher's identity, we have:

$$\nabla_\theta \log p_\theta(x) = \int \nabla_\theta \log p_\theta(x, z) p_\theta(z \mid x) \mathrm{d}z \ . \tag{9}$$

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):

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

The Importance Weighted Autoencoder (IWAE) [12] is a variant of the VAE that incorporates importance weighting to obtain a tighter ELBO. The IWAE objective can be written as follows:

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

where k corresponds to the number of samples drawn from the variational posterior distribution. The estimator of the gradient of ELBO in IWAE is a biased estimator of $\nabla_\theta \log p_\theta(x)$. In Theorem B.1, we establish that the bias of this estimator is of order $\mathcal{O}(1/k)$, thereby allowing us to derive a convergence rate for IWAE. 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) [14], which is detailed in Appendix B.

Dataset and Model. We conduct our experiments on the CIFAR-10 dataset [51] and 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. We estimate the log-likelihood using 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. Although AMSGRAD is analyzed in our theoretical results, we use Adam for the experiments due to its widespread use in practice. Additional details are provided in Appendix E.

First, we set $k = 5$ samples in both IWAE and BR-IWAE. The test losses are presented in Figure 1. We show the negative log-likelihood on the test dataset for VAE, IWAE, and BR-IWAE with Adagrad, RMSProp, and Adam. As expected, we observe that IWAE outperforms VAE, while BR-IWAE outperforms IWAE by reducing bias in all cases.

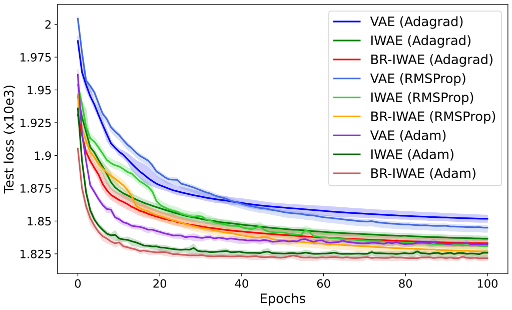

Figure 1: Negative Log-Likelihood on the test set for Different Generative Models with Adagrad, RMSProp, and Adam on CIFAR-10. Bold lines represent the mean over 5 independent runs.

Then, we illustrate empirically the convergence rates obtained in Corollary 4.5 and Theorem 4.6 for IWAE. Since the bias of the estimator of the gradient in IWAE is of the order $\mathcal{O}(1/k)$, choosing a bias of order $\mathcal{O}(n^{-\alpha})$ is equivalent to using n^α samples at iteration n to estimate the gradient. We plot in Figure 2 the gradient squared norm $\|\nabla V(\theta_n)\|^2$ and the Negative Log-Likelihood is given in Appendix E.2. Note that all figures are with respect to epochs, whereas here, n represents the number of updates of the gradient. The dashed curves correspond to the expected convergence rate $\mathcal{O}(n^{-1/4})$ for $\alpha = 1/8$ and $\mathcal{O}(\log n/\sqrt{n})$ for $\alpha = 1/4$ and $\alpha = 1/2$.

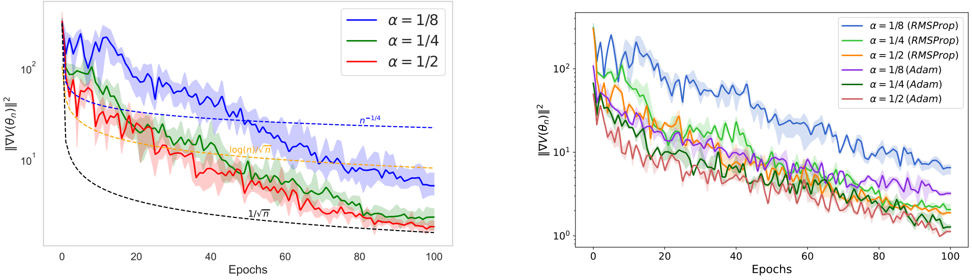

Figure 2: Value of $\|\nabla V(\theta_n)\|^2$ in IWAE with Adagrad (on the left), RMSProp, and Adam (on the right). Bold lines represent the mean over 5 independent runs. Figures are plotted on a logarithmic scale for better visualization. Both figures have the same scale, so we have not shown the dashed theoretical curves on the right for better clarity.

We observe that the algorithms converge at the expected theoretical rates, and even faster. In Appendix E.2, we have included an additional experiment on the FashionMNIST dataset [76], which shows similar behavior, but the convergence is closer to the expected rates, suggesting that our upper bounds may be tight. We see similar convergence rates for Adagrad, RMSProp, and Adam, although, as expected, Adam performs slightly better. Moreover, it is clear that convergence is faster with a larger α but beyond a certain threshold for α the rate of convergence does not change significantly. Since choosing a larger α induces an additional computational cost, it is crucial to choose an appropriate value that achieves fast convergence without being too computationally expensive. Choosing an optimal number of samples at each iteration remains an open problem depending on the chosen generative model.

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 $\mathcal{O}(\log n/\sqrt{n} + b_n)$ 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 AMSGRAD 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. A natural extension of this work is the analysis of the assumptions, the bias and convergence rates for specific deep learning architectures. A theoretical analysis of the Monte Carlo effort required at each iteration to obtain an optimal convergence rate is another interesting perspective.

Acknowledgements

The Ph.D. of Sobihan Surendran was funded by the Paris Region PhD Fellowship Program of Région Ile-de-France. We would like to thank SCAI (Sorbonne Center for Artificial Intelligence) for providing the computing clusters. We also express our gratitude to the reviewers for their insightful comments and suggestions, which have helped improve this paper.

References

[1] Sergios Agapiou, Omiros Papaspiliopoulos, Daniel Sanz-Alonso, and Andrew M Stuart. Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, pages 405–431, 2017.

[2] Ahmad Ajalloeian and Sebastian U Stich. On the Convergence of SGD with Biased Gradients. *arXiv preprint arXiv:2008.00051*, 2020.

[3] Ahmet Alacaoglu and Hanbaek Lyu. Convergence of first-order methods for constrained nonconvex optimization with dependent data. In *International Conference on Machine Learning*, pages 458–489. PMLR, 2023.

[4] Peter Auer, Nicolo Cesa-Bianchi, and Claudio Gentile. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64(1):48–75, 2002.

[5] Jonathan Baxter and Peter L Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.

[6] Aleksandr Beznosikov, Samuel Horváth, Peter Richtárik, and Mher Safaryan. On biased compression for distributed learning. *Journal of Machine Learning Research*, 24(276):1–50, 2023.

[7] Aleksandr Beznosikov, Sergey Samsonov, Marina Sheshukova, Alexander Gasnikov, Alexey Naumov, and Eric Moulines. First order methods with markovian noise: from acceleration to variational inequalities. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

[8] Jalaj Bhandari, Daniel Russo, and Raghav Singal. A finite time analysis of temporal difference learning with linear function approximation. In *Conference On Learning Theory*, pages 1691–1692. PMLR, 2018.

[9] Léon Bottou. *Une approche théorique de l'apprentissage connexioniste; applications à la reconnaissance de la parole*. PhD thesis, Paris 11, 1991.

[10] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

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

[12] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2016.

[13] Gabriel Cardoso, Yazid Janati El Idrissi, Sylvain Le Corff, Eric Moulines, and Jimmy Olsson. State and parameter learning with PaRIS particle Gibbs. In *International Conference on Machine Learning*, pages 3625–3675. PMLR, 2023.

[14] Gabriel Cardoso, Sergey Samsonov, Achille Thin, Eric Moulines, and Jimmy Olsson. BR-SNIS: bias reduced self-normalized importance sampling. In *Advances in Neural Information Processing Systems*, volume 35, pages 716–729, 2022.

[15] Congliang Chen, Li Shen, Fangyu Zou, and Wei Liu. Towards practical adam: Non-convexity, convergence theory, and mini-batch acceleration. *Journal of Machine Learning Research*, 23(229):1–47, 2022.

[16] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 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, 2017.

[17] Tianyi Chen, Yuejiao Sun, and Wotao Yin. Closing the gap: Tighter analysis of alternating stochastic gradient methods for bilevel problems. In *Advances in Neural Information Processing Systems*, volume 34, pages 25294–25307, 2021.

https://doi.org/10.52202/079017-0411

[18] Gal Dalal, Balázs Szörényi, Gugan Thoppe, and Shie Mannor. Finite sample analyses for td (0) with function approximation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6144–6160, 2018.

[19] Alexandre Défossez, Léon Bottou, Francis Bach, and Nicolas Usunier. A simple convergence proof of Adam and Adagrad. *arXiv preprint arXiv:2003.02395*, 2020.

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

[21] Yury Demidovich, Grigory Malinovsky, Igor Sokolov, and Peter Richtárik. A guide through the zoo of biased sgd. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

[22] Aymeric Dieuleveut, Gersende Fort, Eric Moulines, and Hoi-To Wai. Stochastic approximation beyond gradient for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 2023.

[23] Thinh T Doan, Lam M Nguyen, Nhan H Pham, and Justin Romberg. Finite-time analysis of stochastic gradient descent under markov randomness. *arXiv preprint arXiv:2003.10973*, 2020.

[24] Ron Dorfman and Kfir Yehuda Levy. Adapting to mixing time in stochastic optimization with markovian data. In *International Conference on Machine Learning*, pages 5429–5446. PMLR, 2022.

[25] Timothy Dozat. Incorporating Nesterov momentum into Adam. In *ICLR Workshop track*, 2016.

[26] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685. PMLR, 2019.

[27] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7), 2011.

[28] John C Duchi, Alekh Agarwal, Mikael Johansson, and Michael I Jordan. Ergodic mirror descent. *SIAM Journal on Optimization*, 22(4):1549–1578, 2012.

[29] Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In *International Conference on Machine Learning*, pages 1467–1476. PMLR, 2018.

[30] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[31] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pages 1568–1577. PMLR, 2018.

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

[33] Saeed Ghadimi and Mengdi Wang. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*, 2018.

[34] Pierre Gloaguen, Sylvain Le Corff, and Jimmy Olsson. A pseudo-marginal sequential Monte Carlo online smoothing algorithm. *Bernoulli*, 28(4):2606–2633, 2022.

[35] Antoine Godichon-Baggioni and Pierre Tarrago. Non asymptotic analysis of adaptive stochastic gradient algorithms and applications. *arXiv preprint arXiv:2303.01370*, 2023.

[36] Riccardo Grazzi, Massimiliano Pontil, and Saverio Salzo. Bilevel optimization with a lower-level contraction: Optimal sample complexity without warm-start. *Journal of Machine Learning Research*, 24(167):1–37, 2023.

https://doi.org/10.52202/079017-0411

[37] Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic. *SIAM Journal on Optimization*, 33(1):147–180, 2023.

[38] Bin Hu, Peter Seiler, and Laurent Lessard. Analysis of biased stochastic gradient descent using sequential semidefinite programs. *Mathematical programming*, 187:383–408, 2021.

[39] Yifan Hu, Xin Chen, and Niao He. On the bias-variance-cost tradeoff of stochastic optimization. In *Advances in Neural Information Processing Systems*, volume 34, pages 22119–22131, 2021.

[40] Yifan Hu, Siqi Zhang, Xin Chen, and Niao He. Biased stochastic first-order methods for conditional stochastic optimization and applications in meta learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 2759–2770, 2020.

[41] Feihu Huang, Junyi Li, and Shangqian Gao. Biadam: Fast adaptive bilevel optimization methods. *arXiv preprint arXiv:2106.11396*, 2021.

[42] Tommi Jaakkola, Michael Jordan, and Satinder Singh. Convergence of stochastic iterative dynamic programming algorithms. In *Advances in Neural Information Processing Systems*, volume 6, 1993.

[43] Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *International Conference on Machine Learning*, pages 4882–4892. PMLR, 2021.

[44] Belhal Karimi, Blazej Miasojedow, Eric Moulines, and Hoi-To Wai. Non-asymptotic analysis of biased stochastic approximation scheme. In *Conference on Learning Theory*, pages 1944–1974. PMLR, 2019.

[45] Hamed Karimi, Julie Nutini, and Mark Schmidt. 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, 2016.

[46] Ahmed Khaled and Peter Richtárik. Better theory for SGD in the nonconvex world. *arXiv preprint arXiv:2002.03329*, 2020.

[47] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[48] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

[49] Vijaymohan R Konda and Vivek S Borkar. Actor-critic–type learning algorithms for markov decision processes. *SIAM Journal on control and Optimization*, 38(1):94–123, 1999.

[50] Milena Kresoja, Zorana Lužanin, and Irena Stojkovska. Adaptive stochastic approximation algorithm. *Numerical Algorithms*, 76(4):917–937, 2017.

[51] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical Report*, 2009.

[52] Chandrashekar Lakshminarayanan and Csaba Szepesvari. 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, 2018.

[53] Rémi Leluc and François Portier. Sgd with coordinate sampling: Theory and practice. *The Journal of Machine Learning Research*, 23(1):15470–15516, 2022.

[54] Qiang Li and Hoi-To Wai. State dependent performative prediction with stochastic approximation. In *International Conference on Artificial Intelligence and Statistics*, pages 3164–3186. PMLR, 2022.

[55] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *International Conference on Learning Representations*, 2019.

https://doi.org/10.52202/079017-0411

[56] Yin Liu and Sam Davanloo Tajbakhsh. Adaptive stochastic optimization algorithms for problems with biased oracles. *arXiv preprint arXiv:2306.07810*, 2023.

[57] Don McLeish. A general method for debiasing a Monte Carlo estimator. *Monte Carlo methods and applications*, 17(4):301–315, 2011.

[58] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1765–1773, 2017.

[59] Eric Moulines and Francis Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, volume 24, 2011.

[60] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

[61] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17:527–566, 2017.

[62] Sebastian Nowozin. Debiasing evidence approximations: On importance-weighted autoencoders and jackknife variational inference. In *International Conference on Learning Representations*, 2018.

[63] Julie Nutini, Mark Schmidt, Issam Laradji, Michael Friedlander, and Hoyt Koepke. Coordinate descent converges faster with the gauss-southwell rule than random selection. In *International Conference on Machine Learning*, pages 1632–1641. PMLR, 2015.

[64] Jimmy Olsson and Johan Westerborn. Efficient particle-based online smoothing in general hidden Markov models: the PaRIS algorithm. *Bernoulli*, 23(3):1951–1996, 2017.

[65] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

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

[67] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

[68] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

[69] Tao Sun, Yuejiao Sun, and Wotao Yin. On Markov chain gradient descent. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[70] Vladislav B Tadić and Arnaud Doucet. Asymptotic bias of stochastic gradient search. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 722–727. IEEE, 2011.

[71] Yee Teh, David Newman, and Max Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, volume 19, 2006.

[72] Tijmen Tieleman, Geoffrey Hinton, et al. 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, 2012.

[73] Qianqian Tong, Guannan Liang, and Jinbo Bi. Calibrating the adaptive learning rate to improve convergence of adam. *Neurocomputing*, 481:333–356, 2022.

[74] Rachel Ward, Xiaoxia Wu, and Leon Bottou. Adagrad stepsizes: Sharp convergence over nonconvex landscapes. *The Journal of Machine Learning Research*, 21(1):9047–9076, 2020.

https://doi.org/10.52202/079017-0411

[75] Xidong Wu, Jianhui Sun, Zhengmian Hu, Junyi Li, Aidong Zhang, and Heng Huang. Federated conditional stochastic optimization. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

[76] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[77] Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. In *Advances in neural information processing systems*, volume 31, 2018.

[78] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[79] Fangyu Zou, Li Shen, Zequn Jie, Ju Sun, and Wei Liu. Weighted adagrad with unified momentum. *arXiv preprint arXiv:1808.03408*, 2018.

https://doi.org/10.52202/079017-0411

Supplementary Material for “Non-asymptotic Analysis of Biased Adaptive Stochastic Approximation”

Table of Contents

https://doi.org/10.52202/079017-0411

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 [35, Proposition 6.1]

Theorem A.2. *Assume that H1 - H4 hold. Let $\theta_n \in \mathbb{R}^d$ be the n-th iterate of the recursion* (3)*. Then,*

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

where

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

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

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

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

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

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

Then, using H3,

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

https://doi.org/10.52202/079017-0411

Given the smoothness condition (5), with $\theta' = \theta_n$ and $\theta = \theta_n - \frac{1}{L}\nabla V(\theta_n)$, we derive:

$$V(\theta^*) \leq V(\theta_n) - \frac{1}{L}\left\|\nabla V(\theta_n)\right\|^2 + \frac{1}{2L}\left\|\nabla V(\theta_n)\right\|^2 \ ,$$

$$\left\|\nabla V(\theta_n)\right\|^2 \leq 2LV_n \ .$$

Using the above inequality and the Polyak-Łojasiewicz condition (H1), we obtain:

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

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

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

In order to satisfy the assumptions of Lemma A.1, consider

$$C_n = \max\left\{\tilde{\sigma}, \frac{\mu^2\lambda_{n+1}^2}{4\beta_{n+1}^2}\right\} \quad \text{with} \quad \tilde{\sigma} = \frac{\tilde{\sigma}_2 L}{2} + \tilde{\sigma}_1 L^2 \ .$$

Then, since $C_n \geq \tilde{\sigma}$, we have:

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

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:

$$\begin{aligned}\mathbb{E}\left[V(\theta_n) - V(\theta^*)\right] \leq \Big(\mathbb{E}\left[V(\theta_0) - V(\theta^*)\right] + \frac{2}{\tilde{\sigma}}\max_{1\leq k\leq n}\frac{v_k\lambda_{k+1}^2}{\beta_{k+1}^2\bar{\gamma}_{k+1}}\Big)\exp\Big(-\frac{\mu}{2}\sum_{k=n/2}^{n}\bar{\gamma}_{k+1}\Big) \\ \times \exp\Big(2\sum_{k=1}^{n}C_k\beta_{k+1}^2\bar{\gamma}_{k+1}^2/\lambda_{k+1}^2\Big) + \frac{2}{\mu}\max_{n/2\leq k\leq n}\left\{v_k\right\},\end{aligned}$$

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

A.2 Proof of Theorem 4.2

By H2, using (5), we obtain:

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

which, using the recursion (3) of Adaptive SA and H4, yields:

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

with $\delta_{k+1} = L\gamma_{k+1}^2\beta_{k+1}^2/2$. Using Assumption H3, we have:

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

Therefore,

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

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,

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

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

$$\begin{aligned}&\sum_{k=0}^n w_{k+1}\gamma_{k+1}\lambda_{k+1}\left(1-\frac{L\tilde{\sigma}_1}{2\lambda_{k+1}}\gamma_{k+1}\beta_{k+1}^2\right)\mathbb{E}\left[\|\nabla V(\theta_k)\|^2\right]\\ &\leq w_0\mathbb{E}\left[V_0\right]-w_{n+1}\mathbb{E}\left[V_{n+1}\right]+\frac{1}{2}\sum_{k=0}^n w_{k+1}\gamma_{k+1}\lambda_{k+1}r_{k+1}+\sum_{k=0}^n w_{k+1}\delta_{k+1}\sigma_k^2.\end{aligned}$$

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

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

Consequently, by definition of the discrete random variable R,

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

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

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 H2, using (5), we obtain:

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

which, using H4 and H5 yields:

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

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 4.5

Here, we consider the case where the regularization is non-increasing, i.e., where $\delta = \beta_{n+1}^{-2}$. The constant case is strictly analogous. To verify H4, we demonstrate that the control of the maximum and minimum eigenvalues is satisfied for Adagrad and RMSProp.

Adagrad

- **Lower bound for the smallest eigenvalue of $\mathbf{A_n}$.** 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:

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

- **Upper bound for the largest eigenvalue of $\mathbf{A_n}$.**

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

https://doi.org/10.52202/079017-0411

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 $\mathbf{A_n}$.** By assumption H5, we have:

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

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

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

- **Upper bound for the largest eigenvalue of $\mathbf{A_n}$.** Note that

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

Therefore, under H2, H3(i), and H5, we can conclude that

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

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

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

Now, we show that under the control of bias, i.e.,

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

we can verify H3(i) with a similar bound on the bias, where $r = 2\alpha$. This yields the bias term b_n as follows:

$$b_n = \begin{cases} \mathcal{O}\left(n^{-2\alpha}\right) & \text{if } \alpha < 1/4 \ , \\ \mathcal{O}\left(n^{-1/2}\right) & \text{if } \alpha > 1/4 \ , \\ \mathcal{O}\left(n^{-1/2}\log n\right) & \text{if } \alpha = 1/4 \ . \end{cases}$$

Verifying Assumption H3 (i) for Adagrad. Using the tower property, we have:

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

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,

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

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

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

First, writing

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

https://doi.org/10.52202/079017-0411

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

$$u_n = \beta_{n+1}^{-2} + \frac{1}{n+1}\sum_{k=0}^{n}\left(H_{\theta_k}\left(X_{k+1}\right)[i]\right)^2 \quad \text{and} \quad v_n = \beta_{n+1}^{-2} + \frac{1}{n+1}\sum_{k=0}^{n-1}\left(H_{\theta_k}\left(X_{k+1}\right)[i]\right)^2 .$$

Then, since $u_n \geq v_n$,

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

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

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

As $H_{\theta_n}\left(X_{n+1}\right)$ 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

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

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 4.6

The proof of this theorem is inspired by [67] and [73], considering biased gradient estimators and decreasing step sizes. We define the operation $\max(D_1, D_2)$ for diagonal matrices D_1 and D_2 as the matrix formed by taking the maximum between the diagonal elements of D_1 and D_2. We say that the sequence $(A_n)_{n\geq 1}$ of diagonal matrices is decreasing if all diagonal terms are decreasing, in other words, if all eigenvalues are decreasing.

Let $\tilde{\theta}_{k+1} = \theta_{k+1} + \kappa\left(\theta_{k+1} - \theta_k\right)$, 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 AMSGRAD, we have:

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

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 AMSGRAD together with a Taylor expansion with $\tilde{\theta}_k$, we obtain:

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

where

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

Note first that

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

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:

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

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:

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

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:

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

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 [15, Lemma 24] yields

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

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

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

which concludes the proof.

Lemma A.3. *Let $\gamma_{k+1} \leq \gamma_k$ for all $k \in \mathbb{N}$, and let A_k be the adaptive matrix defined in Algorithm 1. Assume that $\rho_1 \in [0, 1)$. Then, for all $k \in \mathbb{N}$:*

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

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 (in the sense that all eigenvalues of 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. □

https://doi.org/10.52202/079017-0411

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 [77, 67, 73], 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 [77] and YOGI [67] with the discussion of the regularization parameter δ. The authors of [73] 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 $\lambda_{\text{max}}(A_k) \leq b/\log 2$, 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.

https://doi.org/10.52202/079017-0411

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. The IWAE objective function is defined as:

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

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:

$$\begin{aligned} \nabla_\theta V(\theta) &= \nabla_\theta \log p_\theta(x) = \mathbb{E}_{p_\theta(\cdot|x)} \left[\nabla_\theta \log p_\theta(x, z) \right] , \\ \widehat{\nabla}_\theta \mathcal{L}_k^{\mathrm{IWAE}}(\theta, \phi; x) &= \sum_{\ell=1}^{k} \frac{w^{(\ell)}}{\sum_{\ell=1}^{k} w^{(\ell)}} \nabla_\theta \log p_\theta(x, z^{(\ell)}) , \end{aligned} \tag{10}$$

where $w^{(\ell)} = p_\theta(x, z^{(\ell)}) / q_\phi(z^{(\ell)}|x)$ the unnormalized importance weights. Theorem B.1 provides an upper bound for the bias of this estimator.

Theorem B.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 there exists M such that for all $\theta \in \Theta \subset \mathbb{R}^d$, $x \in \mathsf{X}$ and $z \in \mathsf{Z}$, $\|\nabla_\theta \log p_\theta(x, z)\| \leq M(x)$. Then, there exists a constant $C > 0$ such that for all $\theta \in \Theta$, $\phi \in \Phi$ and $x \in \mathsf{X}$,*

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

where $\nabla_\theta V(\theta)$ and $\widehat{\nabla}_\theta \mathcal{L}_k^{\mathit{IWAE}}(\theta, \phi; x)$ are defined in (10).

Proof. The proof is adapted from [1, 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 $\mathbb{E}_{q_\phi}[w \tilde{H}(x, z)] = 0$, we have:

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

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

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

so that

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

https://doi.org/10.52202/079017-0411

Therefore,

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

where

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

Note that

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

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

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

and

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

Finally, we deduce that

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

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

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

which concludes the proof. □

Algorithm 2 Adaptive Stochastic Approximation for IWAE

```
Input: Initial point $\theta_0$, 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 $k^\alpha$ samples from the variational posterior distribution and adaptive steps $A_k$.
    Set $\theta_{k+1} = \theta_k - \gamma_{k+1} A_k \nabla_\theta \mathcal{L}_{k^\alpha}^{\text{IWAE}}(\theta_k, \phi_k; X_{k+1})$.
    Set $\phi_{k+1} = \phi_k - \gamma_{k+1} A_k \nabla_\phi \mathcal{L}_{k^\alpha}^{IWAE}(\theta_k, \phi_k; X_{k+1})$.
end for
Output: $(\theta_k)_{0 \leq k \leq n}$
```

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 $\mathbb{E}_{p_\theta(\cdot|x)}\left[\nabla_\theta \log p_\theta(x, z)\right]$ using the BR-SNIS estimator [14]. This estimator aims to reduce the bias of self-normalized importance sampling estimators without increasing the variance.

Algorithm 3 BR-IWAE Gradient Estimator

```
Input: Maximum number of iterations $t_{\text{max}}$ 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 $t_{\text{max}} - 1$ 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:
        $$w_{t+1}^{(\ell)} = \frac{p_\theta(x, z_{t+1}^{(\ell)})}{q_\phi(z_{t+1}^{(\ell)} \mid x)} \quad \forall \ell \in \{1, \ldots, k\}.$$
    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 $z_{t+1}^{\ell}$ with probability $\omega_{t+1}^{(\ell)}$.
end for
Output: $\left(z_t^{1:k}\right)_{1 \leq t \leq t_{\text{max}}}$ and $\left(\omega_t^{1:k}\right)_{1 \leq t \leq t_{\text{max}}}$.
```

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

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

where t_0 corresponds to a burn-in period. By [14, 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 (9) using BR-SNIS as detailed in Algorithm 3:

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

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 [62] 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 $\mathcal{O}(k^{-2})$. This can also be generalized to have a bias of order $\mathcal{O}(k^{-m})$ 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) [71] 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

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

The Monte Carlo estimator of the Delta method Variational Inference objective achieves a reduced bias of $\mathcal{O}(k^{-2})$. Some other techniques for reducing bias include the iterated bootstrap for bias correction, the debiasing lemma [57], and Multi-Level Monte Carlo and its variants [39].

https://doi.org/10.52202/079017-0411

C Application of Our Theorem to Bilevel and Conditional Stochastic Optimization

C.1 Stochastic Bilevel Optimization

We consider the Stochastic Bilevel Optimization problem given by:

$$\min_{\theta \in \mathbb{R}^d} V(\theta) = \mathbb{E}_\xi \left[f(\theta, \phi^*(\theta); \xi) \right] \quad \text{(upper-level)} \tag{11}$$

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 (11) 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 [31], metalearning [30], policy optimization [37] and neural network architecture search [55]. 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, [43, 17] proposes a method to estimate v^*. This estimation introduces bias in the gradient of the objective function.

Consider the following assumptions.

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, ∇f, ∇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 [17] 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. *([17, Lemma 2.2]) Under Assumption 6, we have:*

$$\nabla V(\theta) = \nabla_\theta f\left(\theta, \phi^*(\theta)\right) - \nabla_{\theta\phi}^2 g\left(\theta, \phi^*(\theta)\right) \left[\nabla_\phi^2 g\left(\theta, \phi^*(\theta)\right)\right]^{-1} \nabla_\phi f\left(\theta, \phi^*(\theta)\right) \ . \tag{12}$$

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 ϕ and define

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

Furthermore, by estimating $\left[\nabla_\phi^2 g(\theta, \phi)\right]^{-1}$, we define the stochastic update H_k [17] as follows:

$$H_k = \nabla_\theta f\left(\theta_k, \phi_{k+1}; \xi_k\right) - \nabla_{\theta\phi}^2 g\left(\theta_k, \phi; \zeta_k^{(0)}\right) \widehat{G} \nabla_\phi f\left(\theta_k, \phi_{k+1}; \xi_k\right), \tag{13}$$

where $\widehat{G} = \frac{N}{\ell_{g,1}} \prod_{i=1}^{N'} \left(I - \frac{1}{\ell_{g,1}} \nabla_\phi^2 g\left(\theta_k, \phi_{k+1}; \zeta_k^{(i)}\right)\right)$ with N' is drawn from $\{1, \ldots, N\}$ uniformly at random and $\left\{\zeta^{(1)}, \ldots, \zeta^{(N')}\right\}$ are i.i.d. samples.

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 AMSGRAD.

Lemma C.2. *([33, Lemma 2.2]) Under Assumptions H6 and H7, for all $(\theta, \theta') \in \mathbb{R}^d \times \mathbb{R}^d$,*

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

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

Algorithm 4 Stochastic Bilevel Optimization

```
Input: Initial points $\theta_0, \phi_0$, maximum number of iterations for the upper-level $n$ and for the lower-level $T$, step sizes $\{\gamma_k, \tilde{\gamma}_k\}_{k \geq 1}$, momentum parameters $\rho_1, \rho_2 \in [0, 1)$ and regularization parameter $\delta \geq 0$.
Set $m_0 = 0, V_0 = 0$ and $\hat{V}_0 = 0$
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 $H_k$ using $\phi_{k+1}$.
    $m_k = \rho_1 m_{k-1} + (1 - \rho_1) H_k$
    $V_k = \rho_2 V_{k-1} + (1 - \rho_2) H_k H_k^\top$
    $\hat{V}_k = \max\left(\hat{V}_{k-1}, \mathrm{Diag}(V_k)\right)$
    $A_k = \left[\delta I_d + \hat{V}_k\right]^{-1/2}$
    $\theta_{k+1} = \theta_k - \gamma_{k+1} A_k m_k$
end for
Output: $(\theta_k, \phi_k)_{0 \leq k \leq n}$
```

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

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

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

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, since $\nabla V\left(\theta_k\right) = \bar{\nabla} f\left(\theta_k, \phi^*\left(\theta_k\right)\right)$, we have:

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

where we used [33, Lemma 2.2] for the first term and [37, Lemma 11] for the second term.

For the second inequality, we have:

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

□

Theorem C.4. *Assume that H6 and H7 hold. Let $\theta_n \in \mathbb{R}^d$ be the n-th iterate of Algorithm 4, $\gamma_n = c_\gamma n^{-1/2}$ 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,*

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

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 4.5, we obtain:

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

where

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

Then, with [33, Lemma 2.3] and [17, Lemma 3], we derive:

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

□

We achieve a classical convergence rate of $\mathcal{O}(\log n/\sqrt{n})$ for Stochastic Bilevel Optimization problems. Two types of bias emerge in this context: firstly, the challenge of directly computing $\phi^*(\theta)$, and secondly, the necessity of estimating $[\nabla_\phi^2 g(\theta, \phi)]^{-1}$.

Our results extend those of [17] to the adaptive case, particularly Adagrad, RMSProp, and AMSGRAD. This provides convergence guarantees for the Alternating Stochastic Gradient Descent (ALSET) method. We can apply our convergence analysis to Stochastic Min-Max and Compositional Problems, as well as to the Actor-Critic method with linear value function approximation [49], which can be viewed as a special case of the Stochastic Bilevel algorithm.

C.2 Conditional Stochastic Optimization

We now consider a class of Conditional Stochastic Optimization:

$$\min_{\theta \in \mathbb{R}^d} V(\theta) := \mathbb{E}_\xi\left[f_\xi\left(\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta, \xi)\right]\right)\right], \tag{14}$$

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 (14) can be expressed as:

$$\nabla V(\theta) = \mathbb{E}_\xi\left[\left(\mathbb{E}_{\eta|\xi}\left[\nabla g_\eta(\theta, \xi)\right]\right)^\top \nabla f_\xi\left(\mathbb{E}_{\eta|\xi}\left[g_\eta(\theta, \xi)\right]\right)\right]. \tag{15}$$

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:

$$\widehat{\nabla} V(\theta; \xi, \{\eta_j\}_{j=1}^m) := \left(\frac{1}{m}\sum_{j=1}^{m} \nabla g_{\eta_j}(\theta, \xi)\right)^\top \nabla f_\xi\left(\frac{1}{m}\sum_{j=1}^{m} g_{\eta_j}(\theta, \xi)\right). \tag{16}$$

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 $\mathbb{E}_{\eta|\xi}\left[\left\|g_\eta(\theta, \xi) - \mathbb{E}_{\eta|\xi}\left[g_\eta(\theta, \xi)\right]\right\|^2\right] \leq \sigma_g^2$.

Lemma C.5. *([40, Lemma 2.2]) Under Assumptions H8 and H9, the following holds:*

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

Lemma C.6. *Under Assumption H8, we have:*

$$\|\nabla V(\theta) - \nabla V(\theta')\| \leq \left(\ell_{g,1}\ell_{f,0} + \ell_{g,0}^2 \ell_{f,1}\right) \|\theta - \theta'\| ,$$

$$\|\nabla V(\theta)\| \leq \ell_{g,0}\ell_{f,0} .$$

Proof. Denoting $G_\theta = \mathbb{E}_{\eta|\xi}\left[g_\eta(\theta,\xi)\right]$ and $\nabla G_\theta = \mathbb{E}_{\eta|\xi}\left[\nabla g_\eta(\theta,\xi)\right]$, we establish the smoothness of V and Boundness of ∇V.

Smoothness of V:

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

Boundness of ∇V:

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

□

Theorem C.7. *Assume that H8 and H9 hold. Let $\gamma_n = c_\gamma n^{-1/2}$, A_n denote the adaptive matrix in AMSGRAD and $\rho_1, \rho_2 \in [0,1)$. For any $n \ge 1$, let $R \in \{0,\dots,n\}$ be a uniformly distributed random variable. Then,*

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

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

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

Proof. This is an immediate implication of Theorem 4.6 using Lemmas C.5 and C.6. □

These results can also be extended to the Federated Conditional Stochastic Optimization problem [75], 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 $\mathbb{E}_{\xi_\ell} f_{\xi_\ell}^\ell(\cdot) : \mathbb{R}^q \to \mathbb{R}$ is the outer-layer function on the ℓ-th device with the randomness ξ_ℓ, and $\mathbb{E}_{\eta_\ell|\xi_\ell} g_{\eta_\ell}^\ell(\cdot,\xi_\ell) : \mathbb{R}^d \to \mathbb{R}^q$ 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 $f_{\xi_\ell}^\ell(\cdot)$ and $g_{\eta_\ell}^\ell(\cdot,\xi_\ell)$ for all L devices verify Assumptions H8 and H9, we obtain the same convergence rate.

The following Table 2 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 2: Bilevel and Conditional Stochastic Optimization with our biased adaptive SA framework.

Applications	Stochastic Bilevel Optimization	Conditional Stochastic Optimization
Problem	$\min_{\theta \in \mathbb{R}^d} \mathbb{E}_\xi \left[f(\theta, \phi^*(\theta); \xi) \right]$ s.t. $\phi^*(\theta) \in \operatorname{argmin}_{\phi \in \mathbb{R}^q} \mathbb{E}_\zeta \left[g(\theta, \phi; \zeta) \right]$	$\min_{\theta \in \mathbb{R}^d} \mathbb{E}_\xi \left[f_\xi \left(\mathbb{E}_{\eta \mid \xi} \left[g_\eta(\theta, \xi) \right] \right) \right]$
Gradient	$\nabla_\theta f(\theta, \phi^*(\theta)) - \nabla_{\theta\phi} g(\theta, \phi^*(\theta)) v^*$	$\mathbb{E}_\xi \left[\left(\mathbb{E}_{\eta \mid \xi} \left[\nabla g_\eta(\theta, \xi) \right] \right)^T \nabla f_\xi \left(\mathbb{E}_{\eta \mid \xi} \left[g_\eta(\theta, \xi) \right] \right) \right]$
Lipchitz Constant H2	$\ell_{f,1} + \frac{\ell_{g,1}\left(\ell_{f,1}+L_f\right)}{\mu_g} + \frac{\ell_{f,0}}{\mu_g} \left(\ell_{g,2} + \frac{\ell_{g,1}\ell_{g,2}}{\mu_g} \right)$	$\ell_{g,1}\ell_{f,0} + \ell_{g,0}^2 \ell_{f,1}$
Bias Control H3	$\ell_{g,1}\ell_{f,1} \frac{1}{\mu_g} \left(1 - \frac{\mu_g}{\ell_{g,1}} \right)^N$	$\frac{\ell_{g,0}^2 \ell_{f,1}^2 \sigma_g^2}{m}$
Gradient Bound H5	$\ell_{f,0} + \frac{\ell_{g,1}\ell_{f,0}}{\mu_g}$	$\ell_{g,0}\ell_{f,0}$
Convergence	$\mathcal{O}\left(\frac{\log n}{\sqrt{n}} + b_n \right)$	$\mathcal{O}\left(\frac{\log n}{\sqrt{n}} + b_n \right)$

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

$$\pi(f) = \frac{\lambda(\omega f)}{\lambda(\omega)} \ , \tag{17}$$

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

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

where $X^{1:N} = \left(X^1, \ldots, X^N\right)$ are independent draws from λ and the ω_N^i are called the normalized weights. [1] shows that the bias of the SNIS estimator can be expressed as:

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

This particular type of estimator can be found in the domain of Monte Carlo methods, particularly in the context of Bayesian inference and Sequential Monte Carlo methods.

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 $(X_t)_{t \ge 0}$. The distribution of X_0 has density χ with respect to the Lebesgue measure μ and for all $t \ge 0$, 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 $(Y_t)_{0 \le t \le T}$. The observations $Y_{0:t}$ are assumed to be independent conditionally on $X_{0:t}$ and, for all $0 \le t \le T$, 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,

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

where $s_{t,\theta}\left(x, x'\right) = \nabla_\theta \log\{m_\theta\left(x, x'\right) g_\theta\left(x, y_{t+1}\right)\}$ for $t > 0$ and by convention $s_{0,\theta}\left(x, x'\right) = \nabla_\theta \log g_\theta\left(x, y_0\right)$. 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:

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

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 [20], 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 $H_\theta\left(y_{0:T}\right)$ computed by the Forward-only implementation of FFBSm is biased. The bias and MSE of this estimator are of order $\mathcal{O}\left(1/N\right)$ [20], 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) [64] and its pseudo-marginal extension [34] and Parisian particle Gibbs (PPG) [13]. 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, $R : \mathcal{S} \times \mathcal{A} \to [0, R_{\text{max}}]$ 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:

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

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:

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

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 $\mathcal{O}(1 - \lambda)$ [44].

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

https://doi.org/10.52202/079017-0411

challenging or computationally expensive. The zeroth-order gradient oracle obtained by Gaussian smoothing [61] is given by:

$$H_\theta\left(X\right) = \frac{V(\theta + \tau X) - V(\theta)}{\tau} X \ , \tag{18}$$

where $\tau > 0$ is a smoothing parameter and $X \sim \mathcal{N}(0, I_d)$ a random Gaussian vector. [61, Lemma 3] provide the bias of this estimator:

$$\| \mathbb{E}\left[H_\theta\left(X\right)\right] - \nabla V\left(\theta\right) \| \le \frac{\tau}{2} L(d+3)^{3/2} \ . \tag{19}$$

The application of these zeroth-order gradient methods can be found in generative adversarial networks [58, 16].

D.5 Compressed Stochastic Approximation: Coordinate Sampling

The coordinate descent method is based on the iteration:

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

where $\{e_1, \dots, e_d\}$ is the canonical basis of $\mathbb{R}^d$ and $H_{\theta_n}\left(X_{n+1}\right)_j$ is the j-th coordinate of the gradient. The randomized coordinate selection rule chooses j_n uniformly from the set $\{1, 2, \dots, d\}$. Alternatively, the Gauss-Southwell selection rule [63] uses:

$$j_{n+1} := \operatorname*{argmax}_{j \in \{1,\dots,d\}} |H_{\theta_n}\left(X_{n+1}\right)_j| \ .$$

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 [53], 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 $(w_n^{(1)}, \dots, w_n^{(d)})$ defined as:

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

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

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

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 $D_n = \mathbb{E}[D(\zeta_{n+1}) | \mathcal{F}_n] = \mathrm{Diag}(w_n^{(1)}, \dots, w_n^{(d)})$. 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 $\beta_{n+1} \le 1$ 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.

https://doi.org/10.52202/079017-0411

E Experiment details and supplementary results

E.1 Experiment with a Synthetic Time-Dependent Bias

To illustrate Theorem 4.1 and the impact of bias, we consider in Figure 3 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 with variance $\sigma^2 = 0.01$ 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 $\mathcal{O}(n^{-1/2} + n^{-r})$.

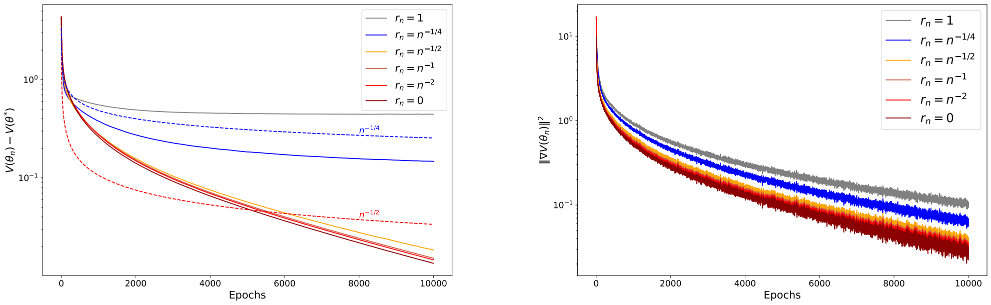

Figure 3: Value of $V(\theta_n) - V(\theta^*)$ (on the left) and $\|\nabla V(\theta_n)\|^2$ (on the right) with Adagrad for different values of $r_n = n^{-r}$ and a learning rate $\gamma_n = n^{-1/2}$. The dashed curve corresponds to the expected convergence rate $\mathcal{O}(n^{-1/4})$ for $r = 1/4$ and $\mathcal{O}(n^{-1/2})$ for $r \geq 1/2$.

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. First, note that the impact of a constant bias term ($r_n = 1$) on the risk and the norm of gradients 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.

E.2 Additional Experiments of 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 [76] 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 3. 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 3: 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 $\mathcal{O}(n^{-\alpha})$ at iteration n. We vary the value of α for both FashionMNIST and CIFAR-100.

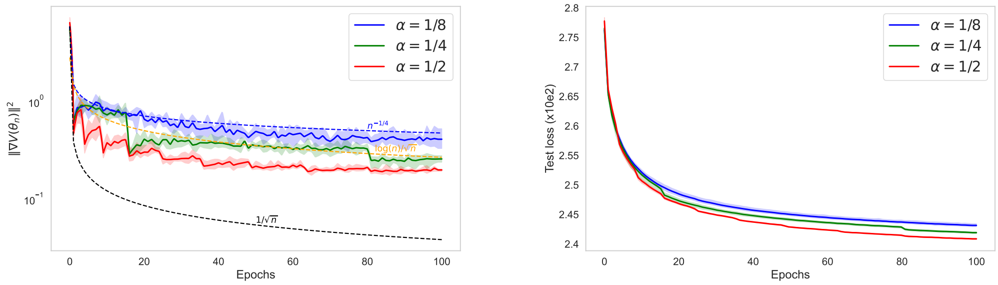

Figure 4: IWAE on the FashionMNIST Dataset with Adagrad for different values of α. Bold lines represent the mean over 5 independent runs.

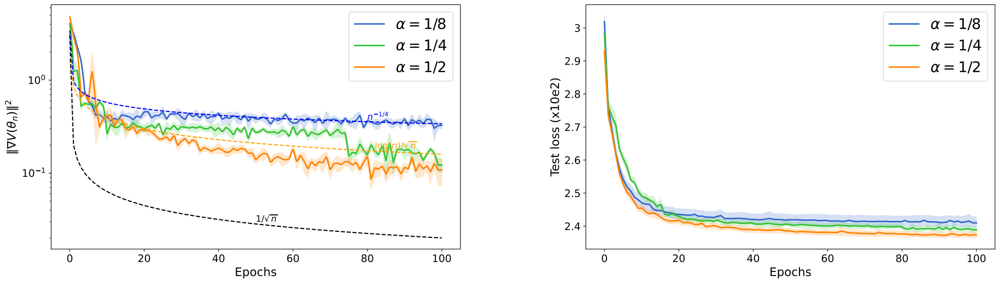

Figure 5: IWAE on the FashionMNIST Dataset with RMSProp for different values of α. Bold lines represent the mean over 5 independent runs.

https://doi.org/10.52202/079017-0411

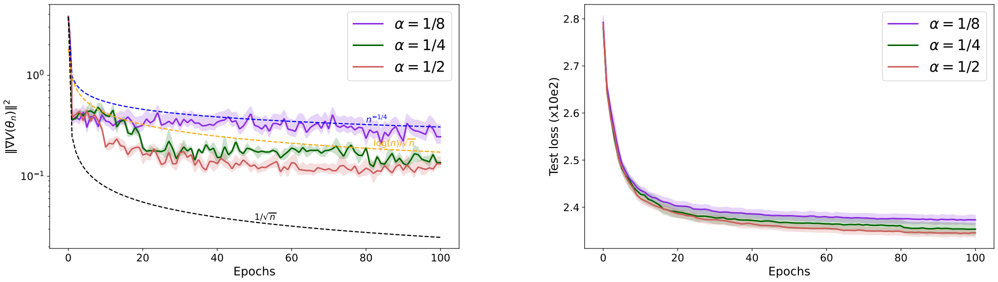

Figure 6: IWAE on the FashionMNIST Dataset with Adam for different values of α. Bold lines represent the mean over 5 independent runs.

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 $\mathcal{O}(n^{-1/4})$ for $\alpha = 1/8$, and $\mathcal{O}(\log n/\sqrt{n})$ 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 $\mathcal{O}(n^{-1/2})$ 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 $\mathcal{O}(n^{-1/2})$ convergence rate. For all figures, with a larger α, the convergence in both the squared gradient norm and negative log-likelihood occurs more rapidly.

Additional Experiments on CIFAR-10 Dataset.

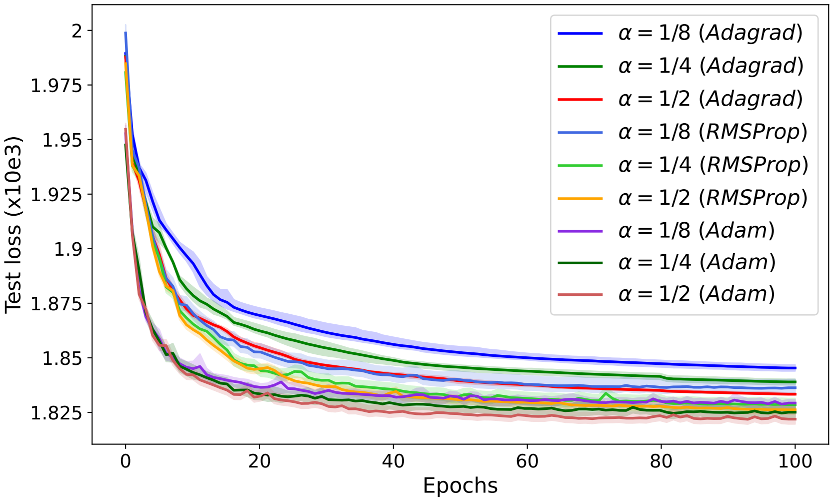

Figure 7: Negative Log-Likelihood on the test set on the CIFAR-10 Dataset for IWAE with Adagrad, RMSProp, and Adam. Bold lines represent the mean over 5 independent runs.

The effect of C_γ.

Figure 8 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 2. 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 $C_\gamma = 0.01$.

https://doi.org/10.52202/079017-0411

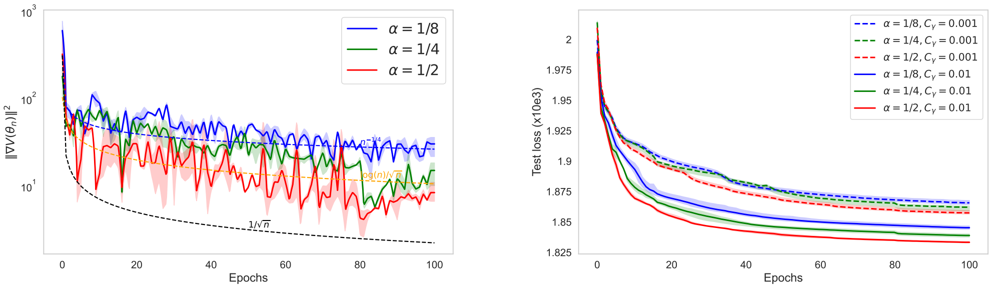

Figure 8: IWAE on the CIFAR-10 Dataset with Adagrad for different values of α and C_γ. Bold lines represent the mean over 5 independent runs.

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 9.

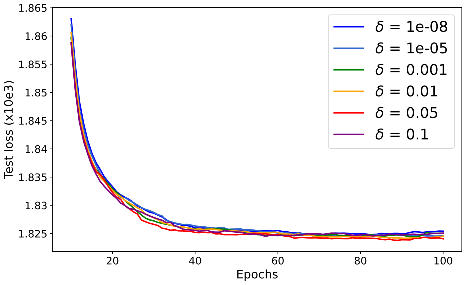

Figure 9: IWAE on the CIFAR-10 Dataset with Adam for different values of δ. Lines represent the mean over 5 independent runs.

Our experimental results align with prior work [77, 67, 73], affirming the consistent impact of δ. Notably, we find that employing $\delta = 5 \times 10^{-2}$ yields improved performance in IWAE.

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 10. 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.

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.

https://doi.org/10.52202/079017-0411

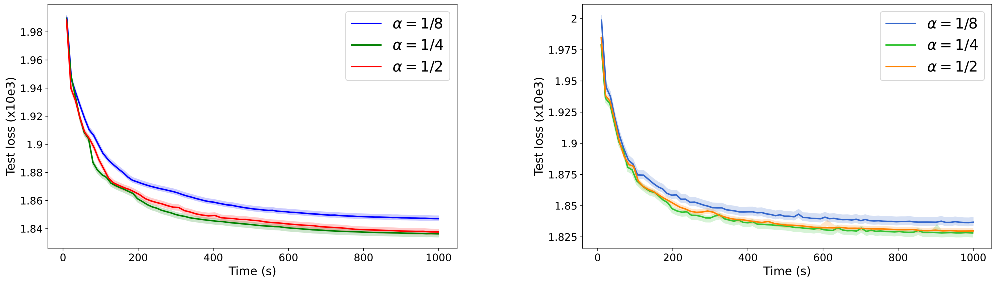

Figure 10: Negative Log-Likelihood on the test set of the CIFAR-10 Dataset for IWAE with Adagrad (on the left) RMSProp (on the right) for Different Values of α over time (in seconds). Bold lines represent the mean over 5 independent runs.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper's contributions are mentioned in the abstract and clearly detailed in the introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In our paper, we discuss the limitations of our results. Furthermore, for each assumption, we also discuss where it can be verified and provide the limitations of these assumptions.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution

is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper provides a comprehensive set of assumptions with discussions, along with complete proofs for each theoretical result in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Our work is primarily theoretical, and the experiments are conducted to illustrate our results. The paper provides detailed descriptions of the experimental setup, parameters, and methodologies, ensuring that the main results can be reproduced and verified.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case

https://doi.org/10.52202/079017-0411

of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code supporting our experiments is available on GitHub, as referenced in the Experiments section. All data used in our study is publicly accessible (CIFAR dataset).

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper provides comprehensive details on the experimental setup, including hyperparameters, all optimizer algorithms, and other relevant algorithms with pseudo code, ensuring the reproducibility of the results.

   Guidelines:

https://doi.org/10.52202/079017-0411

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All algorithms used to illustrate the results are run 5 times, and the mean and significance of all results are plotted, ensuring an appropriate representation of statistical significance.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides detailed information on the computer resources required for all experiments, including the type of compute workers (CPU or GPU), and approximate time of execution.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

https://doi.org/10.52202/079017-0411

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research conducted in the paper aligns with the NeurIPS Code of Ethics, ensuring adherence to ethical guidelines throughout the study.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: Given that the paper is primarily theoretical and the experiments are conducted solely for illustrative purposes, the work does not involve significant broader societal impacts to discuss. The focus of the paper is on theoretical contributions rather than practical applications with societal implications.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper is theoretical and does not involve the release of data or models that have a high risk for misuse. Therefore, no safeguards were necessary.

Guidelines:

- The answer NA means that the paper poses no such risks.

https://doi.org/10.52202/079017-0411

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: The paper does not use existing assets; therefore, no creators or original owners need to be credited, and no license or terms of use need to be mentioned or respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not introduce new assets; therefore, there is no documentation provided alongside any assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

https://doi.org/10.52202/079017-0411

Justification: The paper does not involve crowdsourcing experiments or research with human subjects, so there are no instructions provided to participants or details about compensation.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve research with human subjects; thus, there are no potential risks, disclosure of risks, or IRB approvals to describe.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-0411