THE NUMBER OF STEPS NEEDED FOR NONCONVEX OPTIMIZATION OF AN OPTIMIZER IS A RATIONAL FUNCTION OF BATCH SIZE

Anonymous authors
Paper under double-blind review

ABSTRACT

Recently, convergence as well as convergence rate analyses of optimizers for nonconvex optimization have been widely studied. Meanwhile, numerical evaluations for the optimizers have precisely clarified the relationship between batch size and the number of steps needed for training deep neural networks. The main contribution of this paper is to show theoretically that the number of steps needed for nonconvex optimization of each of the optimizers can be expressed as a rational function of batch size. Having these rational functions leads to two particularly important facts, which were validated numerically in previous studies. The first fact is that there exists an optimal batch size such that the number of steps needed for nonconvex optimization is minimized. This implies that using larger batch sizes than the optimal batch size does not decrease the number of steps needed for nonconvex optimization. The second fact is that the optimal batch size depends on the optimizer. In particular, it is shown theoretically that momentum and Adam-type optimizers can exploit larger optimal batches and further reduce the minimum number of steps needed for nonconvex optimization than the stochastic gradient descent optimizer.

1 INTRODUCTION

One way to train deep neural networks is to find the model parameters of the deep neural networks that minimize loss functions called the expected risk and empirical risk using first-order optimization methods (Bottou et al., 2018, Section 4). The simplest optimizer is stochastic gradient descent (SGD) (Robbins & Monro, 1951; Zinkevich, 2003; Nemirovski et al., 2009; Ghadimi & Lan, 2012; 2013). There have been many deep learning optimizers to accelerate SGD, such as momentum methods (Polyak, 1964; Nesterov, 1983) and adaptive methods, e.g., Adaptive Gradient (AdaGrad) (Duchi et al., 2011), Root Mean Square Propagation (RMSProp) (Tieleman & Hinton, 2012), Adaptive Moment Estimation (Adam) (Kingma & Ba, 2015), and Adaptive Mean Square Gradient (AMSGrad) (Reddi et al., 2018) (Table 2 in (Schmidt et al., 2021) lists useful deep learning optimizers).

Convergence and convergence rate analyses of deep learning optimizers have been widely studied for convex optimization (Zinkevich et al., 2010; Kingma & Ba, 2015; Reddi et al., 2018; Luo et al., 2019; Mendler-Dünner et al., 2020). Meanwhile, theoretical investigation of deep learning optimizers for nonconvex optimization is needed so that these optimizers can be put into practice for nonconvex optimization in deep learning (Xu et al., 2015; Arjovsky et al., 2017; Vaswani et al., 2017).

Convergence analyses of SGD for nonconvex optimization were studied in (Fehrman et al., 2020; Chen et al., 2020; Scaman & Malherbe, 2020; Loizou et al., 2021) (see (Gower et al., 2021; Loizou et al., 2021) for convergence analyses of SGD for two classes of nonconvex optimization problems, quasar-convex and Polyak–Lojasiewicz optimization problems). For example, Theorem 11 in (Scaman & Malherbe, 2020) indicates that SGD with a diminishing learning rate $\alpha_k = 1/\sqrt{k}$ has $\mathcal{O}(1/\sqrt{K})$ convergence, where K denotes the number of steps. Convergence analyses of SGD depending on the batch size were presented in (Chen et al., 2020). In particular, Theorem 3.2 in (Chen et al., 2020) indicates that running SGD with a diminishing learning rate $\alpha_k = 1/k$ and large batch size for sufficiently many steps leads to convergence to a local minimizer of a sum of loss functions.

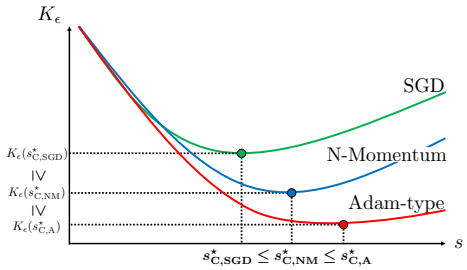

Figure 1: Relationships between the optimizers in terms of the results in Table 1 (relations $s^\star_{\mathrm{C,SGD}} \leq s^\star_{\mathrm{C,NM}} \leq s^\star_{\mathrm{C,A}}$ hold generally, but those of $K_\epsilon(s^\star_{\mathrm{C,SGD}}) \geq K_\epsilon(s^\star_{\mathrm{C,NM}}) \geq K_\epsilon(s^\star_{\mathrm{C,A}})$ depend on momentum coefficient β)

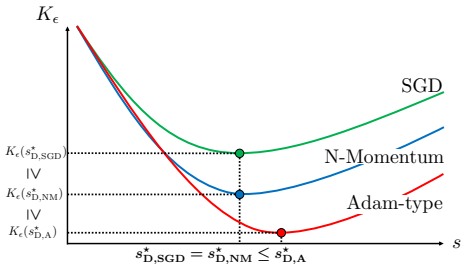

Figure 2: Relationships between the optimizers in terms of the results in Table 2 (relations $s^\star_{\mathrm{D,SGD}} = s^\star_{\mathrm{D,NM}} \leq s^\star_{\mathrm{D,A}}$ hold generally, but those of $K_\epsilon(s^\star_{\mathrm{D,SGD}}) \geq K_\epsilon(s^\star_{\mathrm{D,NM}}) \geq K_\epsilon(s^\star_{\mathrm{D,A}})$ depend on momentum coefficient β)

Convergence analyses of adaptive methods for nonconvex optimization were studied in (Fang et al., 2018; Chen et al., 2019; Zhuang et al., 2020; Iiduka, 2021). In (Chen et al., 2019), it was shown that generalized Adam, which includes the Heavy-ball method, AdaGrad, RMSProp, AMSGrad, and AdaGrad with First Order Momentum (AdaFom), using a diminishing learning rate $\alpha_k = 1/\sqrt{k}$ has an $\mathcal{O}(\log K/\sqrt{K})$ convergence rate. AdaBelief (named for adapting stepsizes by the belief in observed gradients) using $\alpha_k = 1/\sqrt{k}$ has $\mathcal{O}(\log K/\sqrt{K})$ convergence (Zhuang et al., 2020). In (Iiduka, 2021), a method was presented to unify useful adaptive methods such as AMSGrad and AdaBelief, and it was shown that the method with $\alpha_k = 1/\sqrt{k}$ has an $\mathcal{O}(1/\sqrt{K})$ convergence rate, which improves on the results in (Chen et al., 2019; Zhuang et al., 2020).

Meanwhile, in (Shallue et al., 2019), it was studied how increasing the batch size affects the performances of SGD, SGD with momentum (Polyak, 1964; Rumelhart et al., 1986), and Nesterov momentum (Nesterov, 1983; Sutskever et al., 2013). The relationships between batch size and performance for Adam and K-FAC (Kronecker-Factored Approximate Curvature (Martens & Grosse, 2015)) were studied in (Zhang et al., 2019). In both studies, it was numerically shown that increasing batch size tends to decrease the number of steps K needed for training deep neural networks, but with diminishing returns. Moreover, it was shown that SGD with momentum and Nesterov momentum can exploit larger batches than SGD (Shallue et al., 2019), and that K-FAC and Adam can exploit larger batches than SGD with momentum (Zhang et al., 2019). Thus, it was shown that momentum and adaptive methods can significantly reduce the number of steps K needed for training deep neural networks (Shallue et al., 2019, Figure 4), (Zhang et al., 2019, Figure 5). In (Smith et al., 2018), it was numerically shown that using enormous batch sizes leads to reducing the number of parameter updates and model training time.

1.1 Motivation

As indicated above, the performance of the optimizer strongly depends on not only learning rate α_k but also the batch size, s. However, the previous studies did not clarify any relationship between α_k and s. Moreover, the previous studies did not show theoretically the relationship between batch size s and the performance of the optimizer. Hence, the motivation of this paper is to clarify theoretically the relationship between α_k and s and that between s and the number of steps, K, needed for nonconvex optimization of the optimizer.

1.2 Notation

$\mathbb{N}$ denotes the set of nonnegative integers. Let $n \in \mathbb{N}\backslash\{0\}$. We define $[n] := \{1, 2, \ldots, n\}$. $\mathbb{R}^d$ denotes d-dimensional Euclidean space with inner product $\langle \cdot, \cdot \rangle$ inducing norm $\|\cdot\|$. Let $\mathbb{S}^d_{++}$ be the set of $d \times d$ symmetric positive-definite matrices and let $\mathbb{D}^d$ be the set of $d \times d$ diagonal matrices:

$\mathbb{D}^d = \{M \in \mathbb{R}^{d\times d} \colon M = \mathsf{diag}(x_i),\ x_i \in \mathbb{R}\ (i \in [d])\}$. For a random variable Z, we use $\mathbb{E}[Z]$ to indicate its expectation. Throughout this paper, let $\epsilon > 0$, $\alpha > 0$, $\tilde{b} := 1 - b$ $(b \in (0,1))$, $\tilde{\gamma} := 1 - \gamma$ $(\gamma \in [0,1))$, and $H \geq h_0^* > 0$. The number of samples is denoted by n, L_i is the Lipschitz constant of $\nabla f_i \colon \mathbb{R}^d \to \mathbb{R}^d$ $(i \in [n])$, and L denotes the maximum value of L_i. $\nabla f \colon \mathbb{R}^d \to \mathbb{R}^d$ denotes the gradient of a nonconvex loss function $f \colon \mathbb{R}^d \to \mathbb{R}$. D is the upper bound of $(x_{k,i} - x_i)^2$ $((x_i) \in \mathbb{R}^d)$, where $(\boldsymbol{x}_k)_{k\in\mathbb{N}} = ((x_{k,i}))_{k\in\mathbb{N}}$ is generated by an optimizer. A_α and B_α are positive constants depending on a learning rate α and C_β is a positive constant depending on a momentum coefficient β (see Theorem 3.1 for detailed definitions of the constants). A batch size $s^\star$ is said to be optimal if the number of steps K needed for nonconvex optimization is minimized at $s^\star$.

1.3 Contribution

The contribution of this paper is to construct a theory guaranteeing the useful numerical results in (Shallue et al., 2019; Zhang et al., 2019). Table 1 (resp. Table 2) summarizes our results for SGD, Nesterov momentum (N-Momentum), and Adam-type optimizers with a constant learning rate rule (resp. diminishing learning rate rule), described in Theorem 3.1 (resp. Theorem 3.2). See Theorem A.2 in Appendix for other result for the optimizers with a diminishing learning rate rule. Figure 1 (resp. Figure 2) visualizes the relationships between the optimizers for the results shown in Table 1 (resp. Table 2) for an appropriately set momentum coefficient β.

Table 1: Relationship between batch size s and the number of steps K_ϵ needed for nonconvex optimization in the sense of (1) of optimizers with constant learning rates

	Constant Learning Rate Rule $(\alpha_k(s) = \frac{\alpha}{s},\ \beta_k = \beta \in [0,b] \subset [0,1))$		
	Rational Function	Optimal Batch Size $s^\star$	Minimum Steps $K_\epsilon(s^\star)$
SGD	$K_\epsilon = \dfrac{A_\alpha s^2}{\epsilon^2 s - B_\alpha}$	$\dfrac{dDL^2n^2\alpha}{\epsilon^2}$	$\dfrac{(dDLn)^2}{\epsilon^4}$
N-Momentum	$K_\epsilon = \dfrac{A_\alpha s^2}{(\epsilon^2 - C_\beta)s - B_\alpha}$	$\dfrac{dDL^2n^2\alpha}{\tilde{b}\epsilon^2 - dDLn\beta}$	$\dfrac{(dDLn)^2}{(\tilde{b}\epsilon^2 - dDLn\beta)^2}$
Adam-type	$K_\epsilon = \dfrac{A_\alpha s^2}{(\epsilon^2 - C_\beta)s - B_\alpha}$	$\dfrac{dDL^2n^2\alpha}{\tilde{\gamma}^2(\tilde{b}\epsilon^2 - dDLn\beta)h_0^*}$	$\dfrac{(dDLn)^2 H}{\tilde{\gamma}^2(\tilde{b}\epsilon^2 - dDLn\beta)^2 h_0^*}$

Table 2: Relationship between batch size s and the number of steps K_ϵ needed for nonconvex optimization in the sense of (1) of optimizers with diminishing learning rates

	Diminishing Learning Rate Rule $(\alpha_k(s) = \frac{\alpha}{s\sqrt{k}},\ \beta_k = \beta \in [0,b] \subset [0,1))$		
	Rational Function	Optimal Batch Size $s^\star$	Minimum Steps $K_\epsilon(s^\star)$
SGD	$K_\epsilon = \left\{\dfrac{A_\alpha s^2 + B_\alpha}{\epsilon^2 s}\right\}^2$	$\sqrt{2}Ln\alpha$	$\dfrac{2(dDLn)^2}{\epsilon^4}$
N-Momentum	$K_\epsilon = \left\{\dfrac{A_\alpha s^2 + B_\alpha}{(\epsilon^2 - C_\beta)s}\right\}^2$	$\sqrt{2}Ln\alpha$	$\dfrac{2(dDLn)^2}{(\tilde{b}\epsilon^2 - dDLn\beta)^2}$
Adam-type	$K_\epsilon = \left\{\dfrac{A_\alpha s^2 + B_\alpha}{(\epsilon^2 - C_\beta)s}\right\}^2$	$\dfrac{\sqrt{2}Ln\alpha}{\tilde{\gamma}\sqrt{Hh_0^*}}$	$\dfrac{2(dDLn)^2 H}{\tilde{\gamma}^2(\tilde{b}\epsilon^2 - dDLn\beta)^2 h_0^*}$

The main contribution of this paper is to clarify that the number of steps $K = K_\epsilon$ needed for nonconvex optimization in the sense of an ϵ-approximation, [1]

$$\min_{k\in[K]} \mathbb{E}\left[\|\nabla f(\boldsymbol{x}_k)\|^2\right] \leq \epsilon^2, \tag{1}$$

of one of SGD, N-Momentum, and Adam-type optimizers can be expressed as a rational function of batch size s (see the "Rational Function" columns of Tables 1 and 2).

[1] Jensen's inequality guarantees that (1) implies that $\min_{k\in[K]} \mathbb{E}\left[\|\nabla f(\boldsymbol{x}_k)\|\right] \leq \epsilon$.

The explicit forms of the rational functions imply the following two significant facts:

(I) There exists an optimal batch size $s^\star$ such that $K_\epsilon(s)$ is minimized. This fact guarantees theoretically the existences of the diminishing returns shown in (Shallue et al., 2019, Figure 4), (Zhang et al., 2019, Figure 8), which are such that increasing the batch size does not decrease the number of steps K_ϵ.

(II) The optimal batch size $s^\star$ and the minimum number of steps $K_\epsilon(s^\star)$ depend on the optimizer. In particular, N-Momentum and Adam-type optimizers can exploit the same sized or larger batches ($s^\star$ in Tables 1 and 2 and Figures 1 and 2) than can SGD. Furthermore, the dependence of N-Momentum and Adam-type optimizers on β allows them to reduce the minimum number of steps ($K_\epsilon(s^\star)$ in Tables 1 and 2 and Figures 1 and 2) more than can SGD (see Section 3 for details).

The relationship between α_k and s and the existence of the optimal batch size $s^\star$ lead to the learning rate $\alpha_k^\star$ (e.g., the constant learning rate is $\alpha_k^\star = \alpha/s^\star$) for nonconvex optimization in the sense of an ϵ-approximation (1). This result justifies that this nonconvex optimization requires not only for the batch size to be set appropriately but also the learning rate.

Comparisons of Optimal Batch Sizes for Different Learning Rate Rules

Tables 1 and 2 ensure that $K_\epsilon(s^\star)$ for the optimizer using constant learning rates is almost the same as $K_\epsilon(s^\star)$ for the optimizer using diminishing learning rates. Meanwhile, we would like to emphasize that the optimal batch size $s_{\mathrm{C}}^\star$ for the optimizer using constant learning rates depends on β, and the optimal batch size $s_{\mathrm{D}}^\star$ for the optimizer using diminishing learning rates does not depend on β. For example, under the precision accuracy $\epsilon = 10^{-1}$, we can know the optimal batch sizes for N-Momentum with the frequently used parameter value $\alpha = 10^{-3}$ are respectively

$$s_{\mathrm{C,NM}}^\star = \frac{dDL^2n^2\epsilon^3}{\tilde{b}\epsilon^2 - dDLn\beta} \text{ and } s_{\mathrm{D,NM}}^\star = \sqrt{2}Ln\epsilon^3$$

before implementing N-Momentum.

2 Nonconvex Optimization and Optimizers

This section provides the assumptions used in this paper and states optimizers for a nonconvex optimization problem under the assumptions.

2.1 Assumptions Regarding Loss Function and Gradient Estimation

This paper considers optimization problems under the following assumptions.

Assumption 2.1

(A1) [Loss function] $f_i\colon \mathbb{R}^d \to \mathbb{R}$ ($i \in [n]$) is differentiable and $f\colon \mathbb{R}^d \to \mathbb{R}$ is defined for all $\boldsymbol{x} \in \mathbb{R}^d$ by

$$f(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x}),$$

where n denotes the number of samples.

(A2) [Gradient estimation] For each iteration k, optimizers sample a batch $\mathcal{S}_k \subset [n]$ of size $s := |\mathcal{S}_k|$ independently of k and estimate the full gradient ∇f as

$$\nabla f_{\mathcal{S}_k} := \frac{1}{s}\sum_{i\in\mathcal{S}_k} \nabla f_i.$$

(A3) [Gradient boundedness] There exists a positive number G such that, for all $\boldsymbol{x} \in X$,

$$\mathbb{E}\left[\|\nabla f_{\mathcal{S}_k}(\boldsymbol{x})\|^2\right] \leq G^2, \tag{2}$$

where X is a subset of $\mathbb{R}^d$.

Assumption (A1) is a standard one for nonconvex optimization in deep neural networks (see, e.g., (Chen et al., 2019, (2)) and (Fang et al., 2018, (1.2))). Assumption (A2) is needed for the optimizers to work (see, e.g., (Chen et al., 2019, Section 2) and (Fang et al., 2018, Notation section)). Assumption (A3) is used to analyze the optimizers (see, e.g., (Chen et al., 2019, Section 2)). Assumption (A3) holds if each of the following holds:

(G1) $X \subset \mathbb{R}^d$ is bounded, the gradient ∇f_i is Lipschitz continuous with Lipschitz constant L_i, and $S_i := \{\boldsymbol{x}^* \in \mathbb{R}^d \colon \nabla f_i(\boldsymbol{x}^*) = \boldsymbol{0}\} \neq \emptyset$ $(i \in [n])$, where $L := \max_{i\in[n]} L_i$. (If we define $G_{k,L} := \sup_{\boldsymbol{x}\in X} \sum_{i\in\mathcal{S}_k} \|\nabla f_i(\boldsymbol{x})\|$, then we can take $G := \sup_{k\in\mathbb{N}} G_{k,L}$.)

(G2) $X \subset \mathbb{R}^d$ is bounded and closed. (If we define $G_k := \sup_{\boldsymbol{x}\in X} \sum_{i\in\mathcal{S}_k} \|\nabla f_i(\boldsymbol{x})\|$, then we can take $G := \sup_{k\in\mathbb{N}} G_k$.)

For example, under (G1), the Lipschitz continuity of ∇f_i, together with the definition of L, ensures that, for all $\boldsymbol{x} \in \mathbb{R}^d$ and all $i \in [n]$, $\|\nabla f_i(\boldsymbol{x})\| = \|\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x}^*)\| \leq L_i\|\boldsymbol{x} - \boldsymbol{x}^*\| \leq L\|\boldsymbol{x} - \boldsymbol{x}^*\|$. Accordingly, $\sum_{i\in\mathcal{S}_k} \|\nabla f_i(\boldsymbol{x})\| \leq Ls\|\boldsymbol{x} - \boldsymbol{x}^*\| \leq Ln\|\boldsymbol{x} - \boldsymbol{x}^*\|$. The definition of $\nabla f_{\mathcal{S}_k}$ and the triangle inequality imply that there exists G such that $\|\nabla f_{\mathcal{S}_k}(\boldsymbol{x})\|^2 \leq (\sum_{i\in\mathcal{S}_k} \|\nabla f_i(\boldsymbol{x})\|)^2 \leq (Ln\|\boldsymbol{x} - \boldsymbol{x}^*\|)^2 \leq G^2$, i.e., (A3) holds (see Proposition A.1 in Appendix for details).

2.2 Nonconvex Optimization

This paper considers the following problem (Chen et al., 2019; Zhuang et al., 2020).

Problem 2.1 *Under Assumption 2.1, we would like to find a point $\boldsymbol{x}^\star \in \mathbb{R}^d$ such that*

$$\boldsymbol{x}^\star \in X^\star := \left\{\boldsymbol{x} \in \mathbb{R}^d \colon \nabla f(\boldsymbol{x}) = \boldsymbol{0}\right\}.$$

See the third and fourth paragraphs of Section 1 for the previous studies on Problem 2.1.

2.3 Optimizers

There are many optimizers (Schmidt et al., 2021, Table 2). In this paper, we consider the following algorithm (Algorithm 1), which is a unified algorithm for useful optimizers, for example, N-Momentum (Nesterov, 1983; Sutskever et al., 2013), AMSGrad (Reddi et al., 2018; Chen et al., 2019), AMSBound (Luo et al., 2019), modified Adam (M-Adam) (Kingma & Ba, 2015; Iiduka, 2021), and AdaBelief (Zhuang et al., 2020), listed in Table 4 in Appendix.

Algorithm 1 Optimizer for solving Problem 2.1

Require: $(\alpha_k)_{k\in\mathbb{N}} \subset (0, +\infty)$, $(\beta_k)_{k\in\mathbb{N}} \subset [0, b] \subset [0, 1)$, $\gamma \in [0, 1)$
$k \leftarrow 0$, $\boldsymbol{x}_0, \boldsymbol{m}_{-1} := \boldsymbol{0} \in \mathbb{R}^d$, $\mathsf{H}_0 \in \mathbb{S}^d_{++} \cap \mathbb{D}^d$, $\mathcal{S}_0 \subset [n]$
loop
  $\boldsymbol{m}_k := \beta_k \boldsymbol{m}_{k-1} + (1 - \beta_k)\nabla f_{\mathcal{S}_k}(\boldsymbol{x}_k)$
  $\hat{\boldsymbol{m}}_k := \dfrac{\boldsymbol{m}_n}{1 - \gamma^{k+1}}$
  $\mathsf{H}_k \in \mathbb{S}^d_{++} \cap \mathbb{D}^d$ (see Table 4 for examples of H_k)
  Find $\mathbf{d}_k \in \mathbb{R}^d$ that solves $\mathsf{H}_k \mathbf{d} = -\hat{\boldsymbol{m}}_k$
  $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k + \alpha_k \mathbf{d}_k$
  $k \leftarrow k + 1$
end loop

The useful optimizers, such as N-Momentum, AMSGrad, AMSBound, M-Adam, and AdaBelief (Table 4), all satisfy the following conditions:

Assumption 2.2 The sequence $(\mathsf{H}_k)_{k\in\mathbb{N}} \subset \mathbb{S}^d_{++} \cap \mathbb{D}^d$, with $\mathsf{H}_k := \mathsf{diag}(h_{k,i})$, in Algorithm 1 satisfies the following conditions:

(A4) $h_{k+1,i} \geq h_{k,i}$ for all $k \in \mathbb{N}$ and all $i \in [d]$;

(A5) For all $i \in [d]$, a positive number H_i exists such that $\sup_{k\in\mathbb{N}} \mathbb{E}[h_{k,i}] \leq H_i$.

Moreover, the following condition holds:

(A6) $D := \max_{i\in[d]} \sup_{k\in\mathbb{N}}(x_{k,i} - x_i)^2 < +\infty$, where $\boldsymbol{x} := (x_i) \in \mathbb{R}^d$ and $(\boldsymbol{x}_k)_{k\in\mathbb{N}} := ((x_{k,i}))_{k\in\mathbb{N}}$ is the sequence generated by Algorithm 1.

We define

$$h_0^* := \min_{i\in[d]} h_{0,i} \text{ and } H := \max_{i\in[d]} H_i,$$

where $h_{k,i}$ and H_i are defined as in Assumption 2.2. Assumption (A6) is assumed in (Nemirovski et al., 2009, p.1574), (Kingma & Ba, 2015, Theorem 4.1), (Reddi et al., 2018, p.2), (Luo et al., 2019, Theorem 4), and (Zhuang et al., 2020, Theorem 2.1). If (A6) holds, then there exists a bounded set $X \subset \mathbb{R}^d$ such that $(\boldsymbol{x}_k)_{k\in\mathbb{N}} \subset X$. Accordingly, the Lipschitz continuity of ∇f_i, the nonemptiness of S_i, and (A6) (i.e., (G1)) imply that (A3) with $G := Ln\sqrt{dD}$ holds (see Proposition A.1 in Appendix for details). The previous results in (Chen et al., 2019, p.29), (Zhuang et al., 2020, p.18), and (Iiduka, 2021) show that $(\mathsf{H}_k)_{k\in\mathbb{N}}$ in Table 4 satisfies (A4) and (A5). For example, AMSGrad (resp. M-Adam) satisfies (A4) and (A5) with $H = \sqrt{M}$ (resp. $H = \sqrt{M/(1-\zeta)}$), where $M := \sup_{k\in\mathbb{N}} \|\nabla f_{\mathcal{S}_k}(\boldsymbol{x}_k) \odot \nabla f_{\mathcal{S}_k}(\boldsymbol{x}_k)\| < +\infty$ (Iiduka, 2021). AMSBound coincides with AMSGrad using $H = \sup_{k\in\mathbb{N}} l_k^{-1}$. In general, the performance of AMSGrad with $H = \sqrt{M}$ differs from the that of AMSBound (i.e., AMSGrad with $H = \sup_{k\in\mathbb{N}} l_k^{-1}$) since the optimal batch size and the minimum number of steps depend on H, as seen in Tables 1 and 2 (see also (Luo et al., 2019, Section 5) for numerical comparisons of AMSGrad and AMSBound).

3 Main Results

This section gives our results (Theorems 3.1 and 3.2) indicating the relationship between batch size s and the number of steps K_ϵ needed for (1) for Algorithm 1 with each of constant and diminishing learning rates (see Tables 1 and 2 for the specific results in Theorems 3.1 and 3.2 with $G := Ln\sqrt{dD}$ (i.e., under condition (G1))).

3.1 Constant Learning Rate Rule

Theorem 3.1 *Suppose that Assumptions 2.1 and 2.2 hold and let $\epsilon > 0$ and $\alpha \in (0, 1]$.*

(i) *Consider Algorithm 1 with*

$$\alpha_k = \alpha_k(s) := \frac{\alpha}{s} \ (s > 0) \text{ and } \beta_k := \beta \in [0, b] \subset [0, 1).$$

Then, for all $K \geq 1$ and all $s > 0$,

$$\min_{k\in[K]} \mathbb{E}\left[\|\nabla f(\boldsymbol{x}_k)\|^2\right] \leq \underbrace{\frac{dDH}{2(1-b)\alpha}}_{A_\alpha} \frac{s}{K} + \underbrace{\frac{G^2\alpha}{2(1-b)(1-\gamma)^2 h_0^*}}_{B_\alpha} \frac{1}{s} + \underbrace{\frac{\sqrt{dD}G}{1-b}}_{C_\beta} \beta.$$

(ii) *Consider Algorithm 1 with*

$$\alpha_k = \alpha_k(s) = \frac{\alpha}{s} \ (s > 0) \text{ and } \beta_k := \beta < \min\left\{\frac{1-b}{\sqrt{dD}G}\epsilon^2, b\right\}.$$

Then, the number of steps K_ϵ needed to achieve an ϵ-approximation (1) is expressed as the following rational function of batch size s:

$$K_\epsilon(s) = \frac{A_\alpha s^2}{(\epsilon^2 - C_\beta)s - B_\alpha} \quad \left(s \in \left(\frac{B_\alpha}{\epsilon^2 - C_\beta}, +\infty\right)\right). \tag{3}$$

In particular, the minimum value of K_ϵ needed to achieve (1) is

$$K_\epsilon(s^\star) = \frac{4A_\alpha B_\alpha}{(\epsilon^2 - C_\beta)^2} = \frac{dDG^2H}{(1-\gamma)^2\{(1-b)\epsilon^2 - \sqrt{dD}G\beta\}^2 h_0^*}$$

when

$$s^\star = \frac{2B_\alpha}{\epsilon^2 - C_\beta} = \frac{G^2\alpha}{(1-\gamma)^2\{(1-b)\epsilon^2 - \sqrt{dD}G\beta\}h_0^*}.$$

3.1.1 DISCUSSION OF THEOREM 3.1

[Performance of Algorithm 1] SGD is Algorithm 1 with $\beta = \gamma = 0$ and $h_0^* = H = 1$, N-Momentum is Algorithm 1 with $\gamma = 0$ and $h_0^* = H = 1$, and the Adam-type optimizer is Algorithm 1 with $\gamma \in [0, 1)$ and $h_{k,i}$ defined by one of $\sqrt{\hat{v}_{k,i}}$, $\tilde{v}_{k,i}$, and $\sqrt{\hat{s}_{k,i}}$ (see Table 4). Theorem 3.1(i) indicates that, for all $K \geq 1$, all $\alpha \in (0, 1]$, all $\beta \in [0, b] \subset [0, 1)$, and all $s > 0$,

$$\min_{k\in[K]} \mathbb{E}\left[\|\nabla f(\boldsymbol{x}_k)\|^2\right] \leq \begin{cases} \frac{dD_{\mathrm{SGD}}}{2\alpha}\frac{s}{K} + \frac{G^2\alpha}{2}\frac{1}{s} & \text{(SGD)}, \\ \frac{dD_{\mathrm{NM}}}{2(1-b)\alpha}\frac{s}{K} + \frac{G^2\alpha}{2(1-b)}\frac{1}{s} + \frac{\sqrt{dD_{\mathrm{NM}}}G}{1-b}\beta & \text{(N-Momentum)}, \\ \frac{dD_{\mathrm{A}}H}{2(1-b)\alpha}\frac{s}{K} + \frac{G^2\alpha}{2(1-b)(1-\gamma)^2h_0^*}\frac{1}{s} + \frac{\sqrt{dD_{\mathrm{A}}}G}{1-b}\beta & \text{(Adam-type)}. \end{cases} \tag{4}$$

Note that D depends on the optimizer, which we distinguish by the notation D_{SGD}, D_{NM}, and D_{A}. For fixed s, if α and β are sufficiently small, (4) indicates that SGD, N-Momentum, and Adam-type optimizers have approximately $\mathcal{O}(1/K)$ convergence. For fixed s and K, if α is sufficiently small, the second term on the right-hand side of (4) will be small, whereas the first term will be large. Hence, there is no evidence that Algorithm 1 with a sufficiently small learning rate α would perform arbitrarily well. For fixed α and K, if s is sufficiently large, again the second term of the right-hand side of (4) will be small and the first term will be large. Hence, (4) indicates that there is no evidence that Algorithm 1 with a large batch size s performs better than with a smaller batch size.

[Existence of optimal batch size] The function $K_\epsilon(s)$ defined by (3) satisfies the following:

$$\frac{\mathrm{d}K_\epsilon(s)}{\mathrm{d}s} \begin{cases} < 0 & \text{if } s \in \left(\frac{B_\alpha}{\epsilon^2 - C_\beta}, s^\star\right), \\ = 0 & \text{if } s = s^\star = \frac{2B_\alpha}{\epsilon^2 - C_\beta}, \\ > 0 & \text{if } s \in (s^\star, +\infty). \end{cases}$$

The above shows that increasing the batch size initially decreases the number of steps K_ϵ needed to achieve (1). Then, there is an optimal batch size ($s = s^\star$) minimizing $K_\epsilon(s)$. We note that the optimal batch size depends on the upper bound defined by the right-hand side of (4).

[Comparison of optimal batch sizes] We assume that SGD, N-Momentum, and Adam-type optimizers all use the same G. For example, under (G1), we have $G = Ln\sqrt{dD}$, where $D = \max\{D_{\mathrm{SGD}}, D_{\mathrm{NM}}, D_{\mathrm{A}}\}$. From Theorem 3.1(ii), we find that

$$s^\star_{\mathrm{C,SGD}} = \frac{G^2\alpha}{\epsilon^2} \leq s^\star_{\mathrm{C,NM}} = \frac{G^2\alpha}{(1-b)\epsilon^2 - \sqrt{dD_{\mathrm{NM}}}G\beta}. \tag{5}$$

Moreover, if $(1-\gamma)^2 \leq 1/h_0^*$, then we have that

$$s^\star_{\mathrm{C,SGD}} = \frac{G^2\alpha}{\epsilon^2} \leq s^\star_{\mathrm{C,A}} = \frac{G^2\alpha}{(1-\gamma)^2\{(1-b)\epsilon^2 - \sqrt{dD_{\mathrm{A}}}G\beta\}h_0^*}. \tag{6}$$

Moreover, if $(1-\gamma)^2 \leq 1/h_0^*$ holds and if $D_{\mathrm{NM}} \leq D_{\mathrm{A}}$, then

$$s^\star_{\mathrm{C,NM}} \leq s^\star_{\mathrm{C,A}}.$$

The right-hand side of (4) with $\beta = 0$ is the smallest. Hence, it might seem that N-Momentum and Adam-type optimizers are not useful. However, using $\beta \neq 0$ leads to the finding that N-Momentum and Adam-type optimizers exploit larger batches than SGD, as seen in (5) and (6).

[Comparison of minimum numbers of steps] Theorem 3.1(ii) guarantees that the dependence of N-Momentum and Adam-type optimizers of β allows them to satisfy that

$$K_\epsilon(s^\star_{\mathrm{C,A}}) \leq K_\epsilon(s^\star_{\mathrm{C,NM}}) \leq K_\epsilon(s^\star_{\mathrm{C,SGD}}) \tag{7}$$

(see (27), (28), (29), (30), (31), and (32) in Appendix).

3.2 DIMINISHING LEARNING RATE RULE

Theorem 3.2 *Suppose that Assumptions 2.1 and 2.2 hold and let $\epsilon > 0$ and $\alpha \in (0, 1]$.*

(i) *Consider Algorithm 1 with*

$$\alpha_k = \alpha_k(s) := \frac{\alpha}{s\sqrt{k}} \ (s > 0) \text{ and } \beta_k := \beta \in [0, b] \subset [0, 1).$$

Then, for all $K \geq 1$ and all $s > 0$,

$$\min_{k\in[K]} \mathbb{E}\left[\|\nabla f(\boldsymbol{x}_k)\|^2\right] \leq \underbrace{\frac{dDH}{2(1-b)\alpha}}_{A_\alpha} \frac{s}{\sqrt{K}} + \underbrace{\frac{G^2\alpha}{(1-b)(1-\gamma)^2 h_0^*}}_{B_\alpha} \frac{1}{s\sqrt{K}} + \underbrace{\frac{\sqrt{dD}G}{1-b}}_{C_\beta}\beta.$$

(ii) *Consider Algorithm 1 with*

$$\alpha_k = \alpha_k(s) := \frac{\alpha}{s\sqrt{k}}\ (s > 0) \text{ and } \beta_k := \beta < \min\left\{\frac{1-b}{\sqrt{dD}G}\epsilon^2, b\right\}.$$

Then, the number of steps K_ϵ needed to achieve an ϵ-approximation (1) is expressed as the following rational function of batch size s:

$$K_\epsilon(s) = \left\{\frac{A_\alpha s^2 + B_\alpha}{(\epsilon^2 - C_\beta)s}\right\}^2. \tag{8}$$

In particular, the minimum value of K_ϵ needed to achieve (1) is

$$K_\epsilon(s^\star) = \frac{4A_\alpha B_\alpha}{(\epsilon^2 - C_\beta)^2} = \frac{2dDG^2H}{(1-\gamma)^2\{(1-b)\epsilon^2 - \sqrt{dD}G\beta\}^2 h_0^*}$$

when

$$s^\star = \sqrt{\frac{B_\alpha}{A_\alpha}} = \frac{\sqrt{2}G\alpha}{(1-\gamma)\sqrt{dDHh_0^*}}.$$

3.2.1 Discussion of Theorem 3.2

[Performance of Algorithm 1] Theorem 3.2(i) indicates that Algorithm 1 satisfies that, for all $K \geq 1$, all $\alpha \in (0, 1]$, all $\beta \in [0, b]$, and all $s > 0$,

$$\min_{k\in[K]} \mathbb{E}\left[\|\nabla f(\boldsymbol{x}_k)\|^2\right] \leq \begin{cases} \frac{dD_{\mathrm{SGD}}}{2\alpha}\frac{s}{\sqrt{K}} + \frac{G^2\alpha}{s\sqrt{K}} & \text{(SGD)}, \\ \frac{dD_{\mathrm{NM}}}{2(1-b)\alpha}\frac{s}{\sqrt{K}} + \frac{G^2\alpha}{(1-b)}\frac{1}{s\sqrt{K}} + \frac{\sqrt{dD_{\mathrm{NM}}}G}{1-b}\beta & \text{(N-Momentum)}, \\ \frac{dD_{\mathrm{A}}H}{2(1-b)\alpha}\frac{s}{\sqrt{K}} + \frac{G^2\alpha}{(1-b)(1-\gamma)^2h_0^*}\frac{1}{s\sqrt{K}} + \frac{\sqrt{dD_{\mathrm{A}}}G}{1-b}\beta & \text{(Adam-type)}. \end{cases}$$

By a similar argument to that in Section 3.1.1, SGD, N-Momentum, and Adam-type optimizers have approximately $\mathcal{O}(1/\sqrt{K})$ convergence and there is no evidence that Algorithm 1 with a large batch size s performs better than with a smaller batch size.

[Existence of optimal batch size] K_ϵ defined by (8) guarantees that there exists $s^\star$ such that $\mathrm{d}K_\epsilon(s^\star)/\mathrm{d}s = 0$, the same as seen in Section 3.1.1 for Theorem 3.1. This implies that there is an optimal batch size ($s = s^\star$) such that $K_\epsilon(s)$ is minimized.

[Comparison of optimal batch sizes] For simplicity, let us consider the case where (G1) holds. Theorem 3.2(ii) with $G = Ln\sqrt{dD}$ ensures that the optimal batch sizes for SGD, N-Momentum, and Adam-type optimizers satisfy that $s^\star_{\mathrm{D,SGD}} = \frac{\sqrt{2}G\alpha}{\sqrt{dD_{\mathrm{SGD}}}} = \sqrt{2}Ln\alpha = \frac{\sqrt{2}G\alpha}{\sqrt{dD_{\mathrm{NM}}}} = s^\star_{\mathrm{D,NM}}$. Furthermore, if $(1-\gamma)^2 \leq 1/(Hh_0^*)$, then $s^\star_{\mathrm{D,SGD}} = s^\star_{\mathrm{D,NM}} \leq s^\star_{\mathrm{D,A}} = \frac{\sqrt{2}Ln\alpha}{(1-\gamma)\sqrt{Hh_0^*}}$. Therefore, N-Momentum and Adam-type optimizers exploit the same sized or larger batches than SGD. Here, we notice that $s^\star_{\mathrm{C,SGD}}$, $s^\star_{\mathrm{C,NM}}$, and $s^\star_{\mathrm{C,A}}$ defined as in (5) and (6) depend on ϵ and β, while $s^\star_{\mathrm{D,SGD}}$, $s^\star_{\mathrm{D,NM}}$, and $s^\star_{\mathrm{D,A}}$ do not depend on ϵ and β.

[Comparison of minimum numbers of steps] Again, by a similar argument to (7) in Section 3.1.1, the restrictions on β (see (27), (28), (29), (30), (31), and (32) in Appendix) imply that

$$K_\epsilon(s^\star_{\mathrm{D,A}}) \leq K_\epsilon(s^\star_{\mathrm{D,NM}}) \leq K_\epsilon(s^\star_{\mathrm{D,SGD}}). \tag{9}$$

The previous studies (Kingma & Ba, 2015; Reddi et al., 2018; Luo et al., 2019) used $\beta = 0.9$ or 0.99, which is close to 1, for adaptive methods. Meanwhile, a sufficient condition (see (31)) for $K_\epsilon(s^\star_{\mathrm{D,A}}) \leq K_\epsilon(s^\star_{\mathrm{D,NM}})$ with $D = D_{\mathrm{NM}} = D_{\mathrm{A}}$ and $G = Ln\sqrt{dD}$ is $\beta \leq (1-b)\epsilon^2/(LndD)$, which implies that adaptive methods using the above β (which is small when the number of samples n and the number of dimension d are both large and the precision accuracy ϵ is small) are good for training deep neural networks in the sense that $K_\epsilon(s^\star_{\mathrm{D,A}}) \leq K_\epsilon(s^\star_{\mathrm{D,NM}})$ (see Sections A.8 and A.9 in Appendix for how to set β and its advantage).

4 NUMERICAL COMPARISONS AND DISCUSSIONS

Table 3: Number of steps, elapsed time, and training accuracy of optimizers when $f(\boldsymbol{x}_K) \leq 10^{-1}$ to train ResNet-20 on CIFAR-10

				SGD						
Batch Size	2^6	2^7	$\mathbf{2^8}$	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}
Steps	537500	287500	142500	146875	—	—	—	—	—	—
Time (m)	34.2	20.8	15.4	14.5	—	—	—	—	—	—
Acc. (%)	96.6	96.8	96.6	96.7	—	—	—	—	—	—

				N-Momentum						
Batch Size	2^6	2^7	$\mathbf{2^8}$	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}
Steps	392187	27734	12402	19531	—	—	—	—	—	—
Time (m)	38.2	21.7	14.3	12.6	—	—	—	—	—	—
Acc. (%)	96.5	96.7	96.7	96.7	—	—	—	—	—	—

					M-Adam					
Batch Size	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	$\mathbf{2^{13}}$	2^{14}	2^{15}
Steps	33593	15234	7226	3125	1367	659	378	320	323	337
Time (m)	14.2	11.3	7.2	6.7	6.4	6.4	6.6	6.6	6.4	6.6
Acc. (%)	96.4	96.6	96.5	96.7	96.7	97.2	97.5	97.7	97.5	99.0

We evaluated the performances of SGD, N-Momentum, and M-Adam with different batch sizes to train ResNet-20 on the CIFAR-10 dataset with $n = 50000$. We set $\alpha = 10^{-3}$, $\beta = 10^{-2}$, $\gamma = 0.9$, $h_0^* = 10^{-2}$, and $L = 10$. $H = 10$ were set so as to satisfy $\tilde{\gamma}\sqrt{Hh_0^*} < 1$, i.e., $s^\star_{\mathrm{D,SGD}} = s^\star_{\mathrm{D,NM}} < s^\star_{\mathrm{D,A}}$. Table 2 confirms that the optimal batch sizes of the optimizers are such that $s^\star_{\mathrm{D,SGD}} = s^\star_{\mathrm{D,NM}} = \sqrt{2}Ln\alpha \approx 2^8 < 2^{13} \approx \sqrt{2}Ln\alpha/(\tilde{\gamma}\sqrt{Hh_0^*}) = s^\star_{\mathrm{D,A}}$. Table 3 shows that the optimizers with $s^\star$ (indicated by bold type) could reduce the number of steps more than the ones with other batch sizes, (9) was satisfied, and Adam with $s^\star$ performed better than other optimizers. We also checked that SGD and N-Momentum with $s \geq 2^{10}$ do not satisfy $f(\boldsymbol{x}_K) \leq 10^{-1}$ until the stopping condition, namely, that the number of epochs is 200 and that the value of f of all of the optimizers is decreasing stably, i.e., the norm of the gradient of f converges to zero.

Finally, we check whether the batch sizes shown in (Shallue et al., 2019) are approximately the same as the optimal batch sizes. For training CNN on the MNIST dataset, SGD exploited the batch size between 2^{12} and 2^{14} and N-Momentum exploited the batch size between 2^{13} and 2^{14} (Shallue et al., 2019, Figure 4(a)). Meanwhile, when $n = 55000$, $\alpha = 10^{-3}$, and $L \approx 174$ (Virmaux & Scaman, 2018, Figure 5), the optimal batch sizes are such that $s^\star_{\mathrm{D,SGD}} = s^\star_{\mathrm{D,NM}} = \sqrt{2}Ln\alpha \approx 13533 \approx 2^{14}$. Hence, the optimal batch sizes of SGD and N-Momentum are almost the same as the ones in (Shallue et al., 2019, Figure 4(a)).

5 CONCLUSION

The main contribution of this paper was to show that the number of steps $K_\epsilon(s)$ needed for nonconvex optimization, $\min_{k\in[K]} \mathbb{E}[\|\nabla f(\boldsymbol{x}_k)\|^2] \leq \epsilon^2$, of a deep learning optimizer is a rational function of batch size. We showed that there exists an optimal batch size $s^\star$ such that $K_\epsilon(s)$ is minimized. Hence, there is no guarantee that the optimizer with a sufficiently large batch size s ($> s^\star$) would perform better than with a smaller batch size. We also showed that the optimal batch size depends on the optimizer. In particular, it was shown that momentum and adaptive methods can exploit the same sized or larger optimal batches than can SGD and that, if we can set an appropriate momentum coefficient β, then momentum and adaptive methods reduce $K_\epsilon(s^\star)$ more than can SGD. Additionally, numerical results were provided to support the theoretical results in this paper.

ACKNOWLEDGMENTS

ETHICS STATEMENT

Ethics approval was not required for this study.

REPRODUCIBILITY STATEMENT

The experimental environment is as follows: two Intel(R) Xeon(R) Gold 6148 at 2.4 GHz CPUs with 20 cores, 16 GB NVIDIA Tesla V100 at 900 Gbps GPU, Red Hat Enterprise Linux 7.6. The code was all written in Python 3.8.2 using the NumPy 1.17.3 and PyTorch 1.3.0 packages. Sufficient conditions for Assumption (A3) and complete proofs of the theoretical results (Theorems 3.1 and 3.2) were included as Appendix.

REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. `https://arxiv.org/pdf/1701.07875.pdf`, 2017.

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

Hao Chen, Lili Zheng, Raed AL Kontar, and Garvesh Raskutti. Stochastic gradient descent in correlated settings: A study on Gaussian processes. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of Adam-type algorithms for non-convex optimization. In *Proceedings of The International Conference on Learning Representations*, 2019.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Benjamin Fehrman, Benjamin Gess, and Arnulf Jentzen. Convergence rates for the stochastic gradient descent method for non-convex objective functions. *Journal of Machine Learning Research*, 21:1–48, 2020.

Saeed Ghadimi and Guanghui Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization I: A generic algorithmic framework. *SIAM Journal on Optimization*, 22:1469–1492, 2012.

Saeed Ghadimi and Guanghui Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization II: Shrinking procedures and optimal algorithms. *SIAM Journal on Optimization*, 23:2061–2089, 2013.

Robert M. Gower, Othmane Sebbouh, and Nicolas Loizou. SGD for structured nonconvex functions: Learning rates, minibatching and interpolation. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130, 2021.

Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 1985.

Hideaki Iiduka. Appropriate learning rates of adaptive learning rate optimization algorithms for training deep neural networks. *IEEE Transactions on Cybernetics*, DOI: 10.1109/TCYB.2021.3107415, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of The International Conference on Learning Representations*, 2015.

Nicolas Loizou, Sharan Vaswani, Issam Laradji, and Simon Lacoste-Julien. Stochastic polyak step-size for SGD: An adaptive learning rate for fast convergence. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130, 2021.

Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. In *Proceedings of The International Conference on Learning Representations*, 2019.

James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *Proceedings of Machine Learning Research*, volume 37, pp. 2408–2417, 2015.

Celestine Mendler-Dünner, Juan C. Perdomo, Tijana Zrnic, and Moritz Hardt. Stochastic optimization for performative prediction. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

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

Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. *Doklady AN USSR*, 269:543–547, 1983.

Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4:1–17, 1964.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of Adam and beyond. In *Proceedings of The International Conference on Learning Representations*, 2018.

Herbert Robbins and Herbert Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22:400–407, 1951.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

Kevin Scaman and Cédric Malherbe. Robustness analysis of non-convex stochastic gradient descent using biased expectations. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Robin M. Schmidt, Frank Schneider, and Philipp Hennig. Descending through a crowded valley–Benchmarking deep learning optimizers. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 9367–9376, 2021.

Christopher J. Shallue, Jaehoon Lee, Joseph Antognini, Jascha Sohl-Dickstein, Roy Frostig, and George E. Dahl. Measuring the effects of data parallelism on neural network training. *Journal of Machine Learning Research*, 20:1–49, 2019.

Samuel L. Smith, Pieter-Jan Kindermans, and Quoc V. Le. Don't decay the learning rate, increase the batch size. In *Proceedings of The International Conference on Learning Representations*, 2018.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 1139–1147, 2013.

Tijmen Tieleman and Geoffrey Hinton. RMSProp: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4:26–31, 2012.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pp. 2048–2057, 2015.

Guodong Zhang, Lala Li, Zachary Nado, James Martens, Sushant Sachdeva, George E. Dahl, Christopher J. Shallue, and Roger Grosse. Which algorithmic choices matter at which batch sizes? Insights from a noisy quadratic model. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Juntang Zhuang, Tommy Tang, Yifan Ding, Sekhar Tatikonda, Nicha Dvornek, Xenophon Papademetris, and James S. Duncan. AdaBelief optimizer: Adapting stepsizes by the belief in observed gradients. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning*, pp. 928–936, 2003.

Martin Zinkevich, Markus Weimer, Lihong Li, and Alex Smola. Parallelized stochastic gradient descent. In *Advances in Neural Information Processing Systems*, volume 23, 2010.

A Appendix

Unless stated otherwise, all relations between random variables are supported to hold almost surely. Let $S \in \mathbb{S}^d_{++}$. The S-inner product of $\mathbb{R}^d$ is defined for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ by $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_S := \langle \boldsymbol{x}, S\boldsymbol{y} \rangle$ and the S-norm is defined by $\|\boldsymbol{x}\|_S := \sqrt{\langle \boldsymbol{x}, S\boldsymbol{x} \rangle}$. The history of process $\xi_0, \xi_1, \ldots$ to time step k is denoted by $\xi_{[k]} = (\xi_0, \xi_1, \ldots, \xi_k)$.

A.1 Sufficient Conditions for Assumption (A3)

Proposition A.1 *Assumption (A3) holds if each of the following holds:*

(G1) *$X \subset \mathbb{R}^d$ is bounded, the gradient ∇f_i is Lipschitz continuous with Lipschitz constant L_i, and $S_i := \{\boldsymbol{x}^* \in \mathbb{R}^d \colon \nabla f_i(\boldsymbol{x}^*) = \boldsymbol{0}\} \neq \emptyset$ $(i \in [n])$, where $L := \max_{i \in [n]} L_i$. (If we define $G_{k,L} := \sup_{\boldsymbol{x} \in X} \sum_{i \in \mathcal{S}_k} \|\nabla f_i(\boldsymbol{x})\|$, then we can take $G := \sup_{k \in \mathbb{N}} G_{k,L}$.)*

(G2) *$X \subset \mathbb{R}^d$ is bounded and closed. (If we define $G_k := \sup_{\boldsymbol{x} \in X} \sum_{i \in \mathcal{S}_k} \|\nabla f_i(\boldsymbol{x})\|$, then we can take $G := \sup_{k \in \mathbb{N}} G_k$.)*

Under (A6), G in (G1) and (G2) are respectively $G = Ln\sqrt{dD}$ and $G = n\tilde{G}$, where $\tilde{G} := \max_{i \in [n]} \sup_{\boldsymbol{x} \in X} \|\nabla f_i(\boldsymbol{x})\|$.

Proof: The definition of $\nabla f_{\mathcal{S}_k}$ and the triangle inequality imply that, for all $\boldsymbol{x} \in \mathbb{R}^d$ and all $k \in \mathbb{N}$,

$$\|\nabla f_{\mathcal{S}_k}(\boldsymbol{x})\|^2 = \left\| \frac{1}{s} \sum_{i \in \mathcal{S}_k} \nabla f_i(\boldsymbol{x}) \right\|^2 \leq \frac{1}{s^2} \left(\sum_{i \in \mathcal{S}_k} \|\nabla f_i(\boldsymbol{x})\| \right)^2 \leq \left(\sum_{i \in \mathcal{S}_k} \|\nabla f_i(\boldsymbol{x})\| \right)^2, \tag{10}$$

where the final inequality comes from $s \geq 1$. Suppose that (G1) holds. Let $\boldsymbol{x}^* \in S_i$ $(i \in [n])$. The Lipschitz continuity of ∇f_i, together with the definition of L, ensures that, for all $\boldsymbol{x} \in \mathbb{R}^d$ and all $i \in [n]$,

$$\|\nabla f_i(\boldsymbol{x})\| = \|\nabla f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x}^*)\| \leq L_i \|\boldsymbol{x} - \boldsymbol{x}^*\| \leq L \|\boldsymbol{x} - \boldsymbol{x}^*\|.$$

Accordingly, we have that, for all $\boldsymbol{x} \in X$ and all $k \in \mathbb{N}$,

$$\sum_{i \in \mathcal{S}_k} \|\nabla f_i(\boldsymbol{x})\| \leq Ls \|\boldsymbol{x} - \boldsymbol{x}^*\| \leq Ln \|\boldsymbol{x} - \boldsymbol{x}^*\|. \tag{11}$$

Hence, there exists $G > 0$ such that $G_{k,L} \leq Ln \sup_{\boldsymbol{x} \in X} \|\boldsymbol{x} - \boldsymbol{x}^*\| \leq G$. Taking the expectation of (10), together with (11), thus implies that

$$\mathbb{E}[\|\nabla f_{\mathcal{S}_k}(\boldsymbol{x})\|^2] \leq \mathbb{E}\left[\left(\sum_{i \in \mathcal{S}_k} \|\nabla f_i(\boldsymbol{x})\|\right)^2\right] \leq (Ln\|\boldsymbol{x} - \boldsymbol{x}^*\|)^2 \leq G^2.$$

Assumption (A6) implies that there exists a bounded set $X \subset \mathbb{R}^d$ such that $(\boldsymbol{x}_k)_{k \in \mathbb{N}} \subset X$. From $\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2 = \sum_{i \in [d]} (x_{k,i} - x_i)^2 \leq dD$, we have that, for all $k \in \mathbb{N}$,

$$G_{k,L} \leq Ln\sqrt{dD} =: G.$$

Suppose that (G2) holds. Since ∇f_i is continuous and X is compact, we have that $G = \sup_{k \in \mathbb{N}} G_k < +\infty$. Taking the expectation of (10) thus implies (A3). Assumption (A6) ensures that there exists a bounded, closed set $X \subset \mathbb{R}^d$ such that $(\boldsymbol{x}_k)_{k \in \mathbb{N}} \subset X$. Define $\tilde{G}_i := \sup_{\boldsymbol{x} \in X} \|\nabla f_i(\boldsymbol{x})\| < +\infty$ and $\tilde{G} := \max_{i \in [n]} \tilde{G}_i$. Then, we have that, for all $\boldsymbol{x} \in X$,

$$\sum_{i \in \mathcal{S}_k} \|\nabla f_i(\boldsymbol{x})\| \leq s\tilde{G} \leq n\tilde{G} =: G.$$

This completes the proof. □

A.2 Examples of Algorithm 1

We list some examples of $\mathsf{H}_k \in \mathbb{S}^d_{++} \cap \mathbb{D}^d$ (step 5) in Algorithm 1.

Table 4: Examples of $\mathsf{H}_k \in \mathbb{S}^d_{++} \cap \mathbb{D}^d$ (step 5) in Algorithm 1 ($\delta, \zeta \in [0, 1)$)

	H_k
SGD ($\beta_k = \gamma = 0$)	H_k is the identity matrix.
N-Momentum (Nesterov, 1983) ($\gamma = 0$)	H_k is the identity matrix.
AMSGrad (Reddi et al., 2018; Chen et al., 2019) ($\gamma = 0$)	$\boldsymbol{v}_k = \delta \boldsymbol{v}_{k-1} + (1-\delta)\nabla f_{\mathcal{S}_k}(\boldsymbol{x}_k) \odot \nabla f_{\mathcal{S}_k}(\boldsymbol{x}_k)$ $\hat{\boldsymbol{v}}_k = (\max\{\hat{v}_{k-1,i}, v_{k,i}\})_{i=1}^d$ $\mathsf{H}_k = \mathsf{diag}(\sqrt{\hat{v}_{k,i}})$
AMSBound (Luo et al., 2019) ($\gamma = 0$)	$\boldsymbol{v}_k = \delta \boldsymbol{v}_{k-1} + (1-\delta)\nabla f_{\mathcal{S}_k}(\boldsymbol{x}_k) \odot \nabla f_{\mathcal{S}_k}(\boldsymbol{x}_k)$ $\hat{\boldsymbol{v}}_k = (\max\{\hat{v}_{k-1,i}, v_{k,i}\})_{i=1}^d$ $\tilde{\boldsymbol{v}}_k = \left(\mathrm{Clip}\left(\frac{1}{\sqrt{\hat{v}_{k,i}}}, l_k, u_k\right)^{-1}\right)_{i=1}^d$ $\mathsf{H}_k = \mathsf{diag}(\tilde{v}_{k,i})$
M-Adam (Kingma & Ba, 2015; Iiduka, 2021)	$\boldsymbol{v}_k = \delta \boldsymbol{v}_{k-1} + (1-\delta)\nabla f_{\mathcal{S}_k}(\boldsymbol{x}_k) \odot \nabla f_{\mathcal{S}_k}(\boldsymbol{x}_k)$ $\bar{\boldsymbol{v}}_k = \frac{\boldsymbol{v}_k}{1-\zeta^k}$ $\hat{\boldsymbol{v}}_k = (\max\{\hat{v}_{k-1,i}, \bar{v}_{k,i}\})_{i=1}^d$ $\mathsf{H}_k = \mathsf{diag}(\sqrt{\hat{v}_{k,i}})$
AdaBelief (Zhuang et al., 2020) ($s_{k,i} \leq s_{k+1,i}$ is needed)	$\tilde{\boldsymbol{s}}_k = (\nabla f_{\mathcal{S}_k}(\boldsymbol{x}_k) - \boldsymbol{m}_k) \odot (\nabla f_{\mathcal{S}_k}(\boldsymbol{x}_k) - \boldsymbol{m}_k)$ $\boldsymbol{s}_k = \delta \boldsymbol{v}_{k-1} + (1-\delta)\tilde{\boldsymbol{s}}_k$ $\hat{\boldsymbol{s}}_k = \frac{\boldsymbol{s}_k}{1-\zeta^k}$ $\mathsf{H}_k = \mathsf{diag}(\sqrt{\hat{s}_{k,i}})$

We define $\boldsymbol{x} \odot \boldsymbol{x}$ for $\boldsymbol{x} := (x_i)_{i=1}^d \in \mathbb{R}^d$ by $\boldsymbol{x} \odot \boldsymbol{x} := (x_i^2)_{i=1}^d \in \mathbb{R}^d$. $\mathrm{Clip}(\cdot, l, u) \colon \mathbb{R} \to \mathbb{R}$ in AMSBound ($l, u \in \mathbb{R}$ with $l \leq u$ are given) is defined for all $x \in \mathbb{R}$ by

$$\mathrm{Clip}(x, l, u) := \begin{cases} l & \text{if } x < l, \\ x & \text{if } l \leq x \leq u, \\ u & \text{if } x > u. \end{cases}$$

While Adam (Kingma & Ba, 2015) uses $\mathsf{H}_k = \mathsf{diag}(\sqrt{\hat{v}_{k,i}})$, M-Adam (Iiduka, 2021) uses $\mathsf{H}_k = \mathsf{diag}(\sqrt{\hat{v}_{k,i}})$ to satisfy (A4) (see also Theorem 1 in (Reddi et al., 2018) indicating that Adam does not always converge).

A.3 LEMMAS AND THEOREM

The following are the key lemmas to prove the main theorems in this paper.

Lemma A.1 *Suppose that (A1) and (A2) hold and consider Algorithm 1. Then, for all $\boldsymbol{x} \in \mathbb{R}^d$ and all $k \in \mathbb{N}$,*

$$\mathbb{E}\left[\|\boldsymbol{x}_{k+1} - \boldsymbol{x}\|_{\mathsf{H}_k}^2\right] \leq \mathbb{E}\left[\|\boldsymbol{x}_k - \boldsymbol{x}\|_{\mathsf{H}_k}^2\right] + \alpha_k^2 \mathbb{E}\left[\|\mathbf{d}_k\|_{\mathsf{H}_k}^2\right] + 2\alpha_k \left\{\frac{\tilde{\beta}_k}{\tilde{\gamma}_k}\mathbb{E}\left[\langle \boldsymbol{x} - \boldsymbol{x}_k, \nabla f(\boldsymbol{x}_k)\rangle\right] + \frac{\beta_k}{\tilde{\gamma}_k}\mathbb{E}\left[\langle \boldsymbol{x} - \boldsymbol{x}_k, \boldsymbol{m}_{k-1}\rangle\right]\right\},$$

where $\tilde{\beta}_k := 1 - \beta_k$ and $\tilde{\gamma}_k := 1 - \gamma^{k+1}$.

Proof: Let $\boldsymbol{x} \in \mathbb{R}^d$ and $k \in \mathbb{N}$. The definition of $\boldsymbol{x}_{k+1}$ implies that

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}\|_{\mathsf{H}_k}^2 = \|\boldsymbol{x}_k - \boldsymbol{x}\|_{\mathsf{H}_k}^2 + 2\alpha_k \langle \boldsymbol{x}_k - \boldsymbol{x}, \mathbf{d}_k\rangle_{\mathsf{H}_k} + \alpha_k^2 \|\mathbf{d}_k\|_{\mathsf{H}_k}^2.$$

Moreover, the definitions of $\mathbf{d}_k$, $\boldsymbol{m}_k$, and $\hat{\boldsymbol{m}}_k$ ensure that

$$\langle \boldsymbol{x}_k - \boldsymbol{x}, \mathbf{d}_k\rangle_{\mathsf{H}_k} = \frac{1}{\tilde{\gamma}_k}\langle \boldsymbol{x} - \boldsymbol{x}_k, \boldsymbol{m}_k\rangle = \frac{\beta_k}{\tilde{\gamma}_k}\langle \boldsymbol{x} - \boldsymbol{x}_k, \boldsymbol{m}_{k-1}\rangle + \frac{\tilde{\beta}_k}{\tilde{\gamma}_k}\langle \boldsymbol{x} - \boldsymbol{x}_k, \nabla f_{\mathcal{S}_k}(\boldsymbol{x}_k)\rangle,$$

where $\tilde{\beta}_k := 1 - \beta_k$ and $\tilde{\gamma}_k := 1 - \gamma^{k+1}$. Hence,

$$\|\boldsymbol{x}_{k+1} - \boldsymbol{x}\|_{\mathsf{H}_k}^2 \leq \|\boldsymbol{x}_k - \boldsymbol{x}\|_{\mathsf{H}_k}^2 + 2\alpha_k \left\{\frac{\beta_k}{\tilde{\gamma}_k}\langle \boldsymbol{x} - \boldsymbol{x}_k, \boldsymbol{m}_{k-1}\rangle + \frac{\tilde{\beta}_k}{\tilde{\gamma}_k}\langle \boldsymbol{x} - \boldsymbol{x}_k, \nabla f_{\mathcal{S}_k}(\boldsymbol{x}_k)\rangle\right\} + \alpha_k^2 \|\mathbf{d}_k\|_{\mathsf{H}_k}^2. \tag{12}$$

Meanwhile, the relationship between the expectation of the stochastic gradient vector $\nabla f_{\mathcal{S}_k}(\boldsymbol{x})$ and the full gradient vector $\nabla f(\boldsymbol{x})$ is as follows: For all $\boldsymbol{x} \in \mathbb{R}^d$,

$$\mathbb{E}\left[\nabla f_{\mathcal{S}_k}(\boldsymbol{x})\right] = \mathbb{E}\left[\frac{1}{s}\sum_{i \in \mathcal{S}_k} \nabla f_i(\boldsymbol{x})\right] = \mathbb{E}\left[\nabla f_i(\boldsymbol{x})\right] = \nabla f(\boldsymbol{x}), \tag{13}$$

where the first equation comes from (A2), the second equation comes from the existence of T such that $[n] = \cup_{k=0}^{T-1}\mathcal{S}_k$, and the third equation comes from (A1). Condition (13) guarantees that

$$\begin{aligned}\mathbb{E}\left[\langle \boldsymbol{x} - \boldsymbol{x}_k, \nabla f_{\mathcal{S}_k}(\boldsymbol{x}_k)\rangle\right] &= \mathbb{E}\left[\mathbb{E}\left[\langle \boldsymbol{x} - \boldsymbol{x}_k, \nabla f_{\mathcal{S}_k}(\boldsymbol{x}_k)\rangle \,|\xi_{[k-1]}\right]\right] \\ &= \mathbb{E}\left[\left\langle \boldsymbol{x} - \boldsymbol{x}_k, \mathbb{E}\left[\nabla f_{\mathcal{S}_k}(\boldsymbol{x}_k)|\xi_{[k-1]}\right]\right\rangle\right] \\ &= \mathbb{E}\left[\langle \boldsymbol{x} - \boldsymbol{x}_k, \nabla f(\boldsymbol{x}_k)\rangle\right].\end{aligned}$$

Therefore, the lemma follows by taking the expectation of (12). □

Lemma A.2 *Algorithm 1 satisfies that, under (A3), for all $k \in \mathbb{N}$,*

$$\mathbb{E}\left[\|\boldsymbol{m}_k\|^2\right] \leq G^2.$$

Under (A3) and (A4), for all $k \in \mathbb{N}$,

$$\mathbb{E}\left[\|\mathbf{d}_k\|_{\mathsf{H}_k}^2\right] \leq \frac{G^2}{(1-\gamma)^2 h_0^*},$$

where $h_0^* := \min_{i \in [d]} h_{0,i}$.

Proof: The convexity of $\|\cdot\|^2$, together with the definition of $\boldsymbol{m}_k$ and (A3), guarantees that, for all $k \in \mathbb{N}$,

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{m}_k\|^2\right] &\leq \beta_k \mathbb{E}\left[\|\boldsymbol{m}_{k-1}\|^2\right] + (1-\beta_k)\mathbb{E}\left[\|\nabla f_{\mathcal{S}_k}(\boldsymbol{x}_k)\|^2\right] \\ &\leq \beta_k \mathbb{E}\left[\|\boldsymbol{m}_{k-1}\|^2\right] + (1-\beta_k)G^2.\end{aligned}$$

Induction thus ensures that, for all $k \in \mathbb{N}$,

$$\mathbb{E}\left[\|\boldsymbol{m}_k\|^2\right] \leq \max\left\{\|\boldsymbol{m}_{-1}\|^2, G^2\right\} = G^2, \tag{14}$$

where $\boldsymbol{m}_{-1} = \mathbf{0}$ is used. For $k \in \mathbb{N}$, $\mathsf{H}_k \in \mathbb{S}^d_{++}$ guarantees the existence of a unique matrix $\overline{\mathsf{H}}_k \in \mathbb{S}^d_{++}$ such that $\mathsf{H}_k = \overline{\mathsf{H}}_k^2$ (Horn & Johnson, 1985, Theorem 7.2.6). We have that, for all $\boldsymbol{x} \in \mathbb{R}^d$, $\|\boldsymbol{x}\|^2_{\mathsf{H}_k} = \|\overline{\mathsf{H}}_k \boldsymbol{x}\|^2$. Accordingly, the definitions of $\mathbf{d}_k$ and $\hat{\boldsymbol{m}}_k$ imply that, for all $k \in \mathbb{N}$,

$$\mathbb{E}\left[\|\mathbf{d}_k\|^2_{\mathsf{H}_k}\right] = \mathbb{E}\left[\left\|\overline{\mathsf{H}}_k^{-1}\mathsf{H}_k\mathbf{d}_k\right\|^2\right] \leq \frac{1}{\tilde{\gamma}_k^2}\mathbb{E}\left[\left\|\overline{\mathsf{H}}_k^{-1}\right\|^2 \|\boldsymbol{m}_k\|^2\right] \leq \frac{1}{(1-\gamma)^2}\mathbb{E}\left[\left\|\overline{\mathsf{H}}_k^{-1}\right\|^2 \|\boldsymbol{m}_k\|^2\right],$$

where

$$\left\|\overline{\mathsf{H}}_k^{-1}\right\| = \left\|\mathsf{diag}\left(h_{k,i}^{-\frac{1}{2}}\right)\right\| = \max_{i\in[d]} h_{k,i}^{-\frac{1}{2}}$$

and $\tilde{\gamma}_k := 1 - \gamma^{k+1} \geq 1 - \gamma$. Moreover, (A4) ensures that, for all $k \in \mathbb{N}$,

$$h_{k,i} \geq h_{0,i} \geq h_0^* := \min_{i\in[d]} h_{0,i}.$$

Hence, (14) implies that, for all $k \in \mathbb{N}$,

$$\mathbb{E}\left[\|\mathbf{d}_k\|^2_{\mathsf{H}_k}\right] \leq \frac{G^2}{(1-\gamma)^2 h_0^*},$$

completing the proof. □

We are in the position to prove the following theorem, which leads to Theorems 3.1, 3.2, and A.2.

Theorem A.1 *Suppose that Assumptions 2.1 and 2.2 hold and consider Algorithm 1. Let $(\delta_k)_{k\in\mathbb{N}} \subset (0, +\infty)$ be the sequence defined by $\delta_k := \alpha_k\tilde{\beta}_k/\tilde{\gamma}_k$ and $V_k(\boldsymbol{x}) := \mathbb{E}[\langle \boldsymbol{x}_k - \boldsymbol{x}, \nabla f(\boldsymbol{x}_k)\rangle]$ for all $\boldsymbol{x} \in \mathbb{R}^d$ and all $k \in \mathbb{N}$. Assume that $(\delta_k)_{k\in\mathbb{N}}$ is monotone decreasing. Then, for all $\boldsymbol{x} \in \mathbb{R}^d$ and all $K \geq 1$,*

$$\sum_{k=1}^K V_k(\boldsymbol{x}) \leq \frac{dDH}{2\tilde{b}\alpha_K} + \frac{G^2}{2\tilde{b}\tilde{\gamma}^2 h_0^*}\sum_{k=1}^K \alpha_k + \frac{\sqrt{d}DG}{\tilde{b}}\sum_{k=1}^K \beta_k,$$

where $\tilde{b} := 1 - b$, $\tilde{\gamma} := 1 - \gamma$, D and H_i are defined as in Assumption 2.2, and $H := \max_{i\in[d]} H_i$.

Proof: Let $\boldsymbol{x} \in \mathbb{R}^d$. Lemma A.1 guarantees that, for all $k \in \mathbb{N}$,

$$\begin{aligned} V_k(\boldsymbol{x}) \leq{}& \frac{1}{2\delta_k}\left\{\mathbb{E}\left[\|\boldsymbol{x}_k - \boldsymbol{x}\|^2_{\mathsf{H}_k}\right] - \mathbb{E}\left[\|\boldsymbol{x}_{k+1} - \boldsymbol{x}\|^2_{\mathsf{H}_k}\right]\right\} + \frac{\alpha_k\tilde{\gamma}_k}{2\tilde{\beta}_k}\mathbb{E}\left[\|\mathbf{d}_k\|^2_{\mathsf{H}_k}\right] \\ &+ \frac{\beta_k}{\tilde{\beta}_k}\mathbb{E}\left[\langle \boldsymbol{x} - \boldsymbol{x}_k, \boldsymbol{m}_{k-1}\rangle\right]. \end{aligned}$$

Summing the above inequality from $k = 1$ to $K \geq 1$ implies that

$$\begin{aligned} \sum_{k=1}^K V_k(\boldsymbol{x}) \leq{}& \underbrace{\frac{1}{2}\sum_{k=1}^K \frac{1}{\delta_k}\left\{\mathbb{E}\left[\|\boldsymbol{x}_k - \boldsymbol{x}\|^2_{\mathsf{H}_k}\right] - \mathbb{E}\left[\|\boldsymbol{x}_{k+1} - \boldsymbol{x}\|^2_{\mathsf{H}_k}\right]\right\}}_{\Delta_K} \\ &+ \underbrace{\frac{1}{2}\sum_{k=1}^K \frac{\alpha_k\tilde{\gamma}_k}{\tilde{\beta}_k}\mathbb{E}\left[\|\mathbf{d}_k\|^2_{\mathsf{H}_k}\right]}_{A_K} + \underbrace{\sum_{k=1}^K \frac{\beta_k}{\tilde{\beta}_k}\mathbb{E}\left[\langle \boldsymbol{x} - \boldsymbol{x}_k, \boldsymbol{m}_{k-1}\rangle\right]}_{B_K}. \end{aligned} \tag{15}$$

From the definition of Δ_K and $\mathbb{E}[\|\boldsymbol{x}_{K+1} - \boldsymbol{x}\|^2_{\mathsf{H}_K}]/\delta_K \geq 0$,

$$\Delta_K \leq \frac{\mathbb{E}\left[\|\boldsymbol{x}_1 - \boldsymbol{x}\|^2_{\mathsf{H}_1}\right]}{\delta_1} + \underbrace{\sum_{k=2}^K \left\{\frac{\mathbb{E}\left[\|\boldsymbol{x}_k - \boldsymbol{x}\|^2_{\mathsf{H}_k}\right]}{\delta_k} - \frac{\mathbb{E}\left[\|\boldsymbol{x}_k - \boldsymbol{x}\|^2_{\mathsf{H}_{k-1}}\right]}{\delta_{k-1}}\right\}}_{\tilde{\Delta}_K}. \tag{16}$$

Since $\overline{\mathsf{H}}_k \in \mathbb{S}^d_{++}$ exists such that $\mathsf{H}_k = \overline{\mathsf{H}}_k^2$, we have $\|\boldsymbol{x}\|^2_{\mathsf{H}_k} = \|\overline{\mathsf{H}}_k \boldsymbol{x}\|^2$ for all $\boldsymbol{x} \in \mathbb{R}^d$. Accordingly, we have

$$\tilde{\Delta}_K = \mathbb{E}\left[\sum_{k=2}^{K}\left\{\frac{\left\|\overline{\mathsf{H}}_k(\boldsymbol{x}_k - \boldsymbol{x})\right\|^2}{\delta_k} - \frac{\left\|\overline{\mathsf{H}}_{k-1}(\boldsymbol{x}_k - \boldsymbol{x})\right\|^2}{\delta_{k-1}}\right\}\right].$$

From $\overline{\mathsf{H}}_k = \mathsf{diag}(\sqrt{h_{k,i}})$, we have that, for all $\boldsymbol{x} = (x_i)_{i=1}^d \in \mathbb{R}^d$, $\|\overline{\mathsf{H}}_k \boldsymbol{x}\|^2 = \sum_{i=1}^d h_{k,i} x_i^2$. Hence, for all $K \geq 2$,

$$\tilde{\Delta}_K = \mathbb{E}\left[\sum_{k=2}^{K}\sum_{i=1}^{d}\left(\frac{h_{k,i}}{\delta_k} - \frac{h_{k-1,i}}{\delta_{k-1}}\right)(x_{k,i} - x_i)^2\right].$$

Accordingly, from (A4) and the monotone decrease of $(\delta_k)_{k\in\mathbb{N}}$, we have that, for all $k \geq 1$ and all $i \in [d]$,

$$\frac{h_{k,i}}{\delta_k} - \frac{h_{k-1,i}}{\delta_{k-1}} \geq 0.$$

Moreover, from (A6), $D := \max_{i\in[d]} \sup_{k\in\mathbb{N}} (x_{k,i} - x_i)^2 < +\infty$. Accordingly, for all $K \geq 2$,

$$\tilde{\Delta}_K \leq D\mathbb{E}\left[\sum_{k=2}^{K}\sum_{i=1}^{d}\left(\frac{h_{k,i}}{\delta_k} - \frac{h_{k-1,i}}{\delta_{k-1}}\right)\right] = D\mathbb{E}\left[\sum_{i=1}^{d}\left(\frac{h_{K,i}}{\delta_K} - \frac{h_{1,i}}{\delta_1}\right)\right].$$

Therefore, (16), $\mathbb{E}[\|\boldsymbol{x}_1 - \boldsymbol{x}\|^2_{\mathsf{H}_1}]/\delta_1 \leq D\mathbb{E}[\sum_{i=1}^d h_{1,i}/\delta_1]$, and (A5) imply, for all $K \geq 1$,

$$\Delta_K \leq D\mathbb{E}\left[\sum_{i=1}^{d}\frac{h_{1,i}}{\delta_1}\right] + D\mathbb{E}\left[\sum_{i=1}^{d}\left(\frac{h_{K,i}}{\delta_K} - \frac{h_{1,i}}{\delta_1}\right)\right] = \frac{D}{\delta_K}\mathbb{E}\left[\sum_{i=1}^{d} h_{K,i}\right] \leq \frac{D}{\delta_K}\sum_{i=1}^{d} H_i,$$

which, together with $\delta_K := \alpha_K(1-\beta_K)/(1-\gamma^{K+1}) \geq \tilde{b}\alpha_K$ and $H = \max_{i\in[d]} H_i$, implies

$$\frac{1}{2}\Delta_K \leq \frac{dDH}{2\tilde{b}\alpha_K}. \tag{17}$$

Lemma A.2 implies that, for all $K \geq 1$,

$$A_K := \sum_{k=1}^{K}\frac{\alpha_k\tilde{\gamma}_k}{\tilde{\beta}_k}\mathbb{E}\left[\|\mathbf{d}_k\|^2_{\mathsf{H}_k}\right] \leq \sum_{k=1}^{K}\frac{\alpha_k\tilde{\gamma}_k}{\tilde{\beta}_k}\frac{G^2}{\tilde{\gamma}^2 h_0^*},$$

which, together with $\tilde{\gamma}_k \leq 1$ and $\beta_k \leq b$, implies that

$$\frac{1}{2}A_K \leq \frac{G^2}{2\tilde{b}\tilde{\gamma}^2 h_0^*}\sum_{k=1}^{K}\alpha_k. \tag{18}$$

Lemma A.2 and Jensen's inequality ensure that, for all $k \in \mathbb{N}$,

$$\mathbb{E}\left[\|\boldsymbol{m}_k\|\right] \leq G.$$

The Cauchy-Schwarz inequality and (A6) guarantee that, for all $K \geq 1$,

$$B_K := \sum_{k=1}^{K}\frac{\beta_k}{\tilde{\beta}_k}\mathbb{E}\left[\langle \boldsymbol{x} - \boldsymbol{x}_k, \boldsymbol{m}_{k-1}\rangle\right] \leq \sum_{k=1}^{K}\frac{\sqrt{dD}\beta_k}{\tilde{b}}\mathbb{E}\left[\|\boldsymbol{m}_{k-1}\|\right] \leq \frac{\sqrt{dD}G}{\tilde{b}}\sum_{k=1}^{K}\beta_k. \tag{19}$$

Therefore, (15), (17), (18), and (19) lead to the assertion in Theorem A.1. This completes the proof. □

A.4 PROOF OF THEOREM 3.1

(i) Theorem A.1, together with $\alpha_k = \alpha/s$ and $\beta_k = \beta$, guarantees that, for all $K \geq 1$, all $s > 0$, and all $\boldsymbol{x} \in \mathbb{R}^d$,

$$\frac{1}{K}\sum_{k=1}^{K} V_k(\boldsymbol{x}) \leq \frac{dDH}{2\tilde{b}\alpha}\frac{s}{K} + \frac{G^2\alpha}{2\tilde{b}\tilde{\gamma}^2 h_0^*}\frac{1}{s} + \frac{\sqrt{dD}G}{\tilde{b}}\beta. \tag{20}$$

Moreover, there exists $m \in [K]$ such that, for all $\boldsymbol{x} \in \mathbb{R}^d$,

$$\mathbb{E}\left[\langle \boldsymbol{x}_m - \boldsymbol{x}, \nabla f(\boldsymbol{x}_m)\rangle\right] = V_m(\boldsymbol{x}) = \min_{k\in[K]} V_k(\boldsymbol{x}) \leq \frac{1}{K}\sum_{k=1}^{K} V_k(\boldsymbol{x}). \tag{21}$$

Setting $\boldsymbol{x} = \boldsymbol{x}_m - \nabla f(\boldsymbol{x}_m)$, together with (20) and (21), guarantees that

$$\min_{k\in[K]} \mathbb{E}\left[\|\nabla f(\boldsymbol{x}_k)\|^2\right] \leq \mathbb{E}\left[\|\nabla f(\boldsymbol{x}_m)\|^2\right] \leq \underbrace{\frac{dDH}{2\tilde{b}\alpha}}_{A_\alpha}\frac{s}{K} + \underbrace{\frac{G^2\alpha}{2\tilde{b}\tilde{\gamma}^2 h_0^*}}_{B_\alpha}\frac{1}{s} + \underbrace{\frac{\sqrt{dD}G}{\tilde{b}}}_{C_\beta}\beta. \tag{22}$$

(ii) A sufficient condition for (1), i.e.,

$$\min_{k\in[K]} \mathbb{E}\left[\|\nabla f(\boldsymbol{x}_k)\|^2\right] \leq \epsilon^2$$

is that the right-hand side of (22) is equal to ϵ^2, i.e.,

$$A_\alpha s^2 + B_\alpha K + (C_\beta - \epsilon^2)sK = 0,$$

which implies that

$$K(s) = \frac{A_\alpha s^2}{(\epsilon^2 - C_\beta)s - B_\alpha} \qquad \left(s \in \left(\frac{B_\alpha}{\epsilon^2 - C_\beta}, +\infty\right)\right),$$

where $\epsilon^2 - C_\beta > 0$ is guaranteed from $\beta < \tilde{b}\epsilon^2/\sqrt{dD}G$. We have that

$$\frac{\mathrm{d}K(s)}{\mathrm{d}s} = \frac{A_\alpha s}{\{(C_\beta - \epsilon^2)s + B_\alpha\}^2}\left\{(\epsilon^2 - C_\beta)s - 2B_\alpha\right\}\begin{cases} < 0 & \text{if } s \in \left(\dfrac{B_\alpha}{\epsilon^2 - C_\beta}, s^\star\right), \\ = 0 & \text{if } s = s^\star = \dfrac{2B_\alpha}{\epsilon^2 - C_\beta}, \\ > 0 & \text{if } s \in (s^\star, +\infty). \end{cases}$$

Hence, $K(s)$ attains the minimum $K(s^\star)$ when $s = s^\star$. □

A.5 PROOF OF THEOREM 3.2

(i) Theorem A.1, together with $\alpha_k = \alpha/(s\sqrt{k})$ and $\beta_k = \beta$, guarantees that, for all $K \geq 1$, all $s > 0$, and all $\boldsymbol{x} \in \mathbb{R}^d$,

$$\begin{aligned}\frac{1}{K}\sum_{k=1}^{K} V_k(\boldsymbol{x}) &\leq \frac{dDH}{2\tilde{b}}\frac{1}{\alpha_K K} + \frac{G^2}{2\tilde{b}\tilde{\gamma}^2 h_0^*}\frac{1}{K}\sum_{k=1}^{K}\alpha_k + \frac{\sqrt{dD}G}{\tilde{b}}\beta \\ &\leq \frac{dDH}{2\tilde{b}\alpha}\frac{s}{\sqrt{K}} + \frac{G^2\alpha}{\tilde{b}\tilde{\gamma}^2 h_0^*}\frac{1}{s\sqrt{K}} + \frac{\sqrt{dD}G}{\tilde{b}}\beta,\end{aligned} \tag{23}$$

where we use

$$\frac{1}{K}\sum_{k=1}^{K}\frac{1}{\sqrt{k}} \leq \frac{1}{K}\left(1 + \int_1^K \frac{\mathrm{d}t}{\sqrt{t}}\right) = \frac{1}{K}\left(2\sqrt{K} - 1\right) \leq \frac{2}{\sqrt{K}}.$$

An argument similar to the one for showing (21) and (22) ensures that (23) implies that

$$\min_{k\in[K]} \mathbb{E}\left[\|\nabla f(\boldsymbol{x}_k)\|^2\right] \leq \underbrace{\frac{dDH}{2\tilde{b}\alpha}}_{A_\alpha}\frac{s}{\sqrt{K}} + \underbrace{\frac{G^2\alpha}{\tilde{b}\tilde{\gamma}^2 h_0^*}}_{B_\alpha}\frac{1}{s\sqrt{K}} + \underbrace{\frac{\sqrt{dD}G}{\tilde{b}}}_{C_\beta}\beta\,. \tag{24}$$

(ii) A sufficient condition for (1), i.e.,

$$\min_{k\in[K]} \mathbb{E}\left[\|\nabla f(\boldsymbol{x}_k)\|^2\right] \leq \epsilon^2$$

is that the right-hand side of (24) is equal to ϵ^2, i.e.,

$$A_\alpha s^2 + (C_\beta - \epsilon^2)s\sqrt{K} + B_\alpha = 0,$$

which implies that

$$K(s) = \left\{\frac{A_\alpha s^2 + B_\alpha}{(\epsilon^2 - C_\beta)s}\right\}^2,$$

where $\epsilon^2 - C_\beta > 0$ is guaranteed from $\beta < \tilde{b}\epsilon^2/\sqrt{dD}G$. We have that

$$\frac{\mathrm{d}K(s)}{\mathrm{d}s} = \frac{2(A_\alpha s^2 + B_\alpha)}{(\epsilon^2 - C_\beta)^2 s^3}(A_\alpha s^2 - B_\alpha)\begin{cases} < 0 & \text{if } s \in (0, s^\star), \\ = 0 & \text{if } s = s^\star = \sqrt{\dfrac{B_\alpha}{A_\alpha}}, \\ > 0 & \text{if } s \in (s^\star, +\infty), \end{cases}$$

which implies that $K(s)$ attains the minimum $K(s^\star)$ when $s = s^\star$. □

A.6 Relationship between s and $K_\epsilon(s)$ for Algorithm 1 with diminishing learning rates

The following is a result for Algorithm 1 with diminishing sequences α_k and β_k.

Theorem A.2 *Suppose that Assumptions 2.1 and 2.2 hold and let $\epsilon > 0$, $\alpha \in (0, 1]$, and $\beta \in [0, b] \subset [0, 1)$.*

(i) *Consider Algorithm 1 with*

$$\alpha_k = \alpha_k(s) := \frac{\alpha}{s\sqrt{k}}\ (s > 0) \text{ and } \beta_k := \beta^k.$$

Then, for all $K \geq 1$ and all $s > 0$,

$$\min_{k\in[K]} \mathbb{E}\left[\|\nabla f(\boldsymbol{x}_k)\|^2\right] \leq \underbrace{\frac{dDH}{2(1-b)\alpha}}_{A_\alpha}\frac{s}{\sqrt{K}} + \underbrace{\frac{G^2\alpha}{(1-b)(1-\gamma)^2 h_0^*}}_{B_\alpha}\frac{1}{s\sqrt{K}} + \underbrace{\frac{\beta\sqrt{dD}G}{(1-b)(1-\beta)}}_{C_\beta}\frac{1}{K}.$$

(ii) *The number of steps K_ϵ needed to achieve (1) is expressed as the following rational function of batch size s:*

$$K_\epsilon(s) = \left\{\frac{(A_\alpha s^2 + B_\alpha) + \sqrt{(A_\alpha s^2 + B_\alpha)^2 + 4\epsilon^2 C_\beta s^2}}{2\epsilon^2 s}\right\}^2.$$

In particular, the minimum value of K_ϵ needed to achieve (1) is

$$\begin{aligned} K_\epsilon(s^\star) &= \left\{\frac{\sqrt{A_\alpha B_\alpha} + \sqrt{A_\alpha B_\alpha + \epsilon^2 C_\beta}}{\epsilon^2}\right\}^2 \\ &= \frac{\left\{\sqrt{(1-\beta)dDH}G + \sqrt{\left((1-\beta)dDGH + 2(1-b)\beta(1-\gamma)^2\epsilon^2\sqrt{dD}h_0^*\right)G}\right\}^2}{2(1-b)^2(1-\beta)(1-\gamma)^2\epsilon^4 h_0^*} \end{aligned}$$

when

$$s^\star = \sqrt{\frac{B_\alpha}{A_\alpha}} = \frac{G\alpha}{(1-\gamma)\sqrt{dDHh_0^*}}.$$

Proof: (i) Theorem A.1, together with $\alpha_k = \alpha/(s\sqrt{k})$ and $\beta_k = \beta^k$, guarantees that, for all $K \geq 1$, all $s > 0$, and all $\boldsymbol{x} \in \mathbb{R}^d$,

$$\begin{aligned}\frac{1}{K}\sum_{k=1}^{K} V_k(\boldsymbol{x}) &\leq \frac{dDH}{2\tilde{b}}\frac{1}{\alpha_K K} + \frac{G^2}{2\tilde{b}\tilde{\gamma}^2 h_0^*}\frac{1}{K}\sum_{k=1}^{K}\alpha_k + \frac{\sqrt{dD}G}{\tilde{b}}\frac{1}{K}\sum_{k=1}^{K}\beta^k \\ &\leq \frac{dDH}{2\tilde{b}\alpha}\frac{s}{\sqrt{K}} + \frac{G^2\alpha}{\tilde{b}\tilde{\gamma}^2 h_0^*}\frac{1}{s\sqrt{K}} + \frac{\beta\sqrt{dD}G}{\tilde{b}\tilde{\beta}}\frac{1}{K},\end{aligned} \tag{25}$$

where we use $\tilde{\beta} := 1 - \beta$,

$$\begin{aligned}&\frac{1}{K}\sum_{k=1}^{K}\frac{1}{\sqrt{k}} \leq \frac{1}{K}\left(1 + \int_1^K \frac{\mathrm{d}t}{\sqrt{t}}\right) = \frac{1}{K}\left(2\sqrt{K} - 1\right) \leq \frac{2}{\sqrt{K}}, \\ &\frac{1}{K}\sum_{k=1}^{K}\beta^k \leq \frac{1}{K}\sum_{k=1}^{+\infty}\beta^k = \frac{\beta}{\tilde{\beta}K}.\end{aligned}$$

An argument similar to the one for showing (21) and (22) ensures that (25) implies that

$$\min_{k\in[K]} \mathbb{E}\left[\|\nabla f(\boldsymbol{x}_k)\|^2\right] \leq \underbrace{\frac{dDH}{2\tilde{b}\alpha}}_{A_\alpha}\frac{s}{\sqrt{K}} + \underbrace{\frac{G^2\alpha}{\tilde{b}\tilde{\gamma}^2 h_0^*}}_{B_\alpha}\frac{1}{s\sqrt{K}} + \underbrace{\frac{\beta\sqrt{dD}G}{\tilde{b}\tilde{\beta}}}_{C_\beta}\frac{1}{K}. \tag{26}$$

(ii) A sufficient condition for (1), i.e.,

$$\min_{k\in[K]} \mathbb{E}\left[\|\nabla f(\boldsymbol{x}_k)\|^2\right] \leq \epsilon^2$$

is that the right-hand side of (26) is equal to ϵ^2, i.e.,

$$A_\alpha s^2\sqrt{K} + B_\alpha\sqrt{K} + C_\beta s - \epsilon^2 sK = 0,$$

which implies that

$$K(s) = \left\{\frac{(A_\alpha s^2 + B_\alpha) + \sqrt{(A_\alpha s^2 + B_\alpha)^2 + 4\epsilon^2 C_\beta s^2}}{2\epsilon^2 s}\right\}^2.$$

We have that

$$\frac{\mathrm{d}\sqrt{K(s)}}{\mathrm{d}s} = \frac{(A_\alpha s^2 + B_\alpha) + \sqrt{(A_\alpha s^2 + B_\alpha)^2 + 4\epsilon^2 C_\beta s^2}}{2\epsilon^2 s^2\sqrt{(A_\alpha s^2 + B_\alpha)^2 + 4\epsilon^2 C_\beta s^2}}(A_\alpha s^2 - B_\alpha)\begin{cases} < 0 & \text{if } s \in (0, s^\star), \\ = 0 & \text{if } s = s^\star = \sqrt{\dfrac{B_\alpha}{A_\alpha}}, \\ > 0 & \text{if } s \in (s^\star, +\infty), \end{cases}$$

which implies that $K(s)$ attains the minimum $K(s^\star)$ when $s = s^\star$. □

A.7 Conditions on β satisfying (7)

If β satisfies the condition in Theorem 3.1(ii) and if

$$\beta \leq \frac{(1-b)\sqrt{D_{\mathrm{SGD}}} - \sqrt{D_{\mathrm{NM}}}}{\sqrt{dD_{\mathrm{SGD}}D_{\mathrm{NM}}}G}\epsilon^2, \tag{27}$$

then

$$K_\epsilon(s^\star_{\mathrm{C,SGD}}) = \frac{dD_{\mathrm{SGD}}G^2}{\epsilon^4} \geq K_\epsilon(s^\star_{\mathrm{C,NM}}) = \frac{dD_{\mathrm{NM}}G^2}{\{(1-b)\epsilon^2 - \sqrt{dD_{\mathrm{NM}}}G\beta\}^2}. \tag{28}$$

Moreover, if β satisfies the condition in Theorem 3.1(ii) and if

$$\beta \leq \frac{(1-b)(1-\gamma)\sqrt{D_{\mathrm{SGD}}h_0^*} - \sqrt{D_{\mathrm{A}}H}}{(1-\gamma)\sqrt{dD_{\mathrm{SGD}}D_{\mathrm{A}}h_0^*}G}\epsilon^2, \tag{29}$$

then

$$K_\epsilon(s^\star_{\mathrm{C,SGD}}) \geq K_\epsilon(s^\star_{\mathrm{C,A}}) = \frac{dD_{\mathrm{A}}G^2H}{(1-\gamma)^2\{(1-b)\epsilon^2 - \sqrt{dD_{\mathrm{A}}}G\beta\}^2 h_0^*}. \tag{30}$$

If β satisfies the condition in Theorem 3.1(ii) and if

$$\beta \leq \frac{(1-b)\{(1-\gamma)\sqrt{D_{\mathrm{NM}}h_0^*} - \sqrt{D_{\mathrm{A}}H}\}}{\{(1-\gamma)\sqrt{dD_{\mathrm{NM}}D_{\mathrm{A}}h_0^*} - \sqrt{dD_{\mathrm{NM}}D_{\mathrm{A}}H}\}G}\epsilon^2, \tag{31}$$

then

$$K_\epsilon(s^\star_{\mathrm{C,NM}}) \geq K_\epsilon(s^\star_{\mathrm{C,A}}). \tag{32}$$

A.8 How to set β

Theorems 3.1(ii) and 3.2(ii) under (G1) indicate that the following restriction on β is needed:

$$\beta < \min\left\{\frac{1-b}{LndD}\epsilon^2, b\right\}. \tag{33}$$

Let us suppose that the number of samples n and the number of dimension d are both large and the precision accuracy ϵ is small. Then, β is small. The momentum term $\boldsymbol{m}_k$ (step 3 of Algorithm 1) satisfies

$$\boldsymbol{m}_k = \beta\boldsymbol{m}_{k-1} + (1-\beta)\nabla f_{\mathcal{S}_k}(\boldsymbol{x}_k) \approx \begin{cases} \nabla f_{\mathcal{S}_k}(\boldsymbol{x}_k) & \text{if } \beta \text{ satisfies (33)}, \\ \boldsymbol{m}_{k-1} & \text{if } \beta = 0.99 \text{ or } 0.9. \end{cases}$$

Accordingly, using β in (33) puts considerable emphasis on stochastic mini-batch sampling, which leads to the results such as $s^\star_{\mathrm{C,SGD}} \leq s^\star_{\mathrm{C,NM}} \leq s^\star_{\mathrm{C,A}}$ and $s^\star_{\mathrm{D,SGD}} = s^\star_{\mathrm{D,NM}} \leq s^\star_{\mathrm{D,A}}$.

A.9 Stochastic gradient complexity

Table 5: Stochastic gradient complexity (SGC) for ϵ-approximation of optimizers (SGD, N-Momentum, Adam-type, and SPIDER (Fang et al., 2018)) with constant and diminishing learning rates

	Stochastic Gradient Complexity	
	Constant Learning Rate Rule	Diminishing Learning Rate Rule
SGD	$\mathcal{O}\left(\frac{n^4\alpha}{\epsilon^6}\right)$	$\mathcal{O}\left(\frac{n^3\alpha}{\epsilon^4}\right)$
N-Momentum	$\mathcal{O}\left(\frac{n^4\alpha}{(\tilde{b}\epsilon^2 - dDLn\beta)^3}\right)$	$\mathcal{O}\left(\frac{n^3\alpha}{(\tilde{b}\epsilon^2 - dDLn\beta)^2}\right)$
Adam-type	$\mathcal{O}\left(\frac{n^4\alpha}{(\tilde{b}\epsilon^2 - dDLn\beta)^3}\right)$	$\mathcal{O}\left(\frac{n^3\alpha}{(\tilde{b}\epsilon^2 - dDLn\beta)^2}\right)$
SPIDER	$\mathcal{O}\left(n + \frac{\sqrt{n}}{\epsilon^2}\right)$	——

A theoretical investigation of Stochastic Path-Integrated Differential EstimatoR (SPIDER) for ϵ-approximation in nonconvex optimization was reported in (Fang et al., 2018). In particular, Theorem 2 in (Fang et al., 2018) clarified that SPIDER, which has a constant learning rate, for ϵ-approximation must use the full-batch gradient with the number of samples n or the stochastic gradient with batch size $\sqrt{n}$. Meanwhile, our results show the optimal batch size of SGD, N-Momentum, and Adam-type optimizers (see Tables 1 and 2). Table 5 indicates that the SGCs of SGD, N-Momentum, and Adam-type optimizers depend on a positive parameter α. For example, let us set $\alpha = \epsilon^2$ and focus on N-Momentum using diminishing learning rate. Then, the SGC of N-Momentum is $\mathcal{O}(n^3/\epsilon^2)$. The SGC of SPIDER is $\mathcal{O}(n + \sqrt{n}/\epsilon^2)$ (see (Fang et al., 2018, Table 1) for the SGCs of the variance-reduction type of optimizer).