Non-convex Stochastic Composite Optimization with Polyak Momentum

Yuan Gao [1 2]   Anton Rodomanov [1]   Sebastian U. Stich [1]

Abstract

The stochastic proximal gradient method is a powerful generalization of the widely used stochastic gradient descent (SGD) method and has found numerous applications in Machine Learning. However, it is notoriously known that this method fails to converge in non-convex settings where the stochastic noise is significant (i.e. when only small or bounded batch sizes are used). In this paper, we focus on the stochastic proximal gradient method with Polyak momentum. We prove this method attains an optimal convergence rate for non-convex composite optimization problems, regardless of batch size. Additionally, we rigorously analyze the variance reduction effect of the Polyak momentum in the composite optimization setting and we show the method also converges when the proximal step can only be solved inexactly. Finally, we provide numerical experiments to validate our theoretical results.

1. Introduction

Stochastic gradient descent and its variants are the workhorse of modern machine learning. The stochastic proximal gradient method is a simple yet powerful extension of the vanilla stochastic gradient descent method, which aims to solve the following stochastic composite optimization problem:

$$\min_{\mathbf{x} \in \mathbb{R}^d} \{F(\mathbf{x}) := f(\mathbf{x}) + \psi(\mathbf{x})\}, \qquad (1)$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is smooth and $\psi \colon \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is not necessarily differentiable, but a simple function. Such paradigm is ubiquitous in machine learning and beyond, and it covers a wide range of generalizations of the vanilla optimization problem, including regularized machine learning problems (Liu et al., 2015), signal processing (Combettes &

[1]CISPA, Saarbrücken, Germany [2]Universität des Saarlandes. Correspondence to: Yuan Gao <yuan.gao@cispa.de>.

Pesquet, 2010), image processing (Luke, 2020) and many more. It also naturally covers constrained optimization problems by considering the indicator function of the constraint set, and some recent variants of such ideas have been applied to distributed and federated machine learning problems (Mishchenko et al., 2022).

The problem formulation (1) is often used in Machine Learning as the proximal term ψ allows encoding prior domain knowledge. For instance, it can impose constraints on the variables, handle non-differentiability, induce sparsity, or it can take the form of a regularizer. Recently, there has been a surge of interest in these *preconditioning* techniques in the Deep Learning context (Hendrikx et al., 2020; Woodworth et al., 2023). Given a loss objective function $\ell \colon \mathbb{R}^d \to \mathbb{R}$, an auxiliary function (or preconditioner) $\hat{\ell}$ is constructed—for instance, by approximating ℓ with a subset of the data samples or synthetic samples. This results in the formulation:

$$\min_{\mathbf{x} \in \mathbb{R}^d} \{\underbrace{\ell(\mathbf{x}) - \hat{\ell}(\mathbf{x})}_{f(\mathbf{x})} + \underbrace{\hat{\ell}(\mathbf{x})}_{\psi(\mathbf{x})}\}.$$

This is a special case of Problem (1), with $f := \ell - \hat{\ell}$ and $\psi := \hat{\ell}$. This formulation has the advantage that the training can be accelerated if the proxy function $\hat{\ell}$ remains simple to optimize. These applications motivate the need to further our understanding of the stochastic composite optimization problem in the non-convex regime, especially where large batches are less preferred or even unavailable (LeCun et al., 2012; Rieke et al., 2020).

In the convex and strongly convex case, the complexity of solving Problem (1) is well understood (Ghadimi et al., 2016; Ghadimi & Lan, 2013a). On the other hand, the existing theory in the non-convex regime is unsatisfactory. In particular, when the gradient noise has variance σ^2, with the vanilla algorithm, the squared norm of the gradient of F is only shown to converge to $\mathcal{O}(1/K) + \Omega(\sigma^2)$ after K iterations. This implies that, to converge to an arbitrary ε error, one needs to take mega batches of size $\Omega(\varepsilon^{-1})$ at each iteration of the algorithm to reduce the stochastic noise term; and the total number of stochastic gradient oracle calls is $\Omega(\varepsilon^{-2})$. In practice and theory, smaller-batch methods are often preferred over mega-batch methods. For example, mega-batch methods follow the full gradient methods much more closely at each step than smaller batch methods, and it is observed empirically to be adversarial to the generaliza-

tion performance (Wilson & Martinez, 2003; LeCun et al., 2012; Keskar et al., 2017), and theoretically (Sekhari et al., 2021) for certain Machine Learning tasks. Furthermore, in many practical settings, mega-batches are unavailable or intractable to sample, e.g., in medical tasks (Rieke et al., 2020); federated Reinforcement Learning (Khodadadian et al., 2022; Jin et al., 2022); and multi-agent Reinforcement Learning (Doan et al., 2019).

In this work, we revisit the stochastic composite optimization problem in the non-convex regime and show that the Polyak momentum technique addresses the aforementioned mega-batch issue while retaining the optimal convergence rate $\mathcal{O}(\varepsilon^{-2})$. Polyak momentum is the de-facto standard for training Deep Learning models in practice (Kingma & Ba, 2014), and understanding its theoretical efficacy in various settings is an active research direction (Cutkosky & Mehta, 2020; Fatkhullin et al., 2023).

1.1. Stochastic Proximal Gradient Method

To solve the stochastic composite optimization problem (1), we consider the stochastic proximal gradient method, which solves the following subproblem in each iteration (commonly known as the proximal step): find $\mathbf{x}_{k+1}$ such that

$$\Omega_k(\mathbf{x}_{k+1}) \leq \Omega_k(\mathbf{x}_k) \quad \text{and} \quad \nabla\Omega_k(\mathbf{x}_{k+1}) = 0,$$

where $\Omega_k(\mathbf{x}) := \langle \mathbf{g}_k, \mathbf{x} \rangle + \psi(\mathbf{x}) + \frac{M_k}{2}\|\mathbf{x} - \mathbf{x}_k\|^2$. Here, $\mathbf{g}_k$ denotes a random vector computed at each iteration k using the stochastic first-order oracle of f, and $M_k > 0$ is a regularization (or stepsize) parameter. M_k can be a user defined constant, or it can be a non-decreasing sequence. For the vanilla stochastic proximal gradient method, $\mathbf{g}_k$ is a stochastic gradient of f at $\mathbf{x}_k$.

When ψ is convex, the proximal step is equivalent to minimizing Ω_k:

$$\mathbf{x}_{k+1} = \underset{\mathbf{x} \in \mathbb{R}^d}{\arg\min}\left\{ \langle \mathbf{g}_k, \mathbf{x} \rangle + \psi(\mathbf{x}) + \frac{M_k}{2}\|\mathbf{x} - \mathbf{x}_k\|^2 \right\}. \quad (2)$$

We note that when $\psi \equiv 0$, this reduces to the vanilla SGD with step size $\frac{1}{M_k}$. When $\psi \not\equiv 0$, it has long been known that if error-dependent batch sizes are not allowed, then such a method is only proved to converge to a neighborhood of the stationary point of F up to the variance of the gradient noise (Ghadimi et al., 2016). In Section 4 we give a simple example to illustrate this phenomenon. Therefore, one needs ε-dependent mini-batches (or mega-batches) to converge to an ε neighborhood of the stationary point of F.

1.2. Our Contributions

Stochastic Polyak momentum has seen a huge success in practice (Kingma & Ba, 2014). In this work, we study the effect of Polyak momentum in the non-convex regime and

provide a theoretical analysis of momentum's stabilizing effect in the stochastic composite optimization setting, particularly in the small batch regime.

- First, we establish a lower bound result for the vanilla stochastic proximal gradient method, showing that it cannot converge to the stationary point beyond the variance of the gradient noise.

- We study the effect of incorporating momentum into the stochastic proximal gradient method and prove its optimal convergence to the stationary point without any error-dependent mega-batch access. We also rigorously study the variance reduction effect of momentum in the stochastic composite optimization setting.

- We further extend our analysis to the case where the proximal steps are solved inexactly and give the same convergence guarantees, demonstrating the robustness of the momentum method.

- Finally, we conduct numerical experiments to corroborate our theoretical findings and demonstrate the algorithm's practicality.

2. Related Works

There is a huge body of work on stochastic and composite optimization. Here, we focus on the stochastic and non-convex regime and do not get into the details of the works in the convex (e.g. Ghadimi & Lan, 2012; 2013b) or the deterministic regime (see Nesterov, 2013). The first work that considers the stochastic composite optimization problem in the non-convex case appears to be (Ghadimi et al., 2016), in which they established the convergence of the stochastic proximal gradient method to a neighborhood of the stationary point of F up to the variance of the gradient noise. If mega-batches are allowed, they showed that the algorithm takes asymptotically $\mathcal{O}(\varepsilon^{-2})$ stochastic gradient oracle calls to converge to an ε neighborhood of the stationary point of F. It was later extended to incorporate the acceleration technique, but the convergence still requires mega-batches. In particular, it requires k samples at the kth iteration while the total number of oracle calls is not improved (Ghadimi & Lan, 2013a). Even though these methods require mega-batches, their upper bounds on the total number of stochastic gradient oracle calls match the lower bound asympotically (Arjevani et al., 2023).

In another research direction, to break the $\mathcal{O}(\varepsilon^{-2})$ lower bound for stochastic non-convex optimization, there have been a long line of works on variance reduction techniques, including Prox-Spiderboost (Wang et al., 2019) (composite variant of Spider (Fang et al., 2018)), Hybrid-SGD (Tran-Dinh et al., 2022), and PStorm (Xu & Xu, 2022) (composite variant of Storm (Cutkosky & Orabona, 2019)). These methods achieve a $\mathcal{O}(\varepsilon^{-3/2})$ asymptotic complexity at the cost

of two additional assumptions on the objective and problem structure:

- f is $\widetilde{L}$-average smooth: $\mathbb{E}[\|\nabla f(\mathbf{x}, \xi) - \nabla f(\mathbf{y}, \xi)\|^2] \leq \widetilde{L}^2 \|\mathbf{x} - \mathbf{y}\|^2$. Note that this assumption might not be satisfied for some simple smooth functions, and it is strictly stronger than the standard smoothness assumption.[1]

- These variance reduction techniques require the access to two stochastic gradients $\nabla f(\mathbf{x}_k, \xi_k)$ and $\nabla F(\mathbf{x}_{k-1}, \xi_k)$ at iteration k, which might not always be available in practice (e.g. Doan et al., 2019; Chen et al., 2022).

Convergence Criteria The first-order convergence criteria for non-convex optimization problems is an ongoing research topic. Our work mostly focuses on the convergence in terms of $\|\nabla F(\mathbf{x}_k)\|^2$, which we think is the most natural criteria. Many early works study the convergence of problem (1) in terms of the proximal gradient mapping (Ghadimi et al., 2016; Ghadimi & Lan, 2013a; Wang et al., 2019; Tran-Dinh et al., 2022; Xu & Xu, 2022). More recently, there have been several works proposing to study the convergence in terms of Moreau envelope and discussing how the different criterias affect the corresponding convergence complexities (Davis & Drusvyatskiy, 2019; Zhang et al., 2020). We discuss the differences and connections between these definitions in Appendix G.

Stochastic Polyak Momentum The idea of using Polyak momentum (a.k.a. heavy-ball momentum) was first proposed in (Polyak, 1964) for strongly convex quadratic objective in the deterministic case. The first non-asymptotic analysis of SGD with Polyak momentum in the smooth non-convex regime is given in (Yu et al., 2019) and was later refined in (Liu et al., 2020). Some recent works also studied how Polyak momentum can be used to remove the dependence on large batches in various settings, e.g. for normalized SGD (Cutkosky & Mehta, 2020) and for communication compressed optimization (Fatkhullin et al., 2023). The analysis of stochastic gradient methods with Polyak momentum remains an active research topic (Wang & Abernethy, 2020; Sebbouh et al., 2021; Li et al., 2022; Jelassi & Li, 2022).

3. Problem Formulation and Assumptions

In this work, we consider the composite optimization problem:

$$\min_{\mathbf{x} \in \mathbb{R}^d} \{F(\mathbf{x}) \coloneqq f(\mathbf{x}) + \psi(\mathbf{x})\}.$$

In the main body of the paper we study the convergence of the algorithms in terms of $\|\nabla F(\mathbf{x})\|^2$ where we assume that

---

[1] By Jensen's inequality, we have that any $\widetilde{L}$-average smooth function is also $\widetilde{L}$-smooth, see also Assumption 3.3. There exists L-smooth functions that are not average smooth.

ψ is differentiable. We discuss the case where ψ is convex and non-differentiable in Appendix F. We do not consider the non-convex and non-differentiable case for ψ, as this is an exotic scenario in the literature.

Note that most of the existing works on stochastic composite optimization assume that ψ is convex and analyze the convergence of their algorithms in terms of the proximal gradient mapping (Ghadimi et al., 2016; Ghadimi & Lan, 2013a; Wang et al., 2019; Tran-Dinh et al., 2022; Xu & Xu, 2022), which is closely related to what we consider in this work. In Appendix G, we discuss the proximal gradient mapping and argue that our definition is more natural in the non-convex regime.

We first introduce the following assumption on the monotonicity of the proximal step:

Assumption 3.1. We assume that, at each iteration k of the algorithms, the output $\mathbf{x}_{k+1}$ of the proximal step satisfies $\Omega_k(\mathbf{x}_{k+1}) \leq \Omega_k(\mathbf{x}_k)$.

This is a technical assumption that provides a more nuanced characterization of ψ beyond convexity, but all our theories work with just the simple convexity assumption as well (in fact, several constants can be improved under the convexity assumption). It is natural to assume that the proximal step finds a no worse point than the current iterate, even when ψ is non-convex.

Next, we introduce the lower boundedness assumption of the objective function F:

Assumption 3.2. We assume that there exists $F^\star \in \mathbb{R}$ such that $F^\star \leq F(\mathbf{x}), \forall \mathbf{x} \in \mathbb{R}^d$.

Now, we introduce the smoothness assumption on f. Note that this is a weaker assumption than the average smoothness assumption used in the variance reduction type methods (Wang et al., 2019; Tran-Dinh et al., 2022; Xu & Xu, 2022).

Assumption 3.3. We assume that the function f has L-Lipschitz gradient, i.e. for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, we have:

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|.$$

Next, we make an assumption on the noise of the gradient oracle of f.

Assumption 3.4. We assume that we have access to a gradient oracle $\nabla f(\mathbf{x}, \xi)$ for f such that for all $\mathbf{x} \in \mathbb{R}^d$ it holds that:

$$\begin{aligned} \mathbb{E}[\nabla f(\mathbf{x}, \xi)] &= \nabla f(\mathbf{x}), \\ \mathbb{E}[\|\nabla f(\mathbf{x}, \xi) - \nabla f(\mathbf{x})\|^2] &\leq \sigma^2. \end{aligned} \tag{3}$$

This is a standard assumption in the analysis of stochastic gradient methods (Nemirovskij & Yudin, 1983; Bubeck et al., 2015). Taking mini-batches is equivalent to dividing the variance by the batch size.

4. Lower Bound for the Vanilla Stochastic Proximal Gradient Method

In Section 1.1 we briefly discussed that the vanilla stochastic proximal gradient method cannot converge to the stationary point of F beyond the variance of the gradient noise. We give a simple example to illustrate this phenomenon and give some intuitions on why momentum resolves the issue.

Proposition 4.1. *For any $K \geq 1$ and any (predefined) stepsize coefficients $\{M_k\}_{k=0}^{K-1}$ (possibly depending on the problem parameters L, σ^2 and K), there exists a problem instance of (1) with $f(\mathbf{x}) := \frac{L}{2}\|\mathbf{x}\|^2$ and $\psi(\mathbf{x}) := \frac{a}{2}\|\mathbf{x}\|^2$, $a := \max_{0 \leq k \leq K-1} M_k$, satisfying Assumptions 3.2 and 3.3, and the stochastic gradient oracle $\nabla f(\mathbf{x}, \xi) := \nabla f(\mathbf{x}) + \xi$, $\xi \sim \mathcal{N}(0, \sigma^2\mathbf{I})$, satisfying Assumption 3.4, such that, for the sequence $\{\mathbf{x}_k\}_{k=1}^K$ generated by method (2) started from any initial point $\mathbf{x}_0$, it holds that $\mathbb{E}[\|\nabla F(\mathbf{x}_k)\|^2] \geq \frac{1}{4}\sigma^2$ for any $1 \leq k \leq K$.*

Proof. Clearly, the construction satisfies our assumptions. Let us fix an arbitrary $0 \leq k \leq K-1$. Setting the gradient of the auxiliary function in (2) to zero, we see that $\mathbf{g}_k + a\mathbf{x}_{k+1} + M_k(\mathbf{x}_{k+1} - \mathbf{x}_k) = 0$. Since $\mathbf{g}_k = L\mathbf{x}_k + \xi_k$, it follows that

$$\mathbf{x}_{k+1} = \frac{M_k - L}{M_k + a}\mathbf{x}_k - \frac{\xi_k}{M_k + a}.$$

Substituting $\nabla F(\mathbf{x}_{k+1}) = (L + a)\mathbf{x}_{k+1}$, we obtain

$$\mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2] = \frac{(L+a)^2}{(M_k+a)^2}\mathbb{E}[\|(M_k-L)\mathbf{x}_k - \xi_k\|^2]$$

$$= \frac{(L+a)^2}{(M_k+a)^2}\left((M_k - L)^2\mathbb{E}[\|\mathbf{x}_k\|^2] + \sigma^2\right) \geq \frac{(L+a)^2\sigma^2}{(M_k+a)^2}.$$

If $a \geq M_k$, then $\frac{L+a}{M_k+a} \geq \frac{1}{2}$, and the claim follows. $\square$

Note that Proposition 4.1 holds for any initial point $\mathbf{x}_0$. Even if we start at the optimal point $\mathbf{x}_0 = \mathbf{0}$, the very first step of the method will already incur an $\mathcal{O}(\sigma^2)$ error and no subsequent steps will be able to reduce it.

It is important that, for the composite optimization problem (1), we do not make any significant assumptions on ψ. In particular, we do not assume that ψ is smooth with a certain smoothness constant. Proposition 4.1 demonstrates that, without such extra assumptions, for any fixed choice of parameters for the stochastic proximal gradient method (2), i.e., the stepsize coefficients M_k and the number of iterations K, there is always a "bad" function in our problem class (namely, $F(\mathbf{x}) = f(\mathbf{x}) + \psi(\mathbf{x})$ with $f(\mathbf{x}) = \frac{L}{2}\|\mathbf{x}\|^2$ and $\psi(\mathbf{x}) = \frac{a}{2}\|\mathbf{x}\|^2$ for a sufficiently large a) for which the method cannot reach any error $< \frac{1}{4}\sigma^2$ after K steps. In other words, for any given target accuracy $\varepsilon < \frac{1}{4}\sigma^2$, it is impossible to find *one specific* choice of the parameters for

---

Algorithm 1 Proximal Gradient Method with Polyak Momentum

1: **Input:** $\mathbf{x}_0, \mathbf{m}_{-1}$ and $\{M_k\}_{k=0}^{\infty}, \{\gamma_k\}_{k=-1}^{\infty}, \{\delta_k\}_{k=0}^{\infty}$,
2: **for** $k = 0, 1, 2, \ldots$ **do**
3:     Compute $\mathbf{g}_k = \nabla f(\mathbf{x}_k, \xi_k)$
4:     Update $\mathbf{m}_k = (1 - \gamma_{k-1})\mathbf{m}_{k-1} + \gamma_{k-1}\mathbf{g}_k$
5:     Compute approximate stationary point $\mathbf{x}_{k+1}$ of $\Omega_k(\mathbf{x}) := \langle \mathbf{m}_k, \mathbf{x}\rangle + \psi(\mathbf{x}) + \frac{M_k}{2}\|\mathbf{x} - \mathbf{x}_k\|^2$ such that $\Omega_k(\mathbf{x}_{k+1}) \leq \Omega_k(\mathbf{x}_k)$ and $\|\nabla\Omega_k(\mathbf{x}_{k+1})\|^2 \leq \delta_k$
6: **end for**

---

the method allowing it to reach the ε-error on *any* problem from our class.

The problem is that the variance of the gradient noise keeps the iterates away from the stationary point of F, and we need some mechanism reducing this variance with time. One such mechanism is the Polyak momentum which we discuss next.

5. The Algorithm and Analysis

In this section, we present the stochastic proximal gradient method with the Polyak momentum and establish its convergence guarantees in the non-convex regime.

The method is shown in Algorithm 1. In the algorithm, the stochastic gradient $\mathbf{g}_k$ at each iteration k is replaced by the momentum-aggregated gradient mean:

$$\mathbf{m}_k = (1 - \gamma_{k-1})\mathbf{m}_{k-1} + \gamma_{k-1}\mathbf{g}_k.$$

Our analysis will show that the distance between the momentum and the full gradient decreases as the number of iterations increases, which is the key to resolving the issue of the vanilla stochastic proximal gradient method.

In this section, we assume that the proximal steps are solved exactly, i.e. we further make the following assumption:

Assumption 5.1. At each iteration k, we have $\delta_k = 0$, i.e. $\nabla\Omega_k(\mathbf{x}_{k+1}) = 0$.

We relax Assumption 5.1 and discuss the inexact proximal steps in Section 7.

5.1. Convergence Analysis

We discuss the convergence analysis here, and missing proofs can be found in Appendix B. The convergence analysis of Algorithm 1 revolves around the following quantities:

$$\begin{aligned} F_k &:= \mathbb{E}[F(\mathbf{x}_k) - F^\star], \\ \Delta_k &:= \mathbb{E}[\|\mathbf{m}_k - \nabla f(\mathbf{x}_k)\|^2], \quad\quad (4) \\ R_k &:= \mathbb{E}[\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2]. \end{aligned}$$

F_k quantifies the distance between the current objective value and the lower bound. Δ_k bounds the distance be-

tween the current gradient estimate and the full gradient. R_k bounds the distance between two consecutive iterates. We start by giving a descent lemma on Δ_k, which is key to analyzing the variance reduction effect of momentum. Similar statements can be found in (Cutkosky & Mehta, 2020) and (Fatkhullin et al., 2023).

Lemma 5.2. *Under Assumption 3.4, for any $k \geq 0$:*

$$\Delta_{k+1} \leq (1 - \gamma_k)\Delta_k + \frac{L^2}{\gamma_k}R_k + \gamma_k^2\sigma^2 .$$

The $(1 - \gamma_k)$ factor in the above lemma is the key to show that Δ_k decreases in the optimization process. Next, we discuss the per-iteration descent of F_k.

Recall that in this section, we have Assumption 5.1, which is equivalent to the following stationarity assumption:

$$\nabla\psi(\mathbf{x}_{k+1}) + \mathbf{m}_k + M_k(\mathbf{x}_{k+1} - \mathbf{x}_k) = 0. \qquad (5)$$

When ψ is convex and non-differentiable, $\nabla\Omega_k(\mathbf{x}_{k+1})$ is just a subgradient of Ω_k at $\mathbf{x}_{k+1}$ that equals to zero, whose existence is guaranteed as the optimality condition. In the following lemmas and theorems, we can simply replace the gradient of Ω and ψ with such choices of the subgradients and obtain the same results for convex and non-differentiable ψ.

In Section 7, we give an approximate stationarity assumption, which is more realistic in practice and shows the same convergence guarantees.

Now we give the following descent lemma on F_k:

Lemma 5.3. *Under Assumptions 3.1, 3.3 and 5.1, for any $k \geq 0$, we have*

$$F_{k+1} \leq F_k - \frac{M_k - L}{4}R_k + \frac{\Delta_k}{M_k - L}.$$

Remark 5.4. The constants in Lemma 5.3 can be slightly improved if Assumption 3.1 is replaced by the convexity of ψ.

Proof. By Assumption 3.3, we have:

$$\begin{aligned}
F(\mathbf{x}_{k+1}) &= f(\mathbf{x}_{k+1}) + \psi(\mathbf{x}_{k+1}) \\
&\leq f(\mathbf{x}_k) + \langle\nabla f(\mathbf{x}_k), \mathbf{x}_{k+1} - \mathbf{x}_k\rangle \\
&\quad + \frac{L}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + \psi(\mathbf{x}_{k+1}) \\
&= f(\mathbf{x}_k) + \langle\mathbf{m}_k, \mathbf{x}_{k+1} - \mathbf{x}_k\rangle + \psi(\mathbf{x}_{k+1}) \\
&\quad + \frac{M_k}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 - \frac{M_k - L}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\
&\quad + \langle\nabla f(\mathbf{x}_k) - \mathbf{m}_k, \mathbf{x}_{k+1} - \mathbf{x}_k\rangle .
\end{aligned}$$

Then we apply Assumption 3.1 and notice that $\Omega_k(\mathbf{x}_k) = \psi(\mathbf{x}_k) + \langle\mathbf{m}_k, \mathbf{x}_k\rangle$:

$$\begin{aligned}
F(\mathbf{x}_{k+1}) &\leq F(\mathbf{x}_k) - \frac{M_k - L}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\
&\quad + \langle\nabla f(\mathbf{x}_k) - \mathbf{m}_k, \mathbf{x}_{k+1} - \mathbf{x}_k\rangle .
\end{aligned}$$

Now we apply Young's equality and get

$$\begin{aligned}
F(\mathbf{x}_{k+1}) &\leq F(\mathbf{x}_k) - \frac{M_k - L}{4}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\
&\quad + \frac{\|\mathbf{m}_k - \nabla f(\mathbf{x}_k)\|^2}{M_k - L} .
\end{aligned}$$

We get the desired result by subtracting $F^\star$ and taking expectation on both sides. $\square$

Note that these will be enough if we aim to prove convergence in terms of the distance between the iterates, but not enough to guarantee a convergence in terms of $\mathbb{E}[\|\nabla F(\mathbf{x}_k)\|^2]$. Therefore, we relate the distance between iterates and the norm of the gradient in the following lemma:

Lemma 5.5. *Under Assumptions 3.3 and 5.1, for any $k \geq 0$,*

$$(M_k^2 + L^2)R_k \geq \frac{1}{3}\mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2] - \Delta_k.$$

Proof. We can split $\nabla F(\mathbf{x}_{k+1})$ in the following way:

$$\begin{aligned}
\nabla F(\mathbf{x}_{k+1}) &= \nabla f(\mathbf{x}_{k+1}) + \nabla\psi(\mathbf{x}_{k+1}) \\
&= \mathbf{m}_k + \nabla\psi(\mathbf{x}_{k+1}) + (\nabla f(\mathbf{x}_k) - \mathbf{m}_k) \\
&\quad + (\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)) .
\end{aligned}$$

Therefore,

$$\begin{aligned}
\|\nabla F(\mathbf{x}_{k+1})\|^2 &\leq 3\|\mathbf{m}_k + \nabla\psi(\mathbf{x}_{k+1})\|^2 \\
&\quad + 3\|\mathbf{m}_k - \nabla f(\mathbf{x}_k)\|^2 + 3\|\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)\|^2.
\end{aligned}$$

Now apply the stationarity condition (5) on the first term, and use Assumption 3.3 on the third term, we get:

$$\begin{aligned}
\|\nabla F(\mathbf{x}_{k+1})\|^2 &\leq 3(M_k^2 + L^2)\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\
&\quad + 3\|\mathbf{m}_k - \nabla f(\mathbf{x}_k)\|^2 .
\end{aligned}$$

Rearranging and taking expectations, we get the claim. $\square$

Now, we can piece together all of the above lemmas and consider the following Lyapunov function:

$$\Phi_k := F_k + a\Delta_k , \qquad (6)$$

where a is a constant to be determined later. Note that a does not have an impact on the algorithm itself, and it only shows up in the analysis. Now we give the following lemma on Φ_k:

Lemma 5.6. *Let Assumptions 3.1, 3.3, 3.4 and 5.1 hold, and let $a := \frac{3}{8L}$, $\gamma_k := \frac{3L}{M_k - L}$ and $M_k > 4L$ for any $k \geq 0$. Then, for any $k \geq 0$,*

$$\Phi_{k+1} \leq \Phi_k - \frac{1}{48 M_k} \mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2] + \frac{27 L \sigma^2}{4 M_k^2}. \quad (7)$$

We have the following simple corollary for constant stepsize coefficients:

Corollary 5.7. *Let Algorithm 1 be run for $K \geq 1$ iterations for solving problem (1) under Assumptions 3.3, 3.4 and 5.1, with constant coefficients $M_k = M = 4L + \frac{3^{3/2}}{2}\sqrt{\frac{KL\sigma^2}{\Phi_0}}$ and $\gamma_k = \frac{3L}{M-L}$ for any $0 \leq k \leq K-1$, where $\Phi_0 := F(\mathbf{x}_0) - F^* + \frac{3}{8L}\mathbb{E}[\|\mathbf{m}_0 - \nabla f(\mathbf{x}_0)\|^2]$. Then,*

$$\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \leq 48(3^{3/2})\sqrt{\frac{L\Phi_0\sigma^2}{K}} + \frac{192 L\Phi_0}{K}, \quad (8)$$

where t is chosen uniformly at random from $\{1, \ldots, K\}$.

It is also possible to use time-varying coefficients M_k which do not require fixing the number of iterations in advance. However, in terms of the convergence rate, it incurs an extra logarithmic factor.

Corollary 5.8. *Consider Algorithm 1 for solving problem (1) under Assumptions 3.3, 3.4 and 5.1 with coefficients $M_k = \max\{\sqrt{\frac{(k+1)L\sigma^2}{\Phi_0}}, 4L\}$, $\gamma_k = \frac{3L}{M_k - L}$ for any $k \geq 0$, where $\Phi_0 := F(\mathbf{x}_0) - F^* + \frac{3}{8L}\mathbb{E}[\|\mathbf{m}_0 - \nabla f(\mathbf{x}_0)\|^2]$. Then, for any $k \geq 1$, we have*

$$\mathbb{E}[\|\nabla F(\mathbf{x}_{t(k)})\|^2] \leq 372 \ln(ek)\left(\sqrt{\frac{L\Phi_0\sigma^2}{k}} + \frac{L\Phi_0}{k}\right), \quad (9)$$

where $t(k)$ is chosen randomly from $\{1, \ldots, k\}$ with probabilities $\Pr(t(k) = i) \propto \frac{1}{M_{i-1}}$, $i = 1, \ldots, k$, and $e := \exp(1)$.

Let us point out that using a random iterate as the output of the algorithm is standard in the literature (see, e.g., (Rakhlin et al., 2012)) and can be efficiently implemented without fixing the number of iterations in advance. We discuss this more carefully in Appendix E.

5.2. Initialization and Convergence Guarantees

As we can see from Corollaries 5.7 and 5.8, the convergence rate of Algorithm 1 depends on $\Phi_0 := \mathcal{O}(F_0 + \frac{1}{L}\Delta_0)$, where $\Delta_0 := \mathbb{E}[\|\mathbf{m}_0 - \nabla f(\mathbf{x}_0)\|^2] = \mathbb{E}[\|(1 - \gamma_{-1})\mathbf{m}_{-1} + \gamma_{-1}\mathbf{g}_0 - \nabla f(\mathbf{x}_0)\|^2]$ depends on $\mathbf{m}_{-1}$. There are subtle differences in how $\mathbf{m}_{-1}$ can be initialized between the non-composite and composite cases.

Non-Composite Case: When $\psi \equiv 0$, i.e. $F \equiv f$, we set the initial momentum $\mathbf{m}_{-1} := -\frac{\gamma_{-1}}{1 - \gamma_{-1}}\mathbf{g}_0$, then $\Delta_0 =$

$\mathbb{E}[\|\mathbf{m}_0 - \nabla f(\mathbf{x}_0)\|^2] = \|\nabla f(\mathbf{x}_0)\|^2 \leq 2LF_0$, where the last inequality follows from Assumption 3.3 and the fact that $F \equiv f$. Therefore, in the non-composite case, we can initialize the parameters such that $\Phi_0 = \mathcal{O}(F_0)$.

Composite Case: The composite case is slightly trickier. We set $\mathbf{m}_{-1} := \mathbf{g}_0$ and get that $\mathbf{m}_0 = \mathbf{g}_0$. Hence $\Delta_0 = \mathbb{E}[\|\mathbf{g}_0 - \nabla f(\mathbf{x}_0)\|^2] \leq \sigma^2$. Therefore, $\Phi_0 = \mathcal{O}(F_0 + \frac{\sigma^2}{L})$. When $\sigma^2 = \mathcal{O}(LF_0)$, we get the same $\Phi_0 = \mathcal{O}(F_0)$ as in the non-composite case.

Mini-Batch Initialization: In the case that σ^2 is much larger than F_0, if we have access to a constant size (not depending on the target error) mini-batch initially, then we can set $\mathbf{g}_0 = \frac{1}{b_0}\sum_{i=1}^{b_0} \nabla f(\mathbf{x}_0, \xi_i)$ where $b_0 := \lceil \frac{\sigma^2}{LF_0} \rceil$, i.e. $\mathbf{g}_0$ is a mini-batch stochastic gradient of size b_0. Then we have $\Phi_0 = \mathcal{O}(F_0 + \frac{\sigma^2}{b_0 L}) = \mathcal{O}(F_0)$, which is the same as in the non-composite case.

We have thus proved the following convergence guarantee for Algorithm 1.

Theorem 5.9. *Consider Algorithm 1, as applied to solving problem (1) under Assumptions 3.3, 3.4 and 5.1, run for $K = \mathcal{O}\left(\frac{L\Phi_0\sigma^2}{\varepsilon^2} + \frac{L\Phi_0}{\varepsilon}\right)$ iterations with constant coefficients $M_k = M = 4L + \frac{3^{3/2}}{2}\sqrt{\frac{KL\sigma^2}{\Phi_0}}$ and $\gamma_k = \frac{3L}{M-L}$ for any $0 \leq k \leq K-1$, where $\Phi_0 := F_0 + \frac{3}{8L}\mathbb{E}[\|\mathbf{m}_0 - \nabla f(\mathbf{x}_0)\|^2]$, $F_0 := F(\mathbf{x}_0) - F^*$ and $\varepsilon > 0$ is a given target error. Then, for the point $\mathbf{x}_t$ chosen uniformly at random from $\{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ it holds that $\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \leq \varepsilon$.*

If $\psi \equiv 0$, we can initialize $\mathbf{m}_{-1}$ in such a way that $K = \mathcal{O}\left(\frac{LF_0\sigma^2}{\varepsilon^2} + \frac{LF_0}{\varepsilon}\right)$. Otherwise, we can initialize $\mathbf{m}_{-1}$ in such a way that $K = \mathcal{O}\left(\frac{LF_0\sigma^2}{\varepsilon^2} + \frac{\sigma^4}{\varepsilon^2} + \frac{LF_0}{\varepsilon} + \frac{\sigma^2}{\varepsilon}\right)$. Further, when the initial mini-batch of size $\lceil \frac{\sigma^2}{LF_0} \rceil$ is allowed, we can initialize $\mathbf{m}_{-1}$ in such a way that $K = \mathcal{O}\left(\frac{LF_0\sigma^2}{\varepsilon^2} + \frac{LF_0}{\varepsilon}\right)$.

When an initial mini-batch is not allowed, the convergence rate has an extra $\mathcal{O}(\frac{\sigma^4}{\varepsilon^2} + \frac{\sigma^2}{\varepsilon})$ term that is not present in the non-composite case. One natural question is whether this is an artifact of our analysis or inherent to the problem and the algorithm. Note that the first step of the Algorithm 1 coincides with the vanilla stochastic proximal gradient method. Consider the lower bound construction in Proposition 4.1: when starting at $\mathbf{x}_0 = 0$, which is a stationary point such that $F_0 = 0$, one such step would incur an $\mathcal{O}(\sigma^2)$ error. Therefore, in the non-composite case without the initial mini-batch, the convergence rate must have some term that only depends on σ^2. In other words, the extra terms seem unavoidable.

6. Variance Reduction Effect of Momentum

In Section 4, we demonstrated that the vanilla stochastic proximal gradient method cannot converge because the gradient noise variance keeps the iterates away from the stationary point of F. In this section, we show that the momentum term in Algorithm 1 can reduce the variance of the gradient noise at the same rate as the gradient norm. This is the key to the convergence of Algorithm 1 without batches. This variance reduction effect has been known in practice and implicitly used in the analysis of (Cutkosky & Mehta, 2020) and (Fatkhullin et al., 2023). We precisely characterize such an effect in the composite optimization setting. Proofs are deferred to Appendix C.

We begin by refining the result of Lemma 5.6:

Lemma 6.1. *Let Assumptions 3.1, 3.3, 3.4 and 5.1 hold, and let $a := \frac{\sqrt{2}}{8L}$, $\gamma_k := \frac{3\sqrt{2}L}{M_k - L}$ and $M_k > (1 + 3\sqrt{2})L$ for any $k \geq 0$. Then, for any $k \geq 0$,*

$$\Phi_{k+1} \leq \Phi_k - \frac{1}{48M_k}\mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2]$$
$$- \frac{3\sqrt{2}}{2M_k}\Delta_k + \frac{27\sqrt{2}L}{4M_k^2}\sigma^2. \quad (10)$$

With Lemma 6.1, we can now precisely quantify the variance reduction effect of the momentum method.

Theorem 6.2. *Consider Algorithm 1, as applied to solving problem (1) under Assumptions 3.3, 3.4 and 5.1, run for $K = \mathcal{O}\left(\frac{L\Phi_0\sigma^2}{\varepsilon^2} + \frac{L\Phi_0}{\varepsilon}\right)$ iterations with constant coefficients $M_k = M = (1 + 3\sqrt{2})L + \frac{3^{3/2}}{2^{3/4}}\sqrt{\frac{KL\sigma^2}{\Phi_0}}$ and $\gamma_k = \frac{3\sqrt{2}L}{M-L}$ for any $0 \leq k \leq K - 1$, where $\Phi_0 := F_0 + \frac{\sqrt{2}}{8L}\mathbb{E}[\|\mathbf{m}_0 - \nabla f(\mathbf{x}_0)\|^2]$, $F_0 := F(\mathbf{x}_0) - F^*$ and $\varepsilon > 0$ is a given target error. Then, for the point $\mathbf{x}_t$ chosen uniformly at random from $\{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ it holds that $\mathbb{E}[\|\mathbf{m}_t - \nabla f(\mathbf{x}_t)\|^2] \leq \varepsilon$.*

In words, the squared distance between the momentum $\mathbf{m}_k$ and the full gradient $\nabla f(\mathbf{x}_k)$ decreases at the same rate as the squared norm of the gradient $\nabla F(\mathbf{x}_k)$.

7. Inexact Proximal Step

While many existing works (e.g. (Ghadimi et al., 2016; Ghadimi & Lan, 2013a; Wang et al., 2019; Hendrikx et al., 2020; Tran-Dinh et al., 2022; Xu & Xu, 2022)) rely on the assumption that the proximal step can be solved exactly, this might not always be practically possible (Barré et al., 2023). In this section, we briefly discuss extending our analysis to the case where the proximal step is solved inexactly and give an inexactness criterion similar to that of (Woodworth et al., 2023).

A crucial element in our previous analysis is the assumption that $\nabla\Omega_k(\mathbf{x}_{k+1}) = 0$, i.e. Equation (5). Therefore, defining

the inexactness criteria as an approximate stationarity condition of Ω_k at $\mathbf{x}_{k+1}$ where $\delta_k \neq 0$ is natural. In particular, we define the criteria as follows:

$$\mathbb{E}[\|\nabla\Omega_k(\mathbf{x}_{k+1})\|^2] \leq \frac{M_k^2}{16}\mathbb{E}[\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2] + S_k, \quad (11)$$

where S_k will be decided later. Again, it should be understood that in the non-differentiable case $\nabla\Omega_k(\mathbf{x}_{k+1})$ is a certain subgradient of Ω_k at $\mathbf{x}_{k+1}$ such that Equation (11) holds.

Now we state the convergence result (proofs are deferred to Appendix D):

Theorem 7.1. *Consider Algorithm 1, as applied to solving problem (1) under Assumptions 3.1, 3.3 and 3.4, and the approximate stationarity condition at each iteration k: $\mathbb{E}[\|\nabla\Omega_k(\mathbf{x}_{k+1})\|^2] \leq \frac{M^2}{16}\mathbb{E}[\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2] + S_k$, run for $K = \mathcal{O}\left(\frac{L\Phi_0\sigma^2}{\varepsilon^2} + \frac{L\Phi_0}{\varepsilon}\right)$ iterations with constant coefficients $M_k = M = 4L + \sqrt{\frac{8KL\sigma^2}{\Phi_0}}$ and $\gamma_k = \sqrt{\frac{152}{17}}\frac{L}{M-L}$ for any $0 \leq k \leq K - 1$, where $\Phi + 0 = F_0 + \sqrt{\frac{19}{156}\frac{\mathbb{E}[\|\mathbf{m}_0 - \nabla f(\mathbf{x}_0)\|^2]}{L}}$, $F_0 := F(\mathbf{x}_0) - F^*$ and $\varepsilon > 0$ is a given target error. Then, for the point $\mathbf{x} - t$ chosen uniformly at random from $\{\mathbf{x}_1, \ldots, \mathbf{x}_K\}$ it holds that $\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \leq \frac{\varepsilon}{2} + \frac{8}{K}\sum_{k=0}^{K-1} S_k$. In particular, if for any $0 \leq k \leq K - 1$, $S_k \leq \frac{\varepsilon}{16}$, then $\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \leq \varepsilon$.*

It remains to discuss how to minimize Ω_k such that the inexactness criteria (11) is satisfied. There is a line of research on minimizing gradient norm with SGD variants, e.g. (Allen-Zhu, 2018), and one can also exploit the structure information in ψ if there is any. Here, we state that SGD suffices for our purpose and give the following proposition, which is a simple modification of Proposition 2.6 in (Woodworth et al., 2023):

Proposition 7.2. *If ψ is convex and L_ψ-smooth, and we have access to an unbiased gradient oracle of ψ with variance at most σ_ψ^2, then after at most*

$$T = \mathcal{O}\left(\frac{L_\psi + M}{M}\ln\frac{L_\psi + M}{M} + \frac{(L_\psi + M)\sigma_\psi^2}{MS_k}\right)$$

iterations of SGD, the output $\hat{\mathbf{x}}$ of SGD satisfies the condition (11).

The proof of Proposition 7.2 is simple, and we refer interested readers to (Woodworth et al., 2023) for the discussions therein.

8. Experiments

In this section, we conduct numerical experiments to corroborate our theoretical findings and demonstrate the practical effectiveness of Algorithm 1.
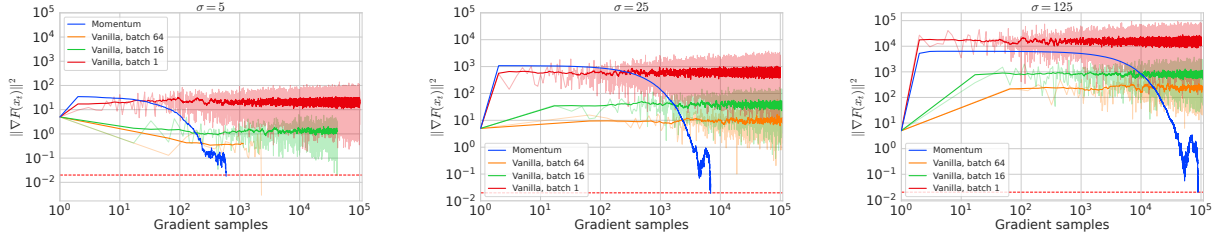
Figure 1: Comparison of Algorithm 1 and the vanilla stochastic proximal gradient method on the synthetic quadratic problem. For the vanilla stohastic proximal methods, we also highlight the smoothed curves on top of the original curves that oscillate much more. The left, middle, and right figures correspond to $\sigma = 5, 25, 125$, respectively. The vanilla stochastic proximal gradient method uses batch sizes $1, 16, 64$. The x-axis represents the number of gradient samples and is truncated to only show the first 10^5 gradient samples.

8.1. Synthetic Quadratic Problem

We first consider a synthetic quadratic problem inspired by our lower bound construction in Proposition 4.1. We consider the following problem in $\mathbb{R}^d$: we set $f(\mathbf{x}) = \frac{L\|\mathbf{x}\|^2}{2}$ and $\psi(\mathbf{x}) = \frac{a\|\mathbf{x}\|^2}{2}$, where a is sufficiently large. Note that the vanilla stochastic proximal gradient method and Algorithm 1 are oblivious to the parameter a. We add Gaussian noise to the gradients to control the stochastic level σ^2 of the stochastic oracle. We simulate the batch sample by dividing the variance by the batch size.

In Figure 1, we compare the performance of Algorithm 1 and the vanilla stochastic proximal gradient method. We set $d = 5, L = 1$ and $a = 10^4$. We run the vanilla method with batch sizes $1, 16, 64$. We set $\sigma = 5, 25, 125$ respectively. The parameter M is tuned by a grid search in $\{10^0, 10^1, 10^2, 10^3, 10^4\}$ for all methods, and the momentum parameter γ is tuned by a grid search in $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$. We set the maximum number of iterations to be 10^4, and the tolerance is 0.02. We see that as σ^2 increases, Algorithm 1 still reaches the desired tolerance, while the vanilla method with batch size 1 fails to converge in all cases, and with batch sizes 16 and 64 only converges when $\sigma = 5$. In particular, with batch size 1, the error of vanilla method oscillates around 22 with $\sigma^2 = 5^2$, around 672 with $\sigma^2 = 25^2$, and around 14381 with $\sigma^2 = 125^2$. In other words, the error of the vanilla method is indeed proportional to σ^2, as predicted by our lower bound result in Proposition 4.1. We also have that for Algorithm 1, M is set to be $10, 10, 100$ and γ is set to be $0.01, 0.001, 0.0001$ for $\sigma = 5, 25, 125$ respectively, which is consistent with our theoretical prediction that M should increase while γ should decrease as σ increases. We also point out that the momentum method exhibits a jump in the error in the first several iterations, consistent with our analysis that the first step of the momentum method incurs an $\mathcal{O}(\sigma^2)$ error as well.

8.2. Regularized Machine Learning Experiment

Now we consider the classical application of composite optimization: regularized machine learning (Liu et al., 2015). We use the $\ell_{\infty,1}$ regularizer to regularize the weights on each layer. The proximal step with the $\ell_{\infty,1}$ regularizer is implemented in (Murray et al., 2019). We evaluate the performances of Algorithm 1 and the vanilla stochastic proximal gradient method on the Cifar-10 dataset (Krizhevsky et al., 2014) with the Resnet-18 (He et al., 2016). The regularization parameter is set to be 0.1, which is observed in (Murray et al., 2019) to achive a balance between enforcing sparsity in the model and maintaining the model performance. We use a batch size of 256 and run 300 epochs. We use the standard step size parameter $M = 10$ (corresponding to a learning rate of 0.1) for the experiment. We apply a multi-step learning rate scheduler at 150 epoch and 250 epoch, with a decay factor of 0.1. For Algorithm 1, momentum parameter γ is set to be 0.1 by a grid search.
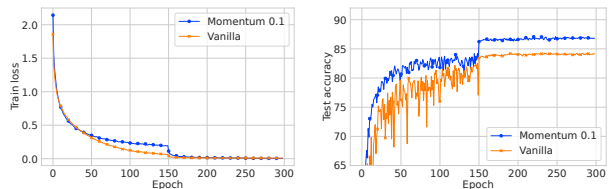

Figure 2: Comparison of Algorithm 1 and the vanilla stochastic proximal gradient method for the $\ell_{\infty,1}$ regularized machine learning problem on Cifar-10 dataset, with Resnet-18. The left and right figures correspond to the training loss and test accuracy, respectively.

We summarize the performances of the two methods in Section 8.2. We see that Algorithm 1 outperforms the vanilla method in terms of both training loss and test accuracy. In terms of the test accuracy, Algorithm 1 displays a much smoother curve than that of the vanilla method.

In connection to our theoretical observation that the step size parameter M should increase while the momentum parame-

ter γ should decrease as the stochastic level σ increases, we observe that, grid searches with respect to the train loss lead to the choices $M = 100, 100, 10$ and $\gamma = 0.1, 0.1, 0.1$ for batch sizes $64, 128$ and 256. It appears that the momentum parameter γ is less sensitive to the batch size than the step size parameter M, and setting $\gamma = 0.1$ might be a good choice in practice. We note that while there is certainly a correlation between the batch size and the stochastic level σ^2, we do not have a direct control over σ^2, as compared to the synthetic problem.

9. Conclusion

In this work we revisit the non-convex stochastic composite optimization problem, and address its convergence issue in the small batch regime. We show that the vanilla stochastic proximal method cannot converge to the stationary point beyond the variance of the gradient noise. We analyze the immensely successful Polyak momentum method in this context and establish its optimal convergence rate without any batch size requirement, demonstrating its superiority over the vanilla method. We conduct numerical experiments to corroborate our theoretical findings. In light of the past successes of proximal methods in ML, and the recent emerging application scenarios for proximal methods in DL our findings reinforce the robustness and the potential of the Polyak momentum method.

Impact Statement

This paper presents work that aims to advance the fields of optimization and Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Acknowledgments

References

Allen-Zhu, Z. How to make the gradients small stochastically: Even faster convex and nonconvex sgd. *Advances in Neural Information Processing Systems*, 31, 2018.

Arjevani, Y., Carmon, Y., Duchi, J. C., Foster, D. J., Srebro, N., and Woodworth, B. Lower bounds for non-convex stochastic optimization. *Mathematical Programming*, 199 (1-2):165–214, 2023.

Barré, M., Taylor, A. B., and Bach, F. Principled analyses and design of first-order methods with inexact proximal operators. *Mathematical Programming*, 201(1):185–230, 2023.

Bubeck, S. et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Chen, X., He, N., Hu, Y., and Ye, Z. Efficient algorithms for minimizing compositions of convex functions and random functions and its applications in network revenue management. *arXiv preprint arXiv:2205.01774*, 2022.

Combettes, P. L. and Pesquet, J.-C. Proximal splitting methods in signal processing, 2010.

Cutkosky, A. and Mehta, H. Momentum improves normalized sgd. In *International conference on machine learning*, pp. 2260–2268. PMLR, 2020.

Cutkosky, A. and Orabona, F. Momentum-based variance reduction in non-convex sgd. *Advances in neural information processing systems*, 32, 2019.

Davis, D. and Drusvyatskiy, D. Stochastic model-based minimization of weakly convex functions. *SIAM Journal on Optimization*, 29(1):207–239, 2019.

Doan, T., Maguluri, S., and Romberg, J. Finite-time analysis of distributed td (0) with linear function approximation on multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 1626–1635. PMLR, 2019.

Fang, C., Li, C. J., Lin, Z., and Zhang, T. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in neural information processing systems*, 31, 2018.

Fatkhullin, I., Tyurin, A., and Richtárik, P. Momentum provably improves error feedback! In *Advances in Neural Information Processing Systems*, 2023.

Ghadimi, S. and Lan, G. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization i: A generic algorithmic framework. *SIAM Journal on Optimization*, 22(4):1469–1492, 2012.

Ghadimi, S. and Lan, G. Accelerated gradient methods for nonconvex nonlinear and stochastic programming, 2013a.

Ghadimi, S. and Lan, G. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, ii: shrinking procedures and optimal algorithms. *SIAM Journal on Optimization*, 23(4):2061–2089, 2013b.

Ghadimi, S., Lan, G., and Zhang, H. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155(1): 267–305, 2016.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hendrikx, H., Xiao, L., Bubeck, S., Bach, F., and Massoulie, L. Statistically preconditioned accelerated gradient method for distributed optimization. In *International conference on machine learning*, pp. 4203–4227. PMLR, 2020.

Jelassi, S. and Li, Y. Towards understanding how momentum improves generalization in deep learning. In *International Conference on Machine Learning*, pp. 9965–10040. PMLR, 2022.

Jin, H., Peng, Y., Yang, W., Wang, S., and Zhang, Z. Federated reinforcement learning with environment heterogeneity. In *International Conference on Artificial Intelligence and Statistics*, pp. 18–37. PMLR, 2022.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. In *Proceedings of International Conference on Learning Representations*, 2017.

Khodadadian, S., Sharma, P., Joshi, G., and Maguluri, S. T. Federated reinforcement learning: Linear speedup under markovian sampling. In *International Conference on Machine Learning*, pp. 10997–11057. PMLR, 2022.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-10 dataset, 2014.

LeCun, Y., Bottou, L., Orr, G., and Muller, K.-R. Efficient backprop. *Neural Networks: Tricks of the Trade, Springer-Verlag*, 2012.

Li, X., Liu, M., and Orabona, F. On the last iterate convergence of momentum methods. In *International Conference on Algorithmic Learning Theory*, pp. 699–717. PMLR, 2022.

Liu, B., Wang, M., Foroosh, H., Tappen, M., and Pensky, M. Sparse convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

Liu, Y., Gao, Y., and Yin, W. An improved analysis of stochastic gradient descent with momentum. *Advances in Neural Information Processing Systems*, 33:18261–18271, 2020.

Luke, D. R. *Proximal Methods for Image Processing*, pp. 165–202. Springer International Publishing, Cham, 2020. ISBN 978-3-030-34413-9. doi: 10.1007/978-3-030-34413-9_6. URL https://doi.org/10.1007/978-3-030-34413-9_6.

Mishchenko, K., Malinovsky, G., Stich, S., and Richtárik, P. Proxskip: Yes! local gradient steps provably lead to communication acceleration! finally! In *International Conference on Machine Learning*, pp. 15750–15769. PMLR, 2022.

Murray, K., Kinnison, J., Nguyen, T. Q., Scheirer, W., and Chiang, D. Auto-sizing the transformer network: Improving speed, efficiency, and performance for low-resource machine translation. In *Proceedings of the Third Workshop on Neural Generation and Translation*, 2019.

Nemirovskij, A. S. and Yudin, D. B. *Problem complexity and method efficiency in optimization*. Wiley-Interscience, 1983.

Nesterov, Y. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Polyak, B. T. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964.

Rakhlin, A., Shamir, O., and Sridharan, K. Making gradient descent optimal for strongly convex stochastic optimization. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 1571–1578, 2012.

Rieke, N., Hancox, J., Li, W., Milletari, F., Roth, H. R., Albarqouni, S., Bakas, S., Galtier, M. N., Landman, B. A., Maier-Hein, K., et al. The future of digital health with federated learning. *NPJ digital medicine*, 3(1):119, 2020.

Sebbouh, O., Gower, R. M., and Defazio, A. Almost sure convergence rates for stochastic gradient descent and stochastic heavy ball. In *Conference on Learning Theory*, pp. 3935–3971. PMLR, 2021.

Sekhari, A., Sridharan, K., and Kale, S. Sgd: The role of implicit regularization, batch-size and multiple-epochs. *Advances In Neural Information Processing Systems*, 34:27422–27433, 2021.

Tran-Dinh, Q., Pham, N. H., Phan, D. T., and Nguyen, L. M. A hybrid stochastic optimization framework for composite nonconvex optimization. *Mathematical Programming*, 191(2):1005–1071, 2022.

Wang, J.-K. and Abernethy, J. Quickly finding a benign region via heavy ball momentum in non-convex optimization. *arXiv preprint arXiv:2010.01449*, 2020.

Wang, Z., Ji, K., Zhou, Y., Liang, Y., and Tarokh, V. Spiderboost and momentum: Faster variance reduction algorithms. *Advances in Neural Information Processing Systems*, 32, 2019.

Wilson, R. and Martinez, T. The general inefficiency of batch training for gradient descent learning. *Neural Networks*, 2003.

Woodworth, B., Mishchenko, K., and Bach, F. Two losses are better than one: Faster optimization using a cheaper proxy, 2023.

Xu, Y. and Xu, Y. Momentum-based variance-reduced proximal stochastic gradient method for composite nonconvex stochastic optimization, 2022.

Yu, H., Jin, R., and Yang, S. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. In *International Conference on Machine Learning*, pp. 7184–7193. PMLR, 2019.

Zhang, J., Lin, H., Jegelka, S., Sra, S., and Jadbabaie, A. Complexity of finding stationary points of nonconvex nonsmooth functions. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 11173–11182. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/zhang20p.html.
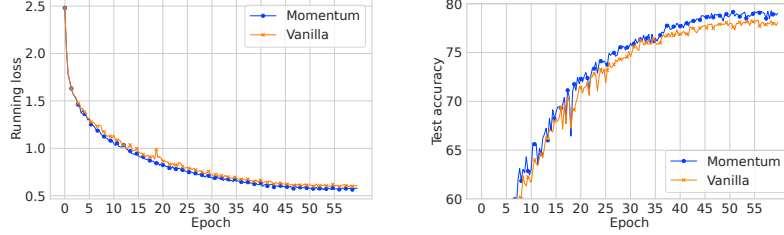
Figure 3: Comparison of Algorithm 1 and the vanilla stochastic proximal gradient method for the statistical preconditioning technique on Cifar-10 dataset. The left and right figures correspond to the training loss and test accuracy, respectively.

A. Additional Experiments

In this section we consider the recent statistical preconditioner (proxy training) technique of (Hendrikx et al., 2020; Woodworth et al., 2023), where for some objective ℓ, we consider $f = \ell - \hat{\ell}$ and $\psi = \hat{\ell}$. $\hat{\ell}$ is a "statistical preconditioner" defined on a sub-sample of the training dataset. Here we simulate the setup on Cifar-10 dataset (Krizhevsky et al., 2014). ℓ is defined on the whole 50000 training images, and $\hat{\ell}$ is defined on a subset of 2560 training images. We follow the implementation of (Woodworth et al., 2023), with one difference: at each iteration k, (Woodworth et al., 2023) computes the full gradient $\nabla \hat{\ell}(\mathbf{x}_k)$ while we only compute a stochastic gradient of batch size 128. In the experiment, we use batch size of 512 for ℓ, and a batch size of 128 for the SGD updates on Ω_k. We perform a grid-search on the parameters M and γ. The SGD on Ω_k takes 20 iterations and a step-size 0.01, which is tuned in (Woodworth et al., 2023). The experiments demonstrate that SGD is effective and reliable for solving the proximal step with sufficient accuracy (see also the experiments in (Woodworth et al., 2023)). We see that the momentum method outperforms the vanilla method, and the convergence of the momentum method seems smoother than the vanilla method.

B. Missing Proofs in Section 5

We start by giving the proof of Lemma 5.2:

Lemma 5.2. *Under Assumption 3.4, for any $k \geq 0$:*

$$\Delta_{k+1} \leq (1 - \gamma_k)\Delta_k + \frac{L^2}{\gamma_k}R_k + \gamma_k^2\sigma^2 .$$

Proof. Indeed,

$$\Delta_{k+1} = \mathbb{E}[\|\mathbf{m}_{k+1} - \nabla f(\mathbf{x}_{k+1})\|^2] = \mathbb{E}[\|(1-\gamma_k)(\mathbf{m}_k - \nabla f(\mathbf{x}_{k+1})) + \gamma_k(\mathbf{g}_{k+1} - \nabla f(\mathbf{x}_{k+1}))\|^2]$$
$$= (1-\gamma_k)^2\mathbb{E}[\|\mathbf{m}_k - \nabla f(\mathbf{x}_{k+1})\|^2] + \mathbb{E}[\gamma_k^2\|\mathbf{g}_{k+1} - \nabla f(\mathbf{x}_{k+1})\|^2]$$
$$\leq (1-\gamma_k)^2\mathbb{E}[\|\mathbf{m}_k - \nabla f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k+1})\|^2] + \gamma_k^2\sigma^2.$$

For the second identity, we have used the fact that $\mathbf{g}_{k+1}$ is unbiased. By Young's inequality, for any $\alpha > 0$, we have:

$$\Delta_{k+1} \leq (1-\gamma_k)^2(1+\alpha)\Delta_k + (1-\gamma_k)^2(1+\alpha^{-1})\mathbb{E}[\|\nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k+1})\|^2] + \gamma_k^2\sigma^2$$
$$\leq (1-\gamma_k)^2(1+\alpha)\Delta_k + (1-\gamma_k)^2(1+\alpha^{-1})L^2 R_k + \gamma_k^2\sigma^2,$$

where the last inequality follows from the smoothness of f. Choosing now $\alpha := \frac{\gamma_k}{1-\gamma_k}$, we get the claimed inequality. $\square$

Now we give the proof of Lemma 5.6:

Lemma 5.6. *Let Assumptions 3.1, 3.3, 3.4 and 5.1 hold, and let $a := \frac{3}{8L}$, $\gamma_k := \frac{3L}{M_k - L}$ and $M_k > 4L$ for any $k \geq 0$. Then, for any $k \geq 0$,*

$$\Phi_{k+1} \leq \Phi_k - \frac{1}{48M_k}\mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2] + \frac{27L\sigma^2}{4M_k^2}. \tag{7}$$

12

Proof. We put together Lemmas 5.2 and 5.3 and get:

$$\Phi_{k+1} = F_{k+1} + a\Delta_{k+1} \le F_k - \frac{M_k - L}{4}R_k + \frac{\Delta_k}{M_k - L} + a\left[(1 - \gamma_k)\Delta_k + \frac{L^2}{\gamma_k}R_k + \gamma_k^2\sigma^2\right]$$

$$= F_k - H_k R_k + a\gamma_k^2\sigma^2 + \left(1 - \gamma_k + \frac{1}{a(M_k - L)}\right)a\Delta_k,$$

where $H_k := \frac{M_k - L}{4} - \frac{aL^2}{\gamma_k}$. If $H_k > 0$, then we can plug Lemma 5.5 in and get:

$$\Phi_{k+1} \le F_k - \frac{H_k}{M_k^2 + L^2}\left(\frac{1}{3}\mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2] - \Delta_k\right) + \left(1 - \gamma_k + \frac{1}{a(M_k - L)}\right)a\Delta_k + a\gamma_k^2\sigma^2$$

$$= F_k - \frac{H_k}{3(M_k^2 + L^2)}\mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2] + \left(1 - \gamma_k + \frac{1}{a(M_k - L)} + \frac{H_k}{a(M_k^2 + L^2)}\right)a\Delta_k + a\gamma_k^2\sigma^2.$$

Now let us choose $a = \frac{\gamma_k(M_k - L)}{8L^2}$, so that $H_k = \frac{M_k - L}{8} > 0$. Then,

$$\frac{1}{a(M_k - L)} + \frac{H_k}{a(M_k^2 + L^2)} = \frac{8L^2}{\gamma_k(M_k - L)^2} + \frac{L^2}{\gamma_k(M_k^2 + L^2)} = \frac{L^2}{\gamma_k}\left(\frac{8}{(M_k - L)^2} + \frac{1}{M_k^2 + L^2}\right).$$

Therefore we need $\frac{L^2}{\gamma_k}\left(\frac{8}{(M_k - L)^2} + \frac{1}{M_k^2 + L^2}\right) \le \gamma_k$ or, equivalently, $\gamma_k^2 \ge L^2\left(\frac{8}{(M_k - L)^2} + \frac{1}{M_k^2 + L^2}\right)$. Since $(M_k - L)^2 \le M_k^2 + L^2$, it suffices to set $\gamma_k^2 = \frac{9L^2}{(M_k - L)^2}$, i.e., $\gamma_k = \frac{3L}{M_k - L}$. Note that this requires $M_k > 4L$ as we want to keep $\gamma_k < 1$. For our value of γ_k, we get $a = \frac{3}{8L}$. Putting everything together, we obtain

$$\Phi_{k+1} \le \Phi_k - \frac{M_k - L}{24(M_k^2 + L^2)}\mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2] + \frac{27L\sigma^2}{8(M_k - L)^2}.$$

Since $\frac{M_k - L}{M_k^2 + L^2} \ge \frac{1}{2(M_k - L)} \ge \frac{1}{2M_k}$, we can further estimate

$$\Phi_{k+1} \le \Phi_k - \frac{1}{48M_k}\mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2] + \frac{27L\sigma^2}{4M_k^2}. \qquad \square$$

Corollary 5.7. *Let Algorithm 1 be run for $K \ge 1$ iterations for solving problem* (1) *under Assumptions 3.3, 3.4 and 5.1, with constant coefficients $M_k = M = 4L + \frac{3^{3/2}}{2}\sqrt{\frac{KL\sigma^2}{\Phi_0}}$ and $\gamma_k = \frac{3L}{M - L}$ for any $0 \le k \le K - 1$, where $\Phi_0 := F(\mathbf{x}_0) - F^* + \frac{3}{8L}\mathbb{E}[\|\mathbf{m}_0 - \nabla f(\mathbf{x}_0)\|^2]$. Then,*

$$\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \le 48(3^{3/2})\sqrt{\frac{L\Phi_0\sigma^2}{K}} + \frac{192L\Phi_0}{K}, \tag{8}$$

where t is chosen uniformly at random from $\{1, \ldots, K\}$.

Proof. Setting $M_k := M$ some constant, rearranging and summing Equation (7) from 0 to $K - 1$, we get

$$\frac{1}{48M}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2] \le \Phi_0 + \frac{27KL\sigma^2}{4M^2}$$

Therefore

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2] \le \frac{48M\Phi_0}{K} + \frac{324L\sigma^2}{M}$$

Recall that $M > 4L$, then for $M = 4L + \frac{3^{3/2}}{2}\sqrt{\frac{KL\sigma^2}{\Phi_0}}$. We have:

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2] \le \frac{192L\Phi_0}{K} + 48(3^{3/2})\sqrt{\frac{L\Phi_0\sigma^2}{K}}$$

Therefore, after at most $\mathcal{O}(\frac{L\Phi_0\sigma^2}{\varepsilon^2} + \frac{L\Phi_0}{\varepsilon})$ iterations, we have $\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2] \leq \varepsilon$.

If t is chosen uniformly at random from $\{1,\ldots,K\}$, then $\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] = \frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2]$. $\qquad\square$

We also discuss the case where M_k is not a constant:

Corollary 5.8. *Consider Algorithm* 1 *for solving problem* (1) *under Assumptions* 3.3, 3.4 *and* 5.1 *with coefficients* $M_k = \max\{\sqrt{\frac{(k+1)L\sigma^2}{\Phi_0}}, 4L\}$, $\gamma_k = \frac{3L}{M_k - L}$ *for any* $k \geq 0$, *where* $\Phi_0 := F(\mathbf{x}_0) - F^* + \frac{3}{8L}\mathbb{E}[\|\mathbf{m}_0 - \nabla f(\mathbf{x}_0)\|^2]$. *Then, for any* $k \geq 1$, *we have*

$$\mathbb{E}[\|\nabla F(\mathbf{x}_{t(k)})\|^2] \leq 372\ln(ek)\Big(\sqrt{\frac{L\Phi_0\sigma^2}{k}} + \frac{L\Phi_0}{k}\Big), \tag{9}$$

where $t(k)$ *is chosen randomly from* $\{1,\ldots,k\}$ *with probabilities* $\Pr(t(k) = i) \propto \frac{1}{M_{i-1}}$, $i = 1,\ldots,k$, *and* $e := \exp(1)$.

Proof. Denote $G_{k+1}^2 := \mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2]$. According to Lemma 5.6, we have

$$\Phi_{k+1} \leq \Phi_k - \frac{1}{48M_k}G_{k+1}^2 + \frac{27L\sigma^2}{4M_k^2}.$$

Rearranging and summing the above from 0 to $K-1$, where $K \geq 1$ is arbitrary, we get:

$$\frac{1}{48}\sum_{k=0}^{K-1}\frac{1}{M_k}G_{k+1}^2 \leq \Phi_0 + \frac{27L\sigma^2}{4}\sum_{k=0}^{K-1}\frac{1}{M_k^2}.$$

Denoting $A_i = \sum_{k=0}^{i-1}\frac{1}{M_k}$, we obtain

$$\frac{1}{A_K}\sum_{k=0}^{K-1}\frac{1}{M_k}G_{k+1}^2 \leq \frac{48\Phi_0 + 324L\sigma^2\sum_{k=0}^{K-1}\frac{1}{M_k^2}}{A_K},$$

Hence, for $M_k := \max\{\sqrt{\frac{(k+1)L\sigma^2}{\phi_0}}, 4L\}$, we have

$$A_K = \sum_{k=1}^{K}\min\Big\{\sqrt{\frac{\Phi_0}{kL\sigma^2}}, \frac{1}{4L}\Big\} \geq K\min\Big\{\sqrt{\frac{\Phi_0}{KL\sigma^2}}, \frac{1}{4L}\Big\} = \min\Big\{\sqrt{\frac{K\Phi_0}{L\sigma^2}}, \frac{K}{4L}\Big\}$$

and

$$48\Phi_0 + 324L\sigma^2\sum_{k=0}^{K-1}\frac{1}{M_k^2} \leq 48\Phi_0 + 324L\sigma^2\sum_{k=1}^{K}\frac{\Phi_0}{kL\sigma^2} \leq 372\Phi_0\sum_{k=1}^{K}\frac{1}{k}.$$

Putting everything together, we get:

$$\frac{1}{A_K}\sum_{k=0}^{K-1}\frac{1}{M_k}G_{k+1}^2 \leq \frac{372\Phi_0\sum_{k=1}^{K-1}\frac{1}{k}}{\min\{\sqrt{\frac{K\Phi_0}{L\sigma^2}}, \frac{K}{4L}\}} \leq \frac{372\Phi_0(\ln K + 1)}{\min\{\sqrt{\frac{K\Phi_0}{L\sigma^2}}, \frac{K}{4L}\}} \leq 372(\ln K + 1)\Big(\sqrt{\frac{L\Phi_0\sigma^2}{K}} + \frac{L\Phi_0}{K}\Big).$$

If $t(k)$ is chosen from $\{1,\ldots,k\}$ with probabilities $\Pr(t(k) = i) = \frac{1}{M_iA_k}$, then $\mathbb{E}[\nabla F(\mathbf{x}_{t(k)})] = \frac{1}{A_k}\sum_{i=0}^{k-1}\frac{G_{i+1}^2}{M_i}$. $\qquad\square$

C. Missing Proofs in Section 6

Lemma C.1. *Under Assumptions* 3.1, 3.3, 3.4 *and* 5.1, *for* $\{\mathbf{x}_k\}_{k>0}$ *generated by Algorithm* 1, *if* $\gamma_k := \frac{3\sqrt{2}L}{M_k - L}$, $a := \frac{\sqrt{2}}{8L}$, *and* $M_k > (1 + 3\sqrt{2})L$, *we have:*

$$\Phi_{k+1} \leq \Phi_k - \frac{1}{48M_k}\mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2] - \frac{3\sqrt{2}}{2M_k}\Delta_k + \frac{27\sqrt{2}L\sigma^2}{4M_k^2}.$$

Proof. We repeat the proof of Lemma 5.6 but now require that $\frac{L^2}{\gamma_k}\left(\frac{8}{(M_k-L)^2} + \frac{1}{M_k^2+L^2}\right) \leq \frac{\gamma_k}{2}$. To satisfy this inequality, it suffices to choose $\gamma_k = \frac{3\sqrt{2}L}{M_k-L}$, which requires that $M_k > (1+3\sqrt{2})L$. Now we have $a = \frac{\sqrt{2}}{8L}$ and:

$$\Phi_{k+1} \leq \Phi_k - \frac{M_k-L}{24(M_k^2+L^2)}\mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2] - \frac{3\sqrt{2}L}{2(M_k-L)}\Delta_k + \frac{27\sqrt{2}L\sigma^2}{4(M_k-L)^2}.$$

Let $M_k = \tau_k L$ for some $\tau_k > (1+3\sqrt{2})$. Note that, for $\tau_k > (1+3\sqrt{2})$, we have $\frac{\tau_k-1}{\tau_k^2+1} \geq \frac{1}{2\tau_k}$, $\frac{1}{2(\tau_k-1)} \geq \frac{1}{2\tau_k}$ and $\frac{1}{(\tau_k-1)^2} \leq \frac{2}{\tau_k^2}$. Therefore, for $M_k > (1+3\sqrt{2})L$:

$$\Phi_{k+1} \leq \Phi_k - \frac{1}{48M_k}\mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2] - \frac{3\sqrt{2}}{2M_k}\Delta_k + \frac{27\sqrt{2}L\sigma^2}{4M_k^2}. \qquad \square$$

D. Missing Proofs in Section 7

First, we notice that, under Assumptions 3.1 and 3.3, Lemma 5.3 still holds in the inexact case. Now we give an analogous result to Lemma 5.5 in the inexact case:

Lemma D.1. *Under Assumption 3.3, for $\{\mathbf{x}_k\}_{k>0}$ generated by Algorithm 1, we have:*

$$(M_k^2+L^2)R_k \geq \frac{1}{4}\mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2] - \Delta_k - \mathbb{E}[\|\nabla\Omega(\mathbf{x}_{k+1})\|^2].$$

Proof. Indeed,

$$\begin{aligned}
\|\nabla F(\mathbf{x}_{k+1})\|^2 &= \|\nabla f(\mathbf{x}_{k+1}) + \nabla\psi(\mathbf{x}_{k+1})\|^2 \\
&= \|\mathbf{m}_k + \nabla\psi(\mathbf{x}_{k+1}) + (\nabla f(\mathbf{x}_k) - \mathbf{m}_k) + (\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k))\|^2 \\
&= \|M_k(\mathbf{x}_k - \mathbf{x}_{k+1}) + \nabla\Omega(\mathbf{x}_{k+1}) + (\nabla f(\mathbf{x}_k) - \mathbf{m}_k) + (\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k))\|^2 \\
&\leq 4M_k^2\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + 4\|\nabla\Omega(\mathbf{x}_{k+1})\|^2 + 4\|\mathbf{m}_k - \nabla f(\mathbf{x}_k)\|^2 + 4\|\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)\|^2 \\
&\leq 4(M_k^2+L^2)\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + 4\|\mathbf{m}_k - \nabla f(\mathbf{x}_k)\|^2 + 4\|\nabla\Omega(\mathbf{x}_{k+1})\|^2.
\end{aligned}$$

Taking expectations and rearranging, we obtain the claim. $\qquad \square$

Notice that Lemma 5.2 still holds. We continue to use the same Lyapunov function $\Phi_k := F_k + a\Delta_k$.

Lemma D.2. *Under Assumptions 3.1, 3.3 and 3.4, for $\{\mathbf{x}_k\}_{k>0}$ generated by Algorithm 1, if $\gamma_k := \sqrt{\frac{456}{17}}\frac{L}{M_k-L}$, $a := \sqrt{\frac{19}{408}}\frac{1}{L}$, that in each iteration condition (11) holds. Then, for $M_k > 7L$, we have:*

$$\Phi_{k+1} \leq \Phi_k - \frac{1}{68M_k}\mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2] + \frac{12L\sigma^2}{M_k^2} + \frac{2S_k}{17M_k}. \tag{12}$$

Proof. Denote $G_{k+1}^2 := \mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2]$. Plugging Equation (11) into Lemma D.1, we obtain

$$(M_k^2+L^2)R_k \geq \frac{1}{4}G_{k+1}^2 - \Delta_k - \mathbb{E}[\|\nabla\Omega(\mathbf{x}_{k+1})\|^2] \geq \frac{1}{4}G_{k+1}^2 - \Delta_k - \frac{M_k^2}{16}R_k - S_k.$$

Rearranging, we get:

$$\frac{17}{16}(M_k^2+L^2)R_k \geq \frac{1}{4}G_{k+1}^2 - \Delta_k - S_k. \tag{13}$$

Recall in the proof of Lemma 5.6, we have:

$$\Phi_{k+1} \leq F_k - H_kR_k + \left(1 - \gamma_k + \frac{1}{a(M_k-L)}\right)a\Delta_k + a\gamma_k^2\sigma^2,$$

where $H_k := \frac{M_k - L}{4} - \frac{aL^2}{\gamma_k}$. If $H_k > 0$, then we can plug Equation (13) in and get:

$$\Phi_{k+1} \leq F_k - \frac{16 H_k}{17(M_k^2 + L^2)}\left(\frac{1}{4}G_{k+1}^2 - \Delta_k - S_k\right) + \left(1 - \gamma_k + \frac{1}{a(M_k - L)}\right)a\Delta_k + a\gamma_k^2\sigma^2$$

$$= F_k - \frac{4H_k}{17(M_k^2 + L^2)}G_{k+1}^2 + \left(1 - \gamma_k + \frac{1}{a(M_k - L)} + \frac{16 H_k}{17a(M_k^2 + L^2)}\right)a\Delta_k + a\gamma_k^2\sigma^2 + \frac{16 H_k S_k}{17(M_k^2 + L^2)}.$$

Now let $a = \frac{\gamma_k(M_k - L)}{8L^2}$, so that $H_k = \frac{M_k - L}{8}$. Then,

$$\frac{1}{a(M_k - L)} + \frac{16 H_k}{17a(M_k^2 + L^2)} = \frac{8L^2}{\gamma_k(M_k - L)^2} + \frac{16 L^2}{17\gamma_k(M_k^2 + L^2)} = \frac{8L^2}{\gamma_k}\left(\frac{1}{(M_k - L)^2} + \frac{2}{17(M_k^2 + L^2)}\right).$$

Therefore we need $\frac{8L^2}{\gamma_k}\left(\frac{1}{(M_k-L)^2} + \frac{2}{17(M_k^2+L^2)}\right) \leq \gamma_k$ or, equivalently, $\gamma_k^2 \geq 8L^2\left(\frac{1}{(M_k-L)^2} + \frac{2}{17(M_k^2+L^2)}\right)$. Since $(M_k - L)^2 \leq M_k^2 + L^2$, it suffices to set $\gamma_k^2 = \frac{152}{17}\frac{L^2}{(M_k-L)^2}$, i.e., $\gamma_k = \sqrt{\frac{152}{17}}\frac{L}{M_k - L}$. Note that this requires $M_k > \frac{(\sqrt{152}+\sqrt{17})L}{\sqrt{17}}$, which is ensured whenever $M_k > 4L$. For our choice of γ_k, we get $a = \frac{1}{8L}\sqrt{\frac{152}{17}} = \sqrt{\frac{19}{156}}\frac{1}{L}$. Putting everything together, we get

$$\Phi_{k+1} \leq \Phi_k - \frac{M_k - L}{34(M_k^2 + L^2)}G_{k+1}^2 + \frac{4L}{(M_k - L)^2}\sigma^2 + \frac{2(M_k - L)S_k}{17(M_k^2 + L^2)}.$$

Let $M_k = \tau_k L$ for some $\tau_k > 7$. Note that $\frac{\tau_k - 1}{\tau_k^2 + 1} \geq \frac{1}{2\tau_k}$, $\frac{1}{(\tau_k - 1)^2} \leq \frac{2}{\tau_k^2}$ and $\frac{\tau_k - 1}{\tau_k^2 + 1} \leq \frac{1}{\tau_k}$. Therefore, for $M_k > 7L$, we get

$$\Phi_{k+1} \leq \Phi_k - \frac{1}{68 M_k}G_{k+1}^2 + \frac{8L}{M_k^2}\sigma^2 + \frac{2S_k}{17 M_k}. \qquad \square$$

Setting $M_k = M$ some constant, and rearranging and summing Equation (12) over $k = 0, \ldots, K - 1$, we get:

$$\frac{1}{68M}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2] \leq \Phi_0 + \frac{8KL}{M^2}\sigma^2 + \frac{2}{17M}\sum_{k=0}^{K-1}S_k.$$

Therefore,

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2] \leq \frac{68M\Phi_0}{K} + \frac{544L\sigma^2}{M} + \frac{8}{K}\sum_{k=0}^{K-1}S_k.$$

Then for $M = 7L + \sqrt{\frac{8KL\sigma^2}{\Phi_0}}$, we have:

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla F(\mathbf{x}_{k+1})\|^2] \leq \mathcal{O}\left(\sqrt{\frac{L\Phi_0\sigma^2}{K}} + \frac{L\sigma^2}{K}\right) + \frac{8}{K}\sum_{k=0}^{K-1}S_k.$$

E. Sampling from a Stream of Data

Following Theorem 5.9, we briefly mentioned that one can efficiently sample the desired output point. In this section, we explain how to perform such sampling at no extra computation and memory cost. This might have been discussed in the literature, but we still provide a detailed explanation for completeness and the reader's convenience.

Proposition E.1. *Given a stream of points $\{\mathbf{x}_k\}_{k=1}^{\infty}$ in $\mathbb{R}^d$ and positive scalars $\{h_k\}_{k=1}^{\infty}$, we can maintain, at each step $k \geq 1$, the random variable $\mathbf{x}_{t(k)}$, where $t(k)$ is a random index from $\{1, \ldots, k\}$ chosen with probabilities $\Pr(t(k) = i) = \frac{h_i}{H_k}$, $i = 1, \ldots, k$, where $H_k := \sum_{i=1}^{k} h_i$. This requires only $\mathcal{O}(d)$ memory and computation.*

Proof. We maintain the variables $\hat{\mathbf{x}}_k \in \mathbb{R}^d$ and $H_k \in \mathbb{R}$ which are both initialized to 0 at step $k = 0$. Then, at each step $k \geq 1$, we update $H_k \leftarrow H_{k-1} + h_k$ and also, with probability $\frac{h_k}{H_k}$, we update $\hat{\mathbf{x}}_k \leftarrow \mathbf{x}_k$ (or, with probability $1 - \frac{h_k}{H_k}$, keep the old $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k-1}$). The memory and computation costs are $\mathcal{O}(d)$. Note that, for any $1 \leq i \leq k$, the event $\hat{\mathbf{x}}_k = \mathbf{x}_i$ happens iff $\hat{\mathbf{x}}$ was updated at step i and then not updated at each step $j = i + 1, \ldots, k$. Hence, for any $1 \leq i \leq k$, we have

$$\Pr(\hat{\mathbf{x}}_k = \mathbf{x}_i) = \frac{h_i}{H_i} \cdot \prod_{j=i+1}^{k}\left(1 - \frac{h_j}{H_j}\right) = \frac{h_i}{H_i} \cdot \prod_{j=i+1}^{k}\frac{H_{j-1}}{H_j} = \frac{h_i}{H_k}. \qquad \square$$

F. Convergence in the Non-Differentiable Case

In this section we briefly discuss the convergence of Algorithm 1 in the non-differentiable case. In this case we assume that ψ is convex. We assume that each subproblem is solved exactly, i.e. $\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} \Omega_k(\mathbf{x})$. In particular, this implies that there exists subgradient $\mathbf{g}_{k+1}^\psi \in \partial\psi(\mathbf{x}_{k+1})$ such that:

$$\mathbf{m}_k + \mathbf{g}_{k+1}^\psi + M(\mathbf{x}_{k+1} - \mathbf{x}_k) = \mathbf{0}$$

Therefore, for convenience, we define $\nabla\psi(\mathbf{x}_{k+1}) \coloneqq \mathbf{g}_{k+1}^\psi$ as the specific subgradient of ψ at $\mathbf{x}_{k+1}$ that we use. Similarly, we define $\nabla F(\mathbf{x}_{k+1}) \coloneqq \nabla f(\mathbf{x}_{k+1}) + \mathbf{g}_{k+1}^\psi = \nabla f(\mathbf{x}_{k+1}) + \nabla\psi(\mathbf{x}_{k+1})$ as the specific subgradient of F at $\mathbf{x}_{k+1}$ that we use. This way, all our previous proofs still hold, and we have the following:

Theorem F.1. *Under Assumptions 3.1, 3.3 and 3.4, for $\{\mathbf{x}_k\}_{k>0}$ generated by Algorithm 1, if $M_k = M$ some constant, $\gamma \coloneqq \frac{3L}{M-L}$, there exists $M = 4L + \frac{3^{3/2}}{2}\sqrt{\frac{KL\sigma^2}{\Phi_0}}$ such that $K = \mathcal{O}\left(\frac{L\Phi_0\sigma^2}{\varepsilon^2} + \frac{L\Phi_0}{\varepsilon}\right)$ iterations of Algorithm 1 is sufficient to achieve $\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|^2] \leq \varepsilon$, where t is chosen from $\{1, \ldots, K\}$ uniformly at random.*

We remark that this can be reformulated in terms of the distance between ∂F and $\mathbf{0}$: we have $\mathrm{dist}^2(\partial F(\mathbf{x}_t), 0) \leq \varepsilon$.

G. Convergence Criterias

In this section, we discuss the differences and connections between the convergence criterias used in the literature.

Proximal Gradient Mapping: The most popular convergenec criteria used in most of the earlier works on non-convex composite optimization is proximal gradient mapping (Ghadimi et al., 2016; Ghadimi & Lan, 2013a; Wang et al., 2019; Hendrikx et al., 2020; Tran-Dinh et al., 2022; Xu & Xu, 2022). Proximal gradient mapping is defined in a very general context, where we consider the constrained composite optimization problem:

$$\min_{\mathbf{x} \in X}\left[F(\mathbf{x}) \coloneqq f(\mathbf{x}) + \psi(\mathbf{x})\right],$$

with the 1-strongly convex mirror map $r : X \to \mathbb{R}$. For any vector $\mathbf{g}$ and $\mathbf{x}$, and scalar $M > 0$, define:

$$\mathbf{x}^+ \coloneqq \arg\min_{\mathbf{x}' \in X}\{\langle \mathbf{g}, \mathbf{x}' - \mathbf{x}\rangle + \psi(\mathbf{x}) + \beta_r(\mathbf{x}', \mathbf{x})\}$$

where $\beta_r(\mathbf{x}', \mathbf{x}) = r(\mathbf{x}') - r(\mathbf{x}) - \langle\nabla r(\mathbf{x}), \mathbf{x}' - \mathbf{x}\rangle$ is the Bregman divergence of r. Then the proximal gradient mapping is defined as:

$$P_X(\mathbf{x}, \mathbf{g}, M) \coloneqq M(\mathbf{x} - \mathbf{x}^+),$$

and in the literature, the convergence is studied in terms of $\|P_X(\mathbf{x}, \nabla f(\mathbf{x}), M)\|^2$. We point out that it is very easy to prove analogous versions of Lemmas 5.3 and 5.5 in terms of $\|P_X(\mathbf{x}, \nabla f(\mathbf{x}), M)\|^2$ (which implies that our results extend to the proximal gradient mapping case), but we omit the details here. Instead, we argue that the size of $\nabla F(\mathbf{x})$ is a more natural convergence criterion. Consider the simplest situation where ψ is convex and differentiable, $X = \mathbb{R}^d$, and the mirror map $r(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2$ (i.e. the Euclidean geometry). Then we have the usual stationarity condition for $\mathbf{x}^+$:

$$\nabla f(\mathbf{x}) + \nabla\psi(\mathbf{x}^+) + M(\mathbf{x}^+ - \mathbf{x}) = 0.$$

Therefore, we have:

$$P_X(\mathbf{x}, \nabla f(\mathbf{x}), M) = \nabla f(\mathbf{x}) + \nabla\psi(\mathbf{x}^+).$$

In other words, the proximal gradient mapping gives the gradient of f at the current point plus the gradient of ψ at the next point. In contrast, in our analysis, we directly consider the gradient of F at the next point, and we believe that this is more natural and intuitive.

Moreau Envelope: A closely related convergence criterion that was proposed to address the non-convergence problem of proximal gradient mapping of the earlier works is the Moreau envelope (Davis & Drusvyatskiy, 2019). In Euclidean geometry, for some parameter λ, the Moreau envelope is defined as the following:

$$F_\lambda(\mathbf{x}) \coloneqq \min_{\mathbf{y}}\left\{F(\mathbf{y}) + \frac{1}{2\lambda}\|\mathbf{y} - \mathbf{x}\|^2\right\},$$

and write $\hat{\mathbf{x}} = \arg\min_{\mathbf{y}}\{F(\mathbf{y}) + \frac{1}{2\lambda}\|\mathbf{y} - \mathbf{x}\|^2\}$. The convergence criteria is then defined as:

$$\nabla F_\lambda(\mathbf{x}) \coloneqq \lambda^{-1}(\mathbf{x} - \hat{\mathbf{x}}).$$

Equivalently, $\nabla F_\lambda(\mathbf{x}) = \nabla F(\hat{\mathbf{x}})$ in the differentiable case. In other words, the convergence criteria with Moreau envelope uses a surrogate point $\hat{\mathbf{x}}$ instead of the actual iterates of the algorithm. Note that, the convergence criteria using Moreau envelop and proximal gradient mapping is with a constant factor of each other for a ρ-weakly convex function (Davis & Drusvyatskiy, 2019):

$$\frac{1}{4}\|\nabla F_{1/2\rho}(\mathbf{x})\| \leq \|P(\mathbf{x}, \nabla f(\mathbf{x}), \rho)\| \leq \frac{3}{2}\left(1 + \frac{1}{\sqrt{2}}\right)\|\nabla F_{1/2\rho}(\mathbf{x})\|.$$