UNDERSTANDING IMPLICIT AVERAGING IN SCHEDULE-FREE STOCHASTIC GRADIENT DESCENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Schedule-Free SGD (SF-SGD) eliminates the need for predefined learning-rate schedules by combining momentum updates with implicit averaging. A distinctive feature of SF-SGD is that it generates two iterates: the gradient-location point $\boldsymbol{y}_t$, where the stochastic gradient is evaluated, and the evaluation point $\boldsymbol{x}_t$, which implicitly aggregates information from past gradient-location iterates. Despite the practical importance of this averaging mechanism, its effect on the relationship between these two points remains insufficiently understood. We study SF-SGD under nonconvex objectives with constant averaging $c_t = c$ and establish convergence guarantees for both $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$, together with an upper bound on their gradient-norm gap. In particular, we show that the upper bound at $\boldsymbol{x}_t$ can be smaller than that at $\boldsymbol{y}_t$, providing theoretical evidence that implicit weighted averaging can improve the stationarity of the evaluation point. We also derive sufficient conditions characterizing the admissible ranges of the learning rate, momentum parameter, and averaging coefficient and extend our analysis to mini-batch stochastic gradients without assuming bounded gradients. Numerical experiments show that constant averaging can outperform uniform averaging in terms of gradient norms, gradient-norm gaps, and test accuracy while being more robust to the momentum parameter. We also observe advantages of increasing over fixed batch sizes under both matched training-sample budgets and equal numbers of optimization steps. The code is available at `https://anonymous.4open.science/r/schedule_free_convergence-CBAD`, which also includes additional experimental results.

1 INTRODUCTION

Stochastic Gradient Descent (SGD) (Robbins & Monro, 1951) and its variants are fundamental optimization methods for large-scale deep learning, with applications ranging from image classification to natural language processing (Krizhevsky et al., 2012; He et al., 2016; Vaswani et al., 2017; Bottou et al., 2018). Their performance, however, is highly sensitive to hyperparameter choices, particularly the learning rate and batch size. Learning-rate scheduling is therefore widely used in practice, with common choices including cosine annealing (Loshchilov & Hutter, 2017) and step decay (Lu, 2024). Such schedulers often depend on the total training horizon T, which may be difficult to specify in advance in settings involving early stopping, fine-tuning, and continual learning.

To reduce this dependence, Defazio et al. (2024) introduced Schedule-Free optimizers, which avoid predefined learning-rate schedules while achieving performance comparable to, and sometimes better than, carefully tuned schedulers across various deep learning tasks (Morwani et al., 2024; Song et al., 2025; Sampson & Melchior, 2025). Schedule-Free SGD (SF-SGD) combines momentum updates with averaging, as described in Algorithm1. In addition to the original choice $c_t = 1/(t+1)$ of Defazio et al. (2024), the constant averaging coefficient $c_t = c$ has been considered in previous work (Ahn et al., 2025). In our analysis, we introduce an arbitrary parameter $\epsilon \in (0, 1)$ and consider $c \in (\epsilon, 1)$, where ϵ is a lower bound on the constant averaging coefficient. The momentum parameter $\beta \in [0, 1]$ controls the interpolation between Polyak–Ruppert averaging ($\beta = 0$) (Polyak, 1990; Ruppert, 1988) and primal averaging ($\beta = 1$) (Nesterov & Shikhman, 2015; Tao et al., 2018; Cutkosky, 2019; Defazio & Jelassi, 2022). For the averaging coefficient, $c_t = 1/(t+1)$ is the original choice of Defazio et al. (2024), while recent studies have explored alternatives (Ahn et al., 2025;

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Algorithm 1 Schedule-Free SGD (Defazio et al., 2024, (3)–(5))

Require: Initial point $\boldsymbol{x}_0 = \boldsymbol{z}_0$, step size $\eta > 0$, batch size $b_t > 0$, momentum parameter $\beta \in [0, 1]$, arbitrary lower bound $\epsilon \in (0, 1)$, averaging coefficient $c_t = c \in (\epsilon, 1) \vee c_t = \frac{1}{t+1}$, the number of steps T

Ensure: $\boldsymbol{x}_T$

1: **for** $t = 0$ to $T - 1$ **do**
2: $\quad \boldsymbol{y}_t \coloneqq (1 - \beta)\boldsymbol{z}_t + \beta \boldsymbol{x}_t$
3: $\quad \boldsymbol{z}_{t+1} \coloneqq \boldsymbol{z}_t - \eta \nabla f_{B_t}(\boldsymbol{y}_t)$
4: $\quad \boldsymbol{x}_{t+1} \coloneqq (1 - c_{t+1})\boldsymbol{x}_t + c_{t+1}\boldsymbol{z}_{t+1}$
5: **end for**

Song et al., 2025). Thus, the role of averaging in SF-SGD remains incompletely understood both empirically and theoretically.

A distinctive feature of SF-SGD is that it generates two important sequences. The point $\boldsymbol{y}_t$ is referred to as the *gradient-location point*, since the stochastic gradient is evaluated at $\boldsymbol{y}_t$, whereas $\boldsymbol{x}_t$ is the *evaluation point*, which is used as the primary output of the algorithm (Defazio et al., 2024). Eliminating $\boldsymbol{z}_t$ from the update equations expresses $\boldsymbol{x}_t$ as a weighted average of its previous value and the current gradient-location point (Song et al., 2025):

$$\boldsymbol{x}_t = \frac{(1 - c_t)(1 - \beta)}{(1 - c_t)(1 - \beta) + c_t} \boldsymbol{x}_{t-1} + \frac{c_t}{(1 - c_t)(1 - \beta) + c_t} \boldsymbol{y}_t. \tag{1}$$

In particular, when $c_t = c$, recursively expanding (1) shows that $\boldsymbol{x}_t$ is an exponentially weighted average (EWA) of past gradient-location iterates (see Section 3). We focus on this implicit exponential averaging structure and investigate how it affects the relationship between $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$. Specifically, $\boldsymbol{x}_t$ accumulates information from past gradient-location iterates, whereas $\boldsymbol{y}_t$ is the current point where the stochastic gradient is evaluated. The gap between $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ therefore reflects how different the current gradient-location iterate is from the information accumulated through implicit weighted averaging. Understanding this gap provides a direct way to characterize the effect of the averaging mechanism intrinsic to SF-SGD.

Despite the central role of these two sequences, their convergence behavior and the relationship between them are not yet fully understood under standard nonconvex assumptions. In particular, existing analyses provide limited insight into both $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ in genuinely Schedule-Free parameter regimes that cannot be reduced to standard optimization methods. As recent work also notes (Chae & Kim, 2026), a theoretical gap remains between convergence analyses at the gradient-location and evaluation points. Bridging this gap and clarifying the effect of implicit weighted averaging are the main goals of this work.

We establish convergence guarantees for both $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ and quantify the gap between their gradient norms, thereby characterizing the effect of implicit weighted averaging in SF-SGD under constant averaging. We also extend the analysis to mini-batch stochastic gradients without imposing bounded-gradient assumptions. This extension reflects practical training, where mini-batches are standard (Brown et al., 2020; Hoffmann et al., 2022; Chowdhery et al., 2023; Llama Team, 2024) and uniformly bounded stochastic gradients can be restrictive (Bengio et al., 1994; Pascanu et al., 2013; Nguyen et al., 2018; Nishida et al., 2024). Our numerical experiments complement the theory by examining how the implicit averaging behavior depends on the averaging coefficient, batch-size setting, and momentum parameter β.

1.1 Previous Works

We compare the theoretical conditions of previous works with ours. Table 1 summarizes the comparison.

Objective Function. Defazio et al. (2024) considered convex objectives rather than the L-smooth nonconvex setting. Since modern neural-network objectives are generally nonconvex (Bottou et al., 2018), subsequent works (Ahn et al., 2025; Brown et al., 2025; Chae & Kim, 2026), including ours, study SF-SGD under nonconvex objectives. Among these studies, the well-behavedness condition of Ahn et al. (2025) is weaker than the L-smoothness assumed by Brown et al. (2025), Chae & Kim

Table 1: Analysis of SF-SGD **(1)** Defazio et al. (2024, Theorem 2), **(2)** Ahn et al. (2025, Corollary 5.1), **(3)** Brown et al. (2025, Theorem 3, Theorem 5), and **(4)** Chae & Kim (2026, Theorem 4.4). "Bounded gradient" means that $G := \sup_{\boldsymbol{x}\in\mathbb{R}^d} \|\nabla f(\boldsymbol{x})\| < \infty$. "Well-behaved" guarantees the fundamental path integral identity $f(\boldsymbol{x}) - f(\boldsymbol{w}) = \int_0^1 \langle \nabla f(\boldsymbol{w} + t(\boldsymbol{x} - \boldsymbol{w})), \boldsymbol{x} - \boldsymbol{w} \rangle dt$ for any two points $\boldsymbol{x}$ and $\boldsymbol{w}$. "—" implies that a mini-batch gradient is not used. We denote $G^2(\cdot_T) := \min_{t\in[0:T-1]} \mathbb{E}[\|\nabla f(\cdot_t)\|^2]$. "$(\lambda, \varepsilon)$-Stationarity" is a broader notion than standard stationarity; an ε-stationary point becomes a (λ, ε)-stationary point. Note that ϵ, the lower bound on c, is an arbitrary constant in $(0, 1)$ rather than the approximation accuracy.

Ref.	Assumption	Batch Size	Parameter	Convergence Rate
(1)	Convex Bounded gradient	—	$c_t = (t+1)^{-1}$ $\beta \in [0, 1]$	$\mathbb{E}[f(\boldsymbol{x}_T) - f(\boldsymbol{x}^*)] = O\left(\frac{1}{T} + \eta G^2\right)$
(2)	Nonconvex Well-behaved Bounded gradient	—	$\eta > 0$ $c = 1 - \zeta$ $\beta_t \sim \mathrm{Unif}[\zeta, 1]$ $(\zeta \in (0, 1))$	$\Omega(\lambda^{1/2}\varepsilon^{-7/2})$-SFO complexity to achieve (λ,ε)-Stationarity
(3)	Nonconvex L-smooth	—	$\eta \in (0, 1/L]$ $c_t = (t+1)^{-1}$ $\beta = 1$	$G^2(\boldsymbol{x}_T) = O\left(\frac{1}{\log T} + 2\sigma^2\right)$
(3)	Nonconvex L-smooth	—	$\eta \in (0, 1/L]$ $c_t = (t+1)^{-\alpha}$ $(\alpha \in [0, 1))$ $\beta = 1$	$G^2(\boldsymbol{x}_T) = O\left(\frac{1}{T^{1-\alpha}} + 2\sigma^2\right)$
(4)	Nonconvex L-smooth	—	$\eta \in (0, 1/L)$ $c_t = (t+1)^{-1}$ $\beta \in [0, 1)$	$G^2(\boldsymbol{y}_T) = O\left(\frac{\log T}{T} + \frac{1}{\eta T} + \eta(1-\beta)L\sigma^2\right)$
Ours	Nonconvex L-smooth	$b \in \mathbb{N}$	$\eta \in (0, \delta(\epsilon, \beta))$ $c \in (\epsilon, 1)$ $\beta \in [0, 1]$	$G^2(\boldsymbol{x}_T) = O\left(\frac{1}{T} + \frac{\sigma^2}{b}\right)$
Ours	Nonconvex L-smooth	$b \in \mathbb{N}$	$\eta \in (0, \delta(\epsilon, \beta))$ $c \in (\epsilon, 1)$ $\beta \in [0, 1]$	$G^2(\boldsymbol{y}_T) = O\left(\frac{1}{T} + 2\frac{\sigma^2}{b}\right)$

(2026), and our work, although their analysis also requires bounded gradients. Thus, Ahn et al. (2025) allows a broader smoothness condition, whereas Brown et al. (2025), Chae & Kim (2026), and our work consider the standard nonconvex optimization setting without bounded-gradient assumptions.

Parameters β and c_t. (I) When $c_t = 1$, SF-SGD reduces to standard SGD; (II) when $c_t \in (0, 1)$ and $\beta = 1$, it reduces to the heavy-ball method (Polyak, 1964); and (III) when $c_t \in (0, 1)$ and $\beta \in [0, 1)$, it follows the update rule specific to SF-SGD. Thus, a central challenge is to establish convergence guarantees for case (III). Defazio et al. (2024) analyzed cases (I)–(III). Ahn et al. (2025) considered cases (II) and (III) with constant averaging in a general online-to-nonconvex conversion framework, treating SF-SGD as a particular instance and analyzing the gradient-location point $\boldsymbol{y}_t$. In contrast, we directly analyze SF-SGD at both $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ and compare the two points to characterize the effect of implicit averaging. Brown et al. (2025) considered cases (I) and (II); although their analysis with a decreasing momentum parameter is novel, it does not cover the SF-SGD-specific case (III). Chae & Kim (2026) considered cases (I) and (III).

Convergence Rate. The original analysis of Defazio et al. (2024) established a finite-horizon rate of $O(1/\sqrt{T})$, while their Schedule-Free convergence bound is of the form $O(1/(\eta T) + \eta G^2)$. Ahn et al. (2025) established an SFO complexity of $\Omega(\lambda^{1/2}\varepsilon^{-7/2})$ for (λ, ε)-stationarity. Under L-smoothness, an ε-stationary point can be related to $(2L^2/\varepsilon, \varepsilon/2)$-stationarity, yielding an $\Omega(\varepsilon^{-4})$ SFO complexity consistent with that implied by our rate $O(1/T + \sigma^2/b)$. The difference is that Ahn et al. (2025) considers online learning, requiring $T = \Omega(\varepsilon^{-4})$, whereas our mini-batch setting allows $T = \Omega(\varepsilon^{-2})$

and $b = \Omega(\varepsilon^{-2})$. An interesting and natural difference between online and mini-batch learning appears at this point. Brown et al. (2025) obtained $O(1/T^{1-\alpha} + 2\sigma^2)$ for $c_t = 1/(t+1)^\alpha$, whose best case, $\alpha = 0$, gives $O(1/T + 2\sigma^2)$. Our rate $O(1/T + \sigma^2/b)$ improves this bound when $b = 1$. Compared with Chae & Kim (2026), our bound has a better convergence term, whereas their noise floor is smaller for $\eta \in (0, 1/L)$ and $b = 1$. Nevertheless, both results imply the same SFO complexity of $\Omega(\varepsilon^{-4})$. Importantly, our analysis establishes this complexity for both the gradient-location point $\boldsymbol{y}_t$ and the evaluation point $\boldsymbol{x}_t$.

1.2 Contributions

Based on Sections 1, 3, and 4, we summarize our contributions as follows.
(1) Focusing on the constant averaging setting $c_t = c$ in SF-SGD, we derive upper bounds on the gradient norms at both $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$, as well as on their gradient-norm gap. In particular, we show that the upper bound at $\boldsymbol{x}_t$ is smaller than that at $\boldsymbol{y}_t$, theoretically demonstrating the beneficial effect of implicit weighted averaging on the stationarity of the evaluation point. To the best of our knowledge, this is the first theoretical characterization of such an effect in SF-SGD with constant averaging.
(2) We introduce an arbitrary lower-bound parameter $\epsilon \in (0, 1)$ for the constant averaging coefficient $c \in (\epsilon, 1)$ and derive explicit sufficient conditions for convergence. Our analysis shows that the admissible range of the learning rate η becomes wider as the momentum parameter β increases or as ϵ increases, thereby clarifying how these algorithmic parameters affect the admissible learning-rate range.
(3) We conduct numerical experiments to investigate the effects of the averaging coefficient and batch-size strategy in SF-SGD. Our results show that constant averaging can outperform uniform averaging ($c_t = 1/(t+1)$), achieving smaller gradient norms and gradient-norm gaps and higher test accuracy in several settings. Increasing batch sizes also outperform fixed batch sizes under matched training-sample budgets and the same number of optimization steps. Moreover, constant averaging is less sensitive to the momentum parameter β than uniform averaging.

2 Problem Setting and Preliminaries

Let $\mathbb{N}$ be the set of natural numbers. For $n \in \mathbb{N}$, denote $[n] := \{1, 2, \ldots, n\}$ and $[0 : n] := \{0, 1, \ldots, n\}$. $\mathbb{R}^d$ is equipped with $\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^\top \boldsymbol{y}$, which induces $\|\boldsymbol{x}\| := \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$. Let $C^1(\mathbb{R}^d)$ be the set of continuously differentiable $\mathbb{R}$-valued functions on $\mathbb{R}^d$. Let Π be a probability distribution on $[n]$. For $\boldsymbol{x} \in \mathbb{R}^d$, a measurable function $h_\cdot(\boldsymbol{x}) : [n] \ni i \mapsto h_i(\boldsymbol{x}) \in \mathbb{R}^d$, and $\xi \sim \Pi$, the expectation is defined as $\mathbb{E}_\xi[h_\xi(\boldsymbol{x})] := \mathbb{E}_{\xi \sim \Pi}[h_\xi(\boldsymbol{x})] := \sum_{i=1}^n h_i(\boldsymbol{x}) \Pi(\{i\})$. For a training dataset $S := \{(\boldsymbol{w}_i, \boldsymbol{y}_i)\}_{i=1}^n$, the empirical risk minimization problem is formulated as

$$\text{minimize } f(\boldsymbol{x}) := \frac{1}{n} \sum_{i=1}^n f_i(\boldsymbol{x}) := \frac{1}{n} \sum_{i=1}^n \ell(\boldsymbol{x}; (\boldsymbol{w}_i, \boldsymbol{y}_i)) \text{ subject to } \boldsymbol{x} \in \mathbb{R}^d,$$

where ℓ is a loss function and f_i is the loss function with the i-th data point $(\boldsymbol{w}_i, \boldsymbol{y}_i) \in S$. For instance, this condition is satisfied by taking ℓ as the cross entropy loss. We impose the following standard assumptions (Ghadimi & Lan, 2013; Arjevani et al., 2023) (see Appendix A.6 for further explanation).

Assumption 1. *(A1) $f \in C^1(\mathbb{R}^d)$ and it has a finite lower bound $f^\star := \inf_{x \in \mathbb{R}^d} f(x) > -\infty$.*
(A2) The objective f is L-smooth: $\forall \boldsymbol{x} \forall \boldsymbol{y} \in \mathbb{R}^d (\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L \|\boldsymbol{x} - \boldsymbol{y}\|)$.
(A3) The mapping $\nabla f_\xi : \mathbb{R}^d \to \mathbb{R}^d$ called a stochastic gradient of ∇f satisfies the following: (i): $\forall \boldsymbol{x} \in \mathbb{R}^d \left(\mathbb{E}_\xi \left[\nabla f_\xi(\boldsymbol{x})\right] = \nabla f(\boldsymbol{x})\right)$, (ii): $\exists \sigma \geq 0 \forall \boldsymbol{x} \in \mathbb{R}^d \left(\mathbb{E}_\xi \left[\|\nabla f_\xi(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2\right] \leq \sigma^2\right)$.

Let $\xi_{t,i}$ be an i.i.d. sample distributed with Π and $\boldsymbol{\xi}_t := (\xi_{t,1}, \ldots, \xi_{t,b_t})^\top$. Note that $\boldsymbol{\xi}_t$ is independent of a $\{\boldsymbol{x}_k\}_{k=0}^t$ generated by SF-SGD. The mini-batch gradient is defined as $\nabla f_{B_t}(\boldsymbol{x}) := (1/b_t) \sum_{i=1}^{b_t} \nabla f_{\xi_{t,i}}(\boldsymbol{x})$. Under Assumption 1 (A3), one can easily check $\mathbb{E}_{\boldsymbol{\xi}_t}[\|\nabla f_{B_t}(\boldsymbol{x}) - \nabla f(\boldsymbol{x})\|^2] \leq \sigma^2/b_t$. The total expectation with respect to the data sampling as the iterated expectation is denoted by $\mathbb{E} := \mathbb{E}_{\boldsymbol{\xi}_0} \mathbb{E}_{\boldsymbol{\xi}_1} \cdots \mathbb{E}_{\boldsymbol{\xi}_t}$.

3 Main Result

We present the convergence analysis of SF-SGD in Algorithm 1. Let $\boldsymbol{\Delta}_t := \boldsymbol{z}_t - \boldsymbol{x}_t$. Our proof uses the following Lyapunov function (Brown et al., 2025):

$$V_t := f(\boldsymbol{x}_t) + \Lambda \|\boldsymbol{\Delta}_t\|^2 \ (t \in \mathbb{N}), \text{ where } \Lambda = \Lambda(c, \eta) := \frac{c(1 - L\eta c)}{2\eta(1-c)^2}. \tag{2}$$

Using this Lyapunov function and Assumption 1, we develop the descent lemma. The definition of Λ is convenient to prove the descent lemma. More explanation and the proof are provided in Appendix A.2.

Lemma 1. *Let $c \in (0,1)$, $\eta > 0$, and $\beta \in [0,1]$. Under Assumption 1, the sequence $\{\boldsymbol{x}_t\}_{t\geq 0}$ generated by Algorithm 1 satisfies that, for all $t \in \mathbb{N}$,*

$$\mathbb{E}[V_{t+1}] \leq \mathbb{E}[V_t] - \frac{c\eta}{2}\mathbb{E}[\|\nabla f(\boldsymbol{x}_t)\|^2] + R(c,\eta,\beta)\mathbb{E}[\|\boldsymbol{\Delta}_t\|^2] + \frac{c\eta\sigma^2}{2b_t},$$

where $R(c,\eta,\beta) := c\{1 + L^2(1-\beta)^2(1+\eta)\}/2 + Lc^2/2 - c^2(1 - L\eta c)/\{4\eta(1-c)\}$.

To ensure the decent of the Lyapunov function V_t, it is necessary for the coefficient of $\mathbb{E}[\|\boldsymbol{\Delta}_t\|^2]$ is non-positive. The next proposition determines a sufficient condition on (c, η, β) for satisfying $R(c,\eta,\beta) < 0$. The proof is in Appendix A.3.

Proposition 1. *For all $\epsilon \in (0,1), c \in (\epsilon, 1)$, and $\beta \in [0,1]$, there exists a positive value*

$$\delta = \begin{cases} \dfrac{\epsilon}{2\max\{1+L, L^2(1-\beta)\} + L\sqrt{1-\beta}} & (\beta \in [0,1)), \\ \dfrac{\epsilon}{2\{(1-\epsilon) + L\epsilon\}} & (\beta = 1), \end{cases}$$

such that, for all $\eta \in (0, \delta)$, $R(c,\eta,\beta) < 0$ holds.

δ is monotonically increasing with respect to β; that is, a larger β allows the learning rate η to be chosen from a wider range $(0, \delta)$. When β is close to 1, i.e., $\beta \approx 1$, the admissible upper bound on the learning rate becomes $\delta = \epsilon/2(1+L)$, which is comparable to the standard condition $\eta = O(1/L)$. Furthermore, a larger ϵ allows a larger learning rate to be chosen. Finally, larger values of c and η reduce the upper bound established in Theorem 1. Now, combining Lemma 1 with Proposition 1, we obtain the convergence analysis of SF-SGD as follows. The proof is in Appendix A.4.

Theorem 1 (Convergence rate of mini-batch SF-SGD with constant c)**.** *Let $\epsilon \in (0,1)$, $c \in (\epsilon, 1)$, $\beta \in [0,1]$, and $\eta \in (0, \delta)$, where δ is provided in Proposition 1. Under Assumption 1, the sequence $\{\boldsymbol{x}_t\}_{t\geq 0}$ generated by Algorithm 1 satisfies that, for all $T \in \mathbb{N}$,*

$$G^2(\boldsymbol{x}_T) := \min_{t \in [0:T-1]} \mathbb{E}\big[\|\nabla f(\boldsymbol{x}_t)\|^2\big] \leq \frac{2(f(\boldsymbol{x}_0) - f^\star)}{c\eta T} + \frac{\sigma^2}{T}\sum_{t=0}^{T-1}\frac{1}{b_t}.$$

Therefore, (i) using a constant batch size $b_t := b$ implies $G^2(\boldsymbol{x}_T) = O(1/T + \sigma^2/b)$; (ii) using a batch size b_t with $B := \sum_{t=0}^{+\infty} b_t^{-1} < +\infty$ implies $G^2(\boldsymbol{x}_T) = O(1/T)$. On the other hand, replacing the assumption $\eta \in (0, \delta)$ with $\eta \in (0, \delta/2)$, we obtain

$$G^2(\boldsymbol{y}_T) \leq \frac{4(f(\boldsymbol{x}_0) - f^\star)}{c\eta T} + 2\frac{\sigma^2}{T}\sum_{t=0}^{T-1}\frac{1}{b_t}, \quad G^2(\boldsymbol{x}_T, \boldsymbol{y}_T) \leq \frac{12(f(\boldsymbol{x}_0) - f^\star)}{c\eta T} + 6\frac{\sigma^2}{T}\sum_{t=0}^{T-1}\frac{1}{b_t},$$

where we denote $G^2(\boldsymbol{x}_T, \boldsymbol{y}_T) := \min_{t\in[0:T-1]} \mathbb{E}[\|\nabla f(\boldsymbol{x}_t) - \nabla f(\boldsymbol{y}_t)\|^2]$.

An example of the batch size scheduler is an increasing batch size (Umeda & Iiduka, 2025) denoted by

$$\{b_t\}_{t\in\mathbb{N}\cup\{0\}} = \{\underbrace{b_0, \dots, b_0}_{\Delta T_0}, \underbrace{b_1\gamma, \dots, b_1\gamma}_{\Delta T_1}, \dots, \underbrace{b_0\gamma^{M-1}, \dots, b_0\gamma^{M-1}}_{\Delta T_M}, \dots\}.$$

The $O(1/T)$ obtained from $b_t \to \infty$ may have limited practical value. However, it is known that training can be accelerated even with a finite number of increments (Oowada & Iiduka, 2025).

The upper bound for the evaluation point $\boldsymbol{x}_t$ is smaller than that for the gradient-location point $\boldsymbol{y}_t$. Moreover, the gradient-norm difference between $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ decreases with the number of iterations.

Why is constant averaging effective?
From (1), we have

$$\boldsymbol{x}_T = \sum_{t=0}^{T} w_{t;T}\boldsymbol{y}_t, \quad w_{t;T} = \frac{c_t}{(1-c_t)(1-\beta)+c_t}\prod_{j=t+1}^{T}\frac{(1-c_j)(1-\beta)}{(1-c_j)(1-\beta)+c_j}.$$

Since $\sum_{t=0}^{T} w_{t;T} = 1$, $\boldsymbol{x}_T$ is a weighted sum of $\{\boldsymbol{y}_t\}_{t=0}^{T}$; namely, $\boldsymbol{x}_T$ accumulates information from past momentum iterates. Comparing $w_{t;T}$ for the uniform averaging coefficient $c_t = 1/(t+1)$ derived by Song et al. (2025) with $w_{t;T}$ for our constant averaging coefficient, we obtain the following:

$$w_{t;T} \approx \frac{1}{(T+1)(1-\beta)}\left(\frac{t+1}{T+1}\right)^{\frac{\beta}{1-\beta}} \ (c_t = \frac{1}{t+1}), \ w_{t;T} = \frac{c}{(1-c)(1-\beta)+c}\left\{\frac{(1-c)(1-\beta)}{(1-c)(1-\beta)+c}\right\}^{T-t} \ (c_t = c).$$

The weights accumulated in $\boldsymbol{x}_T$ increase polynomially toward more recent iterates under uniform averaging, whereas they increase exponentially under constant averaging. In particular, under uniform averaging, a smaller β makes $w_{t;T}$ closer to $1/(T+1)$, causing $\boldsymbol{x}_T$ to reflect past iterates more strongly, whereas a larger β assigns greater weight to recent information than to past information. In language-model experiments, Song et al. (2025) empirically showed that, for small β, $\{\boldsymbol{x}_t\}_{t=0}^{T}$ performs worse than $\{\boldsymbol{y}_t\}_{t=0}^{T}$, whereas for large β, $\{\boldsymbol{x}_t\}_{t=0}^{T}$ outperforms $\{\boldsymbol{y}_t\}_{t=0}^{T}$. This observation is consistent with our theoretical analysis. Under constant averaging, as $\beta \to 0$, the weights approach $c(1-c)^{T-t}$. Unlike uniform averaging, whose weights approach $1/(T+1)$, these limiting weights still increase exponentially with t. Thus, constant averaging places greater weight on recent iterates regardless of the choice of β, resembling the weighting behavior of uniform averaging with a large β. This weighting structure provides a rationale for constant averaging, and, from this perspective, Theorem 1 lends theoretical support to the empirical observation reported by Song et al. (2025). Using the mini-batch settings, Section 4 investigates the convergence behavior of SF-SGD, the effects of the averaging coefficient c_t and momentum parameter β, and the sensitivity of the constant averaging scheme $c_t = c$ to the choice of β. In particular, the robustness of constant averaging to β with respect to the gradient norms is consistent with the reasoning above. Moreover, constant averaging is observed to outperform uniform averaging in some settings, empirically suggesting an advantage.

4 Numerical Experiments

We trained ResNet-18 on CIFAR-100 and ResNet-34 on ImageNet (He et al., 2016) using Algorithm 1 with two averaging coefficients: a uniform coefficient (Uniform, $c_t = 1/(t+1)$) and a constant coefficient (Constant, c). We measured test accuracy and the gradient norms at both $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$. As discussed in Proposition 1, when β is close to 1, the admissible upper bound on the learning rate becomes comparable to the standard $O(1/L)$ condition. Accordingly, we fix $\eta = 0.1$, following the initial learning rate used in the ResNet experiments of He et al. (2016). The learning rate and weight decay are fixed at 0.1 and 0.0001, respectively, in all experiments, and the constant averaging coefficient c is selected from $\{0.001, 0.05, 0.1, 0.2\}$ according to the experimental setting. Since $\boldsymbol{x}_0 = \boldsymbol{y}_0$, the gradient norm difference is trivially zero at initialization. We therefore plot the gradient norm difference from the first evaluation point. Full gradient norms are evaluated every 100 optimization steps on CIFAR-100. On ImageNet, they are evaluated every 10 epochs in the fixed-batch setting and every 5 epochs in the increasing-batch setting. For all experiments, the solid lines in the figures represent the mean values, and the shaded areas indicate the maximum and minimum over three runs. The experiments were conducted on an NVIDIA A100 GPU using Python 3.12, PyTorch (torch $\geq$ 2.10.0, torchvision $\geq$ 0.25.0), and CUDA 12.8.1. The experimental code is available at `https://anonymous.4open.science/r/schedule_free_convergence-CBAD`.

Results for fixed batch size. We first compare the two averaging coefficients on CIFAR-100 with a fixed batch size of 128. We train each model for $80{,}000$ steps, corresponding to approximately 205 epochs with a fixed batch size of 128. We set β to 0.9 and use $c = 0.1$ for Constant.

The top row of Figure 1 shows the results for the fixed batch size, including test accuracy, the running minimum of each gradient norm, and the difference of gradient norms. In the early steps, Uniform gives a smaller gradient norm at $\boldsymbol{x}_t$. In the later steps, Constant gives smaller values at both $\boldsymbol{x}_t$

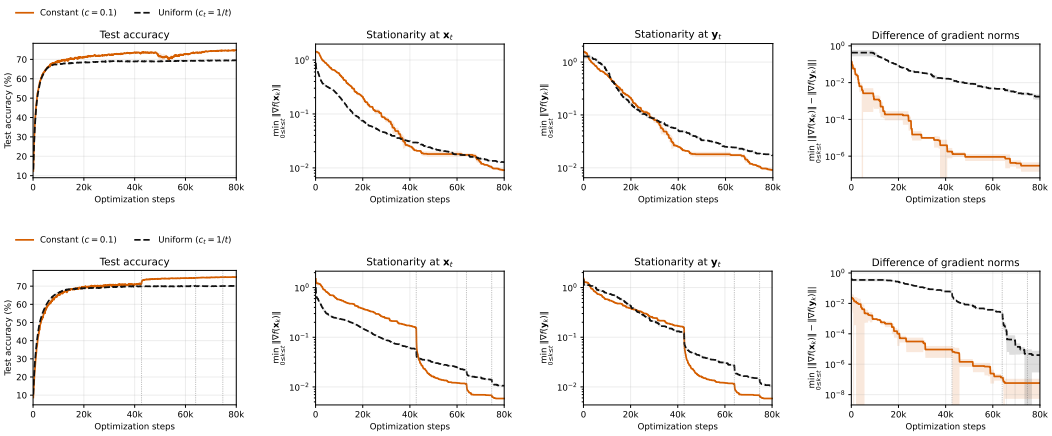

Figure 1: Results on CIFAR-100 with fixed and staged increasing batch sizes. The top row shows the fixed batch size of 128, and the bottom row shows the staged increasing batch-size strategy $b_m = b_0\gamma^m$ with $b_0 = 64$, $\gamma = 2$, and approximately 54.5 epochs per stage. From left to right, the panels show test accuracy, the running minimum full-gradient norms at $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$, and the running minimum difference of gradient norms $|\|\nabla f(\boldsymbol{x}_t)\| - \|\nabla f(\boldsymbol{y}_t)\||$. Vertical lines in the bottom row indicate stage changes.

and $\boldsymbol{y}_t$. Constant also gives a smaller difference of gradient norms and higher accuracy in the later steps. Thus, in this fixed-batch experiment, Constant gives smaller late-stage gradient norms at both parameter sequences and higher test accuracy than Uniform.

Results for increasing batch sizes. We use a staged increasing batch-size strategy that divides training into M stages. In each stage $m \in \{0, 1, \ldots, M-1\}$, the batch size b_m follows $b_m = b_0\gamma^m$ and remains constant throughout the stage. We set $b_0 = 64$ and $\gamma = 2$, and each stage processes approximately 54.5 epochs. We use the same β and c as in the fixed batch experiment.

The bottom row of Figure 1 shows the test accuracy and gradient norms. As in the fixed-batch setting, Constant achieves smaller late-stage gradient norms at both $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$, a smaller gradient-norm difference, and higher mean test accuracy than Uniform. Both methods also reduce the gradient norms as the batch size increases.

Table 2: Results at same training-sample budgets. All runs are evaluated at 10.24 million samples.

Matched training-sample budget				
Method	Batch schedule	Test accuracy (%)	$\min_{0\le k\le t} \|\nabla f(\boldsymbol{x}_k)\|$	$\min_{0\le k\le t} \|\nabla f(\boldsymbol{y}_k)\|$
Uniform	Fixed 128	69.54	1.28×10^{-2}	1.72×10^{-2}
Uniform	Increasing	**70.05**	$\mathbf{1.06} \times 10^{-2}$	$\mathbf{1.04} \times 10^{-2}$
Constant	Fixed 128	74.83	9.09×10^{-3}	9.10×10^{-3}
Constant	Increasing	**75.03**	$\mathbf{5.842} \times 10^{-3}$	$\mathbf{5.840} \times 10^{-3}$

Table 2 compares the two batch schedules under the same sample budgets. For both averaging coefficients, increasing the batch size gives higher test accuracy and smaller gradient norms. The advantage of Constant over Uniform also remains in the increasing-batch setting.

Sensitivity to the momentum parameter and the constant averaging coefficient. We examine the sensitivity to the momentum parameter β on CIFAR-100 using a fixed batch size of 128. We compare $\beta \in \{0.8, 0.9, 0.95, 0.99\}$ for Constant and Uniform over 50,000 steps. For the constant averaging coefficient, we explore $c \in \{0.001, 0.05, 0.1, 0.2\}$. The other settings are the same as in the fixed-batch experiment.

Previous work examined whether the best momentum parameter β changes with the training horizon and found that $\beta = 0.9$ remained optimal across different training durations in its ImageNet set-

ting (Defazio et al., 2024, Section 5). In our CIFAR-100 experiments, Uniform attains its highest test accuracy at $\beta = 0.8$, indicating that the preferred value of β can differ across experimental settings. Moreover, under the same training duration, the achieved test accuracy varies more across β for Uniform than for Constant. A similar difference is observed in the minimum full-gradient norms at both $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$. For Constant, they remain at a similar level across the tested β values, whereas for Uniform they generally increase with β. The gradient norm difference for Constant depends on β, with larger values generally producing smaller differences, although the relationship is not strictly monotonic. Figure 3 shows the sensitivity of the test accuracy and the gradient norms. No significant performance differences are observed among $c = 0.05$, 0.1, and 0.2, whereas the final test accuracy slightly decreases when $c = 0.001$. A clearer difference is observed in the gradient-norm difference, where $c = 0.001$ remains substantially larger than the other three settings throughout training, while $c = 0.05$, 0.1, and 0.2 decrease to much smaller values. Although some differences remain among these three values of c, their overall behavior is similar. These results suggest that the method is relatively robust to the choice of c within a certain range, while some choices of c can adversely affect both test accuracy and the agreement between the gradient norms at $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$.

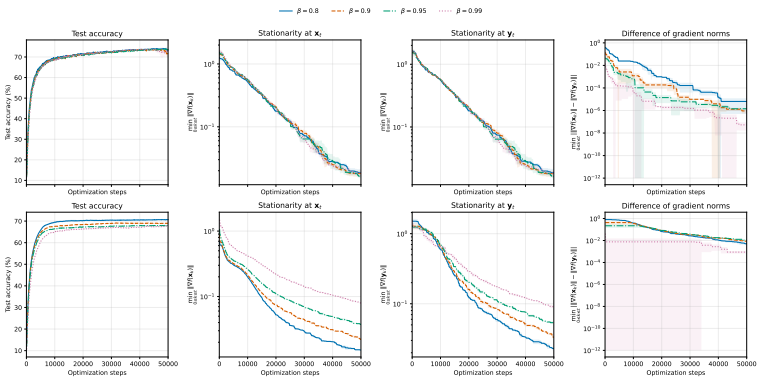

Figure 2: Sensitivity to β on CIFAR-100 with a fixed batch size of 128. The top row uses Constant averaging and the bottom row uses Uniform averaging. From left to right, the panels show test accuracy, the running minimum full-gradient norm at $\boldsymbol{x}_t$, the running minimum full-gradient norm at $\boldsymbol{y}_t$, and the running minimum difference of gradient norms $|\|\nabla f(\boldsymbol{x}_t)\| - \|\nabla f(\boldsymbol{y}_t)\||$.

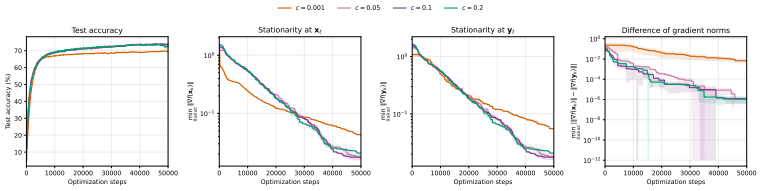

Figure 3: Sensitivity to the constant averaging coefficient c on CIFAR-100 with a fixed batch size of 128. From left to right, the panels show test accuracy, the running minimum full-gradient norm at $\boldsymbol{x}_t$, the running minimum full-gradient norm at $\boldsymbol{y}_t$, and the running minimum difference of gradient norms $|\|\nabla f(\boldsymbol{x}_t)\| - \|\nabla f(\boldsymbol{y}_t)\||$.

Results on ImageNet. We further compare the two averaging coefficients on ImageNet-1K using ResNet-34. For the fixed-batch experiment, we use a batch size of 512, $\beta = 0.9$, and $c = 0.001$ for Constant. We train each model for 500,600 steps, corresponding to approximately 200 epochs with a fixed batch size of 512. For the increasing-batch experiment, we use the same staged strategy $b_m = b_0\gamma^m$ as above, with $b_0 = 64$ and $\gamma = 2$. Each stage consists of 13 epochs. The total number of steps, β, and c are the same as in the fixed-batch experiment.

The top row of Figure 4 shows that Constant achieves higher mean test accuracy than Uniform during most of training for the fixed batch size. The two averaging coefficients behave differently for the full-gradient norms at $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$: Uniform achieves a smaller running minimum gradient norm at $\boldsymbol{x}_t$, whereas Constant achieves a smaller value at $\boldsymbol{y}_t$. Constant also gives a smaller difference of gradient

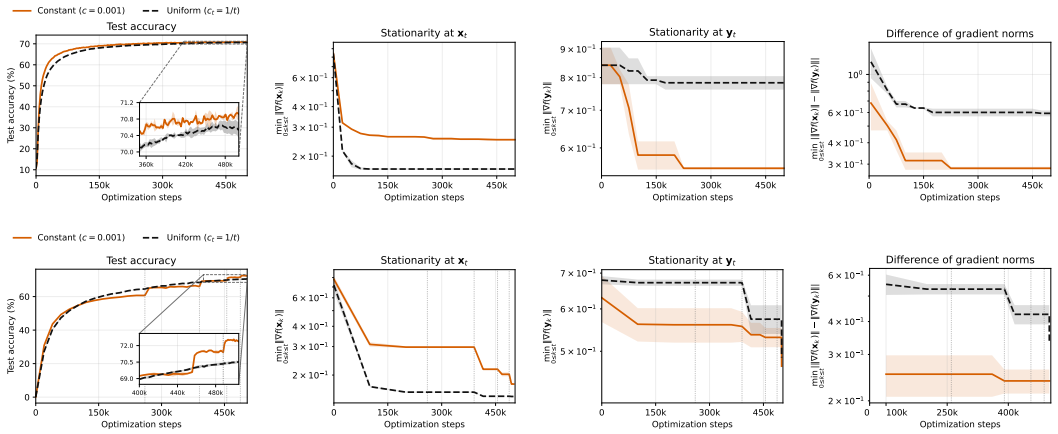

Figure 4: Results on ImageNet-1K with fixed and staged increasing batch sizes. The top row shows the fixed batch size of 512, and the bottom row shows the staged increasing batch-size strategy $b_m = b_0\gamma^m$, with each stage consisting of 13 epochs. From left to right, the panels show test accuracy, the running minimum full-gradient norms at $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$, and the running minimum difference of gradient norms $|\|\nabla f(\boldsymbol{x}_t)\| - \|\nabla f(\boldsymbol{y}_t)\||$. Vertical lines in the bottom row indicate stage changes.

norms. For the increasing batch-size schedule, the bottom of Figure 4 shows the behavior of the two averaging schemes as the batch size increases. As the batch size increases, Constant shows a gradual increase in test accuracy and achieves higher accuracy than Uniform in the later stages of training. When compared after the same number of optimization steps as in the fixed batch setting, Uniform achieves nearly the same test accuracy, whereas Constant achieves clearly higher accuracy. As in the fixed batch setting, Uniform achieves a smaller running minimum gradient norm at $\boldsymbol{x}_t$, while Constant achieves a smaller value at $\boldsymbol{y}_t$. The difference of gradient norms is also smaller for Constant under this schedule.

5 Conclusion and Future Work

We investigated the effect of implicit weighted averaging in Schedule-Free SGD under nonconvex objectives, focusing on the constant averaging setting $c_t = c$. Under this setting, we established convergence guarantees for both the evaluation point $\boldsymbol{x}_t$ and the gradient-location point $\boldsymbol{y}_t$, along with an upper bound on their gradient-norm gap. In particular, our analysis shows that the upper bound at $\boldsymbol{x}_t$ can be smaller than that at $\boldsymbol{y}_t$, providing theoretical evidence that implicit weighted averaging can improve the stationarity of the evaluation point. Our numerical experiments further show that constant averaging can outperform uniform averaging $c_t = 1/(t+1)$ and is more robust to β. We also extended the analysis to mini-batch settings without bounded-gradient assumptions and observed advantages of increasing over fixed batch sizes.

Although our results clarify several properties of implicit weighted averaging in SF-SGD, several questions remain open. One important direction is to establish convergence guarantees for the original uniform averaging coefficient under the same general setting and compare them with both existing results and our constant-averaging analysis. Another is to obtain a finer characterization of the relationship between $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$. For example, under general nonconvex objectives, our current analysis does not appear to yield a bound of the form $\langle \nabla f(\boldsymbol{y}_t), \nabla f(\boldsymbol{x}_t) - \nabla f(\boldsymbol{y}_t)\rangle < -C_1 + C_2$, where $C_1, C_2 > 0$. Establishing such a bound with $C_2 = 0$ would provide a more direct theoretical characterization of when implicit averaging improves stationarity, which we leave as an interesting direction for future work.

6 AI use statement

In this work, we used generative AI tools for the design and refinement of experiments and the implementation and modification of experimental and analysis code. We did not use generative AI

tools to formulate mathematical claims, provide essential ingredients for mathematical proofs, or write proofs. The generation of synthetic datasets, qualitative or thematic data analysis, formulation of survey or interview questions, and transcription of research recordings were not applicable to this work. Additionally, we used generative AI tools for translation, identification of relevant literature, creation and modification of scientific figures, and information searching. We have reviewed all AI-assisted work; all generated code and analysis scripts were verified through execution and inspection, scientific figures were checked against the underlying experimental data, and literature-related suggestions and information obtained through AI-assisted searches were checked against the original sources. We take responsibility for the final content of this work, including text, claims, code, figures, and other artifacts produced with the aid of generative AI.

REFERENCES

Kwangjun Ahn, Gagik Magakyan, and Ashok Cutkosky. General framework for online-to-nonconvex conversion: Schedule-free SGD is also effective for nonconvex optimization. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=etxseIT47b.

Yossi Arjevani, Yair Carmon, John C. Duchi, Dylan J. Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *Mathematical Programming*, 1999(1): 165–214, 2023.

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

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

Connor Brown, Ahmed Khaled, and Chi Jin. Analysis of schedule free non-convex optimization. In *OPT 2025: Optimization for Machine Learning*, 2025. URL https://openreview.net/forum?id=Foeq0QJfMk.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020.

Jiseok Chae and Donghwan Kim. Understanding schedule-free methods in nonconvex optimization: Rate guarantees and escaping saddles, 2026. URL https://arxiv.org/abs/2607.09167.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. PaLM: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Ashok Cutkosky. Anytime online-to-batch, optimism and acceleration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 1446–1454, 2019.

Aaron Defazio and Samy Jelassi. A momentumized, adaptive, dual averaged gradient method. *Journal of Machine Learning Research*, 23(144):1–34, 2022. URL http://jmlr.org/papers/v23/21-0226.html.

Aaron Defazio, Xingyu Yang, Harsh Mehta, Konstantin Mishchenko, Ahmed Khaled, and Ashok Cutkosky. The road less scheduled. *Advances in Neural Information Processing Systems*, 37: 9974–10007, 2024.

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30016–30030, 2022.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

Llama Team. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

Jun Lu. Gradient descent, stochastic optimization, and other tales, 2024. URL https://arxiv.org/abs/2205.00832.

Depen Morwani, Nikhil Vyas, Hanlin Zhang, and Sham M. Kakade. Connections between schedule-free SGD, accelerated SGD variants, and weight averaging. In *OPT 2024: Optimization for Machine Learning*, 2024. URL https://openreview.net/forum?id=dueRRdqvwE.

Yu Nesterov and Vladimir Shikhman. Quasi-monotone subgradient methods for nonsmooth convex minimization. *Journal of Optimization Theory and Applications*, 165(3):917–940, 2015.

Lam Nguyen, Phuong Ha Nguyen, Marten van Dijk, Peter Richtarik, Katya Scheinberg, and Martin Takac. SGD and hogwild! Convergence without the bounded gradients assumption. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 3750–3758. Proceedings of Machine Learning Research, 2018.

Kosuke Nishida, Kyosuke Nishida, and Kuniko Saito. Initialization of large language models via reparameterization to mitigate loss spikes. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22699–22714, November 2024.

Kanata Oowada and Hideaki Iiduka. Faster convergence of riemannian stochastic gradient descent with increasing batch size. In *The 17th Asian Conference on Machine Learning*, volume 304, 2025.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pp. 1310–1318, 2013.

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

Boris T Polyak. New stochastic approximation type procedures. *Automat. i Telemekh*, 7(98-107):2, 1990.

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

David Ruppert. Efficient estimations from a slowly convergent Robbins-Monro process. Technical report, Cornell University Operations Research and Industrial Engineering, 1988.

Matt L. Sampson and Peter Melchior. Dynamics of learning: Generative schedules from latent odes, 2025. URL https://arxiv.org/abs/2509.23052.

Minhak Song, Beomhan Baek, Kwangjun Ahn, and Chulhee Yun. Through the river: Understanding the benefit of schedule-free methods for language model training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=CGx4XU9rCA.

Wei Tao, Zhisong Pan, Gaowei Wu, and Qing Tao. Primal averaging: A new gradient evaluation step to attain the optimal individual convergence. *IEEE transactions on cybernetics*, 50(2):835–845, 2018.

Hikaru Umeda and Hideaki Iiduka. Increasing both batch size and learning rate accelerates stochastic gradient descent. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=sbmp55k6iE.

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.

A Proofs

A.1 Useful Lemma

We prepare a useful lemma for the analysis of SF-SGD.

Lemma 2. *Consider the sequence $\{\boldsymbol{x}_t\}_{t\geq 0}$ generated by Algorithm 1 under Assumption 1. Let $\boldsymbol{\Delta}_t \coloneqq \boldsymbol{z}_t - \boldsymbol{x}_t$ for all $t \in \mathbb{N}$. Then, the following hold.*

(i) $\boldsymbol{x}_t - \boldsymbol{y}_t = -(1-\beta)\boldsymbol{\Delta}_t$;

(ii) $\boldsymbol{x}_{t+1} - \boldsymbol{x}_t = c_{t+1}(\boldsymbol{\Delta}_t - \eta \nabla f_{B_t}(\boldsymbol{y}_t))$;

(iii) $\mathbb{E}_{\boldsymbol{\xi}_t}[\|\boldsymbol{\Delta}_t - \eta \nabla f_{B_t}(\boldsymbol{y}_t)\|^2] \leq \|\boldsymbol{\Delta}_t\|^2 - 2\eta\langle \boldsymbol{\Delta}_t, \nabla f(\boldsymbol{y}_t)\rangle + \eta^2 \|\nabla f(\boldsymbol{y}_t)\|^2 + \frac{\eta^2\sigma^2}{b_t}$;

(iv) $\langle \nabla f(\boldsymbol{x}_t), \boldsymbol{\Delta}_t\rangle \leq \frac{L^2(1-\beta)^2}{2}\|\boldsymbol{\Delta}_t\|^2 + \frac{1}{2}\|\boldsymbol{\Delta}_t\|^2 + \langle \nabla f(\boldsymbol{y}_t), \boldsymbol{\Delta}_t\rangle$;

(v) $\mathbb{E}_{\boldsymbol{\xi}_t}[\langle \nabla f(\boldsymbol{x}_t), \nabla f_{B_t}(\boldsymbol{y}_t)\rangle] \geq \frac{1}{2}\{\|\nabla f(\boldsymbol{x}_t)\|^2 + \|\nabla f(\boldsymbol{y}_t)\|^2 - L^2(1-\beta)^2\|\boldsymbol{\Delta}_t\|^2\}$.

Proof. (i) We have $\boldsymbol{x}_t - \boldsymbol{y}_t = \boldsymbol{x}_t - \{(1-\beta)\boldsymbol{z}_t + \beta \boldsymbol{x}_t\} = (1-\beta)(\boldsymbol{x}_t - \boldsymbol{z}_t) = -(1-\beta)\boldsymbol{\Delta}_t$.

(ii) We have $\boldsymbol{x}_{t+1} = (1-c_{t+1})\boldsymbol{x}_t + c_{t+1}\boldsymbol{z}_{t+1} = (1-c_{t+1})\boldsymbol{x}_t + c_{t+1}(\boldsymbol{z}_t - \eta \nabla f_{B_t}(\boldsymbol{y}_t)) = \boldsymbol{x}_t + c_{t+1}(\boldsymbol{\Delta}_t - \eta \nabla f_{B_t}(\boldsymbol{y}_t))$.

(iii) Assumption 1 (A3) implies that

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\xi}_t}[\|\nabla f(\boldsymbol{y}_t)\|^2] &= \mathbb{E}_{\boldsymbol{\xi}_t}[\nabla f_{B_t}(\boldsymbol{y}_t) - \nabla f(\boldsymbol{y}_t)\|^2] + 2\mathbb{E}_{\boldsymbol{\xi}_t}[\langle \nabla f_{B_t}(\boldsymbol{y}_t), \nabla f(\boldsymbol{y}_t)\rangle] - \mathbb{E}_{\boldsymbol{\xi}_t}[\|\nabla f(\boldsymbol{y}_t)\|^2] \\ &= \mathbb{E}_{\boldsymbol{\xi}_t}[\|\nabla f_{B_t}(\boldsymbol{y}_t) - \nabla f(\boldsymbol{y}_t)\|^2] + \|\nabla f(\boldsymbol{y}_t)\|^2.\end{aligned}$$

Accordingly,

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{\xi}_t}\left[\|\boldsymbol{\Delta}_t - \eta \nabla f_{B_t}(\boldsymbol{y}_t)\|^2\right] \\ &= \mathbb{E}_{\boldsymbol{\xi}_t}\left[\|\boldsymbol{\Delta}_t\|^2 - 2\langle \boldsymbol{\Delta}_t, \nabla f_{B_t}(\boldsymbol{y}_t)\rangle + \eta^2 \|\nabla f_{B_t}(\boldsymbol{y}_t)\|^2\right] \\ &= \|\boldsymbol{\Delta}_t\|^2 - 2\eta\langle \boldsymbol{\Delta}_t, \nabla f(\boldsymbol{y}_t)\rangle + \eta^2 \mathbb{E}_{\boldsymbol{\xi}_t}\left[\|\nabla f_{B_t}(\boldsymbol{y}_t)\|^2\right] \\ &= \|\boldsymbol{\Delta}_t\|^2 - 2\eta\langle \boldsymbol{\Delta}_t, \nabla f(\boldsymbol{y}_t)\rangle + \eta^2 \left\{\|\nabla f(\boldsymbol{y}_t)\|^2 + \mathbb{E}_{\boldsymbol{\xi}_t}\left[\|\nabla f_{B_t}(\boldsymbol{y}_t) - \nabla f(\boldsymbol{y}_t)\|^2\right]\right\} \\ &\leq \|\boldsymbol{\Delta}_t\|^2 - 2\eta\langle \boldsymbol{\Delta}_t, \nabla f(\boldsymbol{y}_t)\rangle + \eta^2 \|\nabla f(\boldsymbol{y}_t)\|^2 + \frac{\eta^2\sigma^2}{b_t}.\end{aligned}$$

(iv) We have

$$\begin{aligned}\langle \nabla f(\boldsymbol{x}_t), \boldsymbol{\Delta}_t \rangle &= \langle \nabla f(\boldsymbol{x}_t) - \nabla f(\boldsymbol{y}_t), \boldsymbol{\Delta}_t \rangle + \langle \nabla f(\boldsymbol{y}_t), \boldsymbol{\Delta}_t \rangle \\ &\leq \frac{1}{2}\|\nabla f(\boldsymbol{x}_t) - \nabla f(\boldsymbol{y}_t)\|^2 + \frac{1}{2}\|\boldsymbol{\Delta}_t\|^2 + \langle \nabla f(\boldsymbol{y}_t), \boldsymbol{\Delta}_t \rangle,\end{aligned}$$

which, together with Assumption 1 (A2) and Lemma 2 (i), implies that

$$\begin{aligned}\langle \nabla f(\boldsymbol{x}_t), \boldsymbol{\Delta}_t \rangle &\leq \frac{L^2}{2}\|\boldsymbol{x}_t - \boldsymbol{y}_t\|^2 + \frac{1}{2}\|\boldsymbol{\Delta}_t\|^2 + \langle \nabla f(\boldsymbol{y}_t), \boldsymbol{\Delta}_t \rangle \\ &\leq \frac{L^2(1-\beta)^2}{2}\|\boldsymbol{\Delta}_t\|^2 + \frac{1}{2}\|\boldsymbol{\Delta}_t\|^2 + \langle \nabla f(\boldsymbol{y}_t), \boldsymbol{\Delta}_t \rangle.\end{aligned}$$

(v) Assumption 1 (A2) and Lemma 2 (i) imply that

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\xi}_t}[\langle \nabla f(\boldsymbol{x}_t), \nabla f_{B_t}(\boldsymbol{y}_t) \rangle] &= \langle \nabla f(\boldsymbol{x}_t), \nabla f(\boldsymbol{y}_t) \rangle \\ &= \frac{1}{2}\left\{\|\nabla f(\boldsymbol{x}_t)\|^2 + \|\nabla f(\boldsymbol{y}_t)\|^2 - \|\nabla f(\boldsymbol{x}_t) - \nabla f(\boldsymbol{y}_t)\|^2\right\} \\ &\geq \frac{1}{2}\left\{\|\nabla f(\boldsymbol{x}_t)\|^2 + \|\nabla f(\boldsymbol{y}_t)\|^2 - L^2\|\boldsymbol{x}_t - \boldsymbol{y}_t\|^2\right\} \\ &= \frac{1}{2}\left\{\|\nabla f(\boldsymbol{x}_t)\|^2 + \|\nabla f(\boldsymbol{y}_t)\|^2 - L^2(1-\beta)^2\|\boldsymbol{\Delta}_t\|^2\right\}.\end{aligned}$$

This completes the proof. □

A.2 Proof of Lemma 1

Based on Lemma 2, we prove Lemma 1. A key point is that the choice of Λ cancels the inner product term $\mathbb{E}[\langle \boldsymbol{\Delta}_t, \nabla f(\boldsymbol{y}_t) \rangle]$.

Proof of Lemma 1. Let $t \in \mathbb{N}$. Assumption 1 (A2) implies that

$$\begin{aligned}f(\boldsymbol{x}_{t+1}) &\leq f(\boldsymbol{x}_t) + \langle \nabla f(\boldsymbol{x}_t), \boldsymbol{x}_{t+1} - \boldsymbol{x}_t \rangle + \frac{L}{2}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t\|^2 \\ &= f(\boldsymbol{x}_t) + c\langle \nabla f(\boldsymbol{x}_t), \boldsymbol{\Delta}_t \rangle - c\eta\langle \nabla f(\boldsymbol{x}_t), \nabla f_{B_t}(\boldsymbol{y}_t) \rangle + \frac{Lc^2}{2}\|\boldsymbol{\Delta}_t - \eta \nabla f_{B_t}(\boldsymbol{y}_t)\|^2.\end{aligned}$$

Taking the expectation for the above inequality, together with Lemma 2 (ii)$\sim$(v), leads to the following inequality:

$$\begin{aligned}\mathbb{E}[f(\boldsymbol{x}_{t+1})] &\leq \mathbb{E}[f(\boldsymbol{x}_t)] + \mathbb{E}\left[c\left(\frac{L^2(1-\beta)^2}{2}\|\boldsymbol{\Delta}_t\|^2 + \frac{1}{2}\|\boldsymbol{\Delta}_t\|^2 + \langle \nabla f(\boldsymbol{y}_t), \boldsymbol{\Delta}_t \rangle\right)\right] \\ &\quad - \frac{c\eta}{2}\mathbb{E}\left[\|\nabla f(\boldsymbol{x}_t)\|^2 + \|\nabla f(\boldsymbol{y}_t)\|^2 - L^2(1-\beta)^2\|\boldsymbol{\Delta}_t\|^2\right] \\ &\quad + \frac{Lc^2}{2}\mathbb{E}\left[\|\boldsymbol{\Delta}_t\|^2 - 2\eta\langle \boldsymbol{\Delta}_t, \nabla f(\boldsymbol{y}_t) \rangle + \eta^2\|\nabla f(\boldsymbol{y}_t)\|^2 + \frac{\eta^2\sigma^2}{b_t}\right] \\ &= \mathbb{E}[f(\boldsymbol{x}_t)] - \frac{c\eta}{2}\mathbb{E}\big[\|\nabla f(\boldsymbol{x}_t)\|^2\big] - \frac{c\eta}{2}\mathbb{E}\big[\|\nabla f(\boldsymbol{y}_t)\|^2\big] + \frac{Lc^2\eta^2}{2}\mathbb{E}\big[\|\nabla f(\boldsymbol{y}_t)\|^2\big] \\ &\quad + (c - Lc^2\eta)\,\mathbb{E}[\langle \boldsymbol{\Delta}_t, \nabla f(\boldsymbol{y}_t) \rangle] \\ &\quad + \left\{\frac{c}{2} + \frac{c}{2}L^2(1-\beta)^2 + \frac{c\eta}{2}L^2(1-\beta)^2 + \frac{Lc^2}{2}\right\}\mathbb{E}\big[\|\boldsymbol{\Delta}_t\|^2\big] + \frac{Lc^2\eta^2\sigma^2}{2b_t}. \qquad (3)\end{aligned}$$

Next, we bound the term $\|\boldsymbol{\Delta}_{t+1}\|^2$. By the definition $\boldsymbol{\Delta}_{t+1} = \boldsymbol{z}_{t+1} - \boldsymbol{x}_{t+1}$, we have

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\xi}_t}[\|\boldsymbol{\Delta}_{t+1}\|^2] &= \mathbb{E}_{\boldsymbol{\xi}_t}[\|\boldsymbol{z}_{t+1} - \boldsymbol{x}_{t+1}\|^2] \\ &= \mathbb{E}_{\boldsymbol{\xi}_t}[\|\boldsymbol{z}_{t+1} - ((1-c)\boldsymbol{x}_t + c\boldsymbol{z}_{t+1})\|^2] \\ &= (1-c)^2\mathbb{E}_{\boldsymbol{\xi}_t}[\|\boldsymbol{z}_{t+1} - \boldsymbol{x}_t\|^2] \\ &= (1-c)^2\mathbb{E}_{\boldsymbol{\xi}_t}[\|\boldsymbol{z}_t - \eta\nabla f_{B_t}(\boldsymbol{y}_t) - \boldsymbol{x}_t\|^2] \\ &= (1-c)^2\mathbb{E}_{\boldsymbol{\xi}_t}[\|\boldsymbol{\Delta}_t - \eta\nabla f_{B_t}(\boldsymbol{y}_t)\|^2].\end{aligned}$$

Using Lemma 2 (iii), we obtain the following bound:

$$\begin{aligned}\mathbb{E}[\|\boldsymbol{\Delta}_{t+1}\|^2] &\leq (1-c)^2\mathbb{E}\left[\|\boldsymbol{\Delta}_t\|^2 - 2\eta\langle\boldsymbol{\Delta}_t, \nabla f(\boldsymbol{y}_t)\rangle + \eta^2\|\nabla f(\boldsymbol{y}_t)\|^2 + \frac{\eta^2\sigma^2}{b_t}\right] \\ &= (1-c)^2\mathbb{E}[\|\boldsymbol{\Delta}_t\|^2] - 2\eta(1-c)^2\mathbb{E}[\langle\boldsymbol{\Delta}_t, \nabla f(\boldsymbol{y}_t)\rangle] \\ &\quad + \eta^2(1-c)^2\mathbb{E}[\|\nabla f(\boldsymbol{y}_t)\|^2] + \frac{\eta^2\sigma^2(1-c)^2}{b_t}. \end{aligned} \tag{4}$$

Taking the total expectations in (3) and (4) gives

$$\begin{aligned}\mathbb{E}[V_{t+1}] &= \mathbb{E}[f(\boldsymbol{x}_{t+1})] + \Lambda\mathbb{E}[\|\boldsymbol{\Delta}_{t+1}\|^2] \\ &\leq \mathbb{E}[f(\boldsymbol{x}_t)] - \frac{c\eta}{2}\mathbb{E}[\|\nabla f(\boldsymbol{x}_t)\|^2] - \frac{c\eta}{2}\mathbb{E}[\|\nabla f(\boldsymbol{y}_t)\|^2] + \frac{Lc^2\eta^2}{2}\mathbb{E}[\|\nabla f(\boldsymbol{y}_t)\|^2] \\ &\quad + (c - Lc^2\eta)\mathbb{E}[\langle\boldsymbol{\Delta}_t, \nabla f(\boldsymbol{y}_t)\rangle] \\ &\quad + \left\{\frac{c}{2} + \frac{c}{2}L^2(1-\beta)^2 + \frac{c\eta}{2}L^2(1-\beta)^2 + \frac{Lc^2}{2}\right\}\mathbb{E}[\|\boldsymbol{\Delta}_t\|^2] + \frac{Lc^2\eta^2\sigma^2}{2b_t} \\ &\quad + \Lambda(1-c)^2\mathbb{E}[\|\boldsymbol{\Delta}_t\|^2] - 2\Lambda\eta(1-c)^2\mathbb{E}[\langle\boldsymbol{\Delta}_t, \nabla f(\boldsymbol{y}_t)\rangle] \\ &\quad + \Lambda\eta^2(1-c)^2\mathbb{E}[\|\nabla f(\boldsymbol{y}_t)\|^2] + \frac{\Lambda\eta^2\sigma^2(1-c)^2}{b_t} \\ &= \mathbb{E}[V_t] - \frac{c\eta}{2}\mathbb{E}[\|\nabla f(\boldsymbol{x}_t)\|^2] \\ &\quad - \left\{\frac{c\eta}{2}(1 - L\eta c) - \Lambda\eta^2(1-c)^2\right\}\mathbb{E}[\|\nabla f(\boldsymbol{y}_t)\|^2] \\ &\quad + \left\{c(1 - L\eta c) - 2\Lambda\eta(1-c)^2\right\}\mathbb{E}[\langle\boldsymbol{\Delta}_t, \nabla f(\boldsymbol{y}_t)\rangle] \\ &\quad + \left\{\frac{c}{2} + \frac{c}{2}L^2(1-\beta)^2 + \frac{c\eta}{2}L^2(1-\beta)^2 + \frac{Lc^2}{2} + \Lambda\big((1-c)^2 - 1\big)\right\}\mathbb{E}[\|\boldsymbol{\Delta}_t\|^2] \\ &\quad + \left\{\frac{Lc^2\eta^2}{2} + \Lambda\eta^2(1-c)^2\right\}\frac{\sigma^2}{b_t}, \end{aligned}$$

which, together with $(1-c)^2 < 1-c$, gives

$$\begin{aligned}\mathbb{E}[V_{t+1}] &\leq \mathbb{E}[V_t] - \frac{c\eta}{2}\mathbb{E}[\|\nabla f(\boldsymbol{x}_t)\|^2] \\ &\quad + \left\{\frac{c}{2} + \frac{c}{2}L^2(1-\beta)^2 + \frac{c\eta}{2}L^2(1-\beta)^2 + \frac{Lc^2}{2} - \frac{c^2(1-L\eta c)}{4\eta(1-c)}\right\}\mathbb{E}[\|\boldsymbol{\Delta}_t\|^2] + \frac{c\eta\sigma^2}{2b_t}. \end{aligned}$$

This completes the proof. □

A.3 Proof of Proposition 1

Proof of Proposition 1. We start with

$$\begin{aligned} &R(c,\eta,\beta) \coloneqq \frac{c}{2}\left\{1 + L^2(1-\beta)^2(1+\eta)\right\} + \frac{L}{2}c^2 - \frac{c^2(1-L\eta c)}{4\eta(1-c)} < 0 \\ &\iff -L\eta c^2 - \left\{1 + 2\eta\left\{1 + L^2(1-\beta)^2(1+\eta) - L\right\}\right\}c + 2\eta\left\{1 + L^2(1-\beta)^2(1+\eta)\right\} < 0 \\ &\iff S(c,\eta,\beta) \coloneqq L\eta c^2 + \left[1 + 2\eta\left\{1 + L^2(1-\beta)^2(1+\eta) - L\right\}\right]c \\ &\qquad - 2\eta\left\{1 + L^2(1-\beta)^2(1+\eta)\right\} > 0. \end{aligned} \tag{5}$$

Since the discriminant of S satisfies

$$D(\eta,\beta) \coloneqq \left[1 + 2\eta\left\{1 + L^2(1-\beta)^2(1+\eta) - L\right\}\right]^2 + 8L\eta^2\left\{1 + L^2(1-\beta)^2(1+\eta)\right\} > 0,$$

the solutions of $S(c,\eta,\beta) > 0$ ($\iff R(c,\eta,\beta) < 0$) are given by

$$c < \frac{-A(\eta,\beta) - \sqrt{D(\eta,\beta)}}{2L\eta} \text{ and } c(\eta,\beta) \coloneqq \frac{-A(\eta,\beta) + \sqrt{D(\eta,\beta)}}{2L\eta} < c, \tag{6}$$

where $A(\eta,\beta) \coloneqq [1+2\eta\{1+L^2(1-\beta)^2(1+\eta)-L\}]$. From

$$c(\eta,\beta) = \frac{D(\eta,\beta)-A(\eta,\beta)^2}{2L\eta\{\sqrt{D(\eta,\beta)}+A(\eta,\beta)\}} = \frac{8L\eta^2\left\{1+L^2(1-\beta)^2(1+\eta)\right\}}{2L\eta\{\sqrt{D(\eta,\beta)}+A(\eta,\beta)\}},$$

we have $c(\eta,\beta)>0$ for all $\beta\in[0,1]$ and $\eta>0$. Moreover, we observe

$$c(\eta,\beta) = \frac{8L\eta^2\left\{1+L^2(1-\beta)^2(1+\eta)\right\}}{2L\eta\{\sqrt{D(\eta,\beta)}+A(\eta,\beta)\}} \to c(0,\beta)=0 \ (\eta\to 0),$$

which implies that $c(\cdot,\beta)$ is continuous at $\eta=0$. Hence, when $\beta\in[0,1]$ is fixed,

$$\forall\epsilon\in(0,1)\ \exists\delta>0\ \forall\eta>0\ \ (\eta\in(0,\delta)\implies c(\eta,\beta)<\epsilon)\,.$$

Moreover, for any $c\in(\epsilon,1)$, $c(\eta,\beta)<\epsilon<c$ holds, which, together with (6), implies that

$$\begin{aligned}&\forall\beta\in[0,1]\ \forall\epsilon\in(0,1)\ \forall c\in(\epsilon,1)\ \exists\delta>0\\ &\quad\forall\eta>0\,(\eta\in(0,\delta)\implies c(\eta,\beta)<\epsilon<c\implies S(c,\eta,\beta)>0)\,.\end{aligned}$$

We explicitly provide δ using ϵ and β. Let $\hat{A}(\epsilon,\beta)\coloneqq\{1+L^2(1-\beta)\}(1-\epsilon)+L\epsilon$. Then, we have

$$\begin{aligned}c(\eta,\beta) &= \frac{-A(\eta,\beta)+\sqrt{D(\eta,\beta)}}{2L\eta} < \epsilon\\ &\iff \sqrt{D(\eta,\beta)} < A(\eta,\beta)+2L\eta\epsilon\\ &\iff D(\eta,\beta) < A(\eta,\beta)^2+4L^2\eta^2\epsilon^2+4L\eta\epsilon A(\eta,\beta)\\ &\iff \{8L\eta^2+8L^3\eta^2(1-\beta)+8L^3\eta^3(1-\beta)\}(1-\epsilon)+8L^2\eta^2\epsilon-4L^2\eta^2\epsilon^2-4L\eta\epsilon<0\\ &\iff C(\epsilon,\eta,\beta)\coloneqq 2L^3(1-\beta)(1-\epsilon)\eta^2+2L\hat{A}(\epsilon,\beta)\eta-L\epsilon<0.\end{aligned}$$

Since $\eta>0$ and for all $\beta\in[0,1)$, the discriminant of $C(\epsilon,\eta,\beta)$ given by

$$\hat{D}(\epsilon,\beta)\coloneqq 4L^2[\{1+L^2(1-\beta)\}(1-\epsilon)+L\epsilon]^2+8L^4\epsilon(1-\beta)(1-\epsilon)$$

is positive. Therefore, we can derive the following positive upper bound on η:

$$\delta \coloneqq \begin{cases}\delta(\epsilon,\beta)=\frac{-\hat{A}(\epsilon,\beta)+\sqrt{\hat{D}(\epsilon,\beta)/4L^2}}{2L^2(1-\beta)(1-\epsilon)} & (\beta\in[0,1)),\\ \frac{\epsilon}{2\{(1-\epsilon)+L\epsilon\}} & (\beta=1).\end{cases}$$

Here, we rearrange $\delta(\epsilon,\beta)$ for $\beta\in[0,1)$: noting $\hat{D}(\eta,\beta)/(4L^2)=\hat{A}(\eta,\beta)^2+4L^2\epsilon(1-\beta)(1-\epsilon)$ and $\hat{A}(\epsilon,\beta)\le 1+L^2(1-\beta)+L\le 2\max\{1+L,L^2(1-\beta)\}$ yields

$$\begin{aligned}\delta(\epsilon,\beta) &= \frac{-\hat{A}(\epsilon,\beta)+\sqrt{\hat{D}(\epsilon,\beta)/4L^2}}{2L^2(1-\beta)(1-\epsilon)} = \frac{\hat{D}(\epsilon,\beta)/4L^2-\hat{A}(\epsilon,\beta)^2}{2L^2(1-\beta)(1-\epsilon)(\hat{A}(\epsilon,\beta)+\sqrt{\hat{D}(\epsilon,\beta)/4L^2})}\\ &= \frac{4L^2\epsilon(1-\beta)(1-\epsilon)}{2L^2(1-\beta)(1-\epsilon)(\hat{A}(\epsilon,\beta)+\sqrt{\hat{D}(\epsilon,\beta)/4L^2})} = \frac{2\epsilon}{\hat{A}(\epsilon,\beta)+\sqrt{\hat{D}(\epsilon,\beta)/4L^2}}\\ &= \frac{2\epsilon}{\hat{A}(\epsilon,\beta)+\sqrt{\hat{A}(\eta,\beta)^2+4L^2\epsilon(1-\beta)(1-\epsilon)}} \ge \frac{\epsilon}{\hat{A}(\epsilon,\beta)+L\sqrt{\epsilon(1-\beta)(1-\epsilon)}}\\ &\ge \frac{\epsilon}{2\max\{1+L,L^2(1-\beta)\}+L\sqrt{1-\beta}} \underset{\beta\approx 1}{\approx} \frac{\epsilon}{2(1+L)}.\end{aligned}$$

□

A.4 Proof of Theorem 1

Proof. Under the conditions in Theorem 1, Lemma 1 and Proposition 1 ensure that, for all $t\in\mathbb{N}$,

$$\frac{c\eta}{2}\mathbb{E}[\|\nabla f(\boldsymbol{x}_t)\|^2]-R(c,\eta,\beta)\mathbb{E}[\|\boldsymbol{\Delta}_t\|^2]\le\mathbb{E}[V_t]-\mathbb{E}[V_{t+1}]+\frac{c\eta\sigma^2}{2b_t}$$

Summing the above inequality from $t = 0$ to $t = T - 1$ and considering $\boldsymbol{z}_0 = \boldsymbol{x}_0$ and Assumption 1(A1) imply

$$\sum_{t=0}^{T-1} \left\{ \frac{c\eta}{2} \mathbb{E}[\|\nabla f(\boldsymbol{x}_t)\|^2] - R(c,\eta,\beta)\mathbb{E}[\|\boldsymbol{\Delta}_t\|^2] \right\}$$
$$\leq V_0 - \mathbb{E}[V_T] + \frac{c\eta\sigma^2}{2} \sum_{t=0}^{T-1} \frac{1}{b_t} = f(\boldsymbol{x}_0) - f(\boldsymbol{x}_T) + \Lambda(\|\boldsymbol{\Delta}_0\|^2 - \mathbb{E}\left[\|\boldsymbol{\Delta}_T\|^2\right]) + \frac{c\eta\sigma^2}{2} \sum_{t=0}^{T-1} \frac{1}{b_t}$$
$$\leq f(\boldsymbol{x}_0) - f^\star + \frac{c\eta\sigma^2}{2} \sum_{t=0}^{T-1} \frac{1}{b_t}.$$

Hence, we have

$$\min_{t\in[0:T-1]} \|\nabla f(\boldsymbol{x}_t)\|^2 \leq \frac{1}{T} \sum_{t=0}^{T-1} \|\nabla f(\boldsymbol{x}_t)\|^2 \leq \frac{2}{c\eta T} \left(\sum_{t=0}^{T-1} \frac{c\eta}{2} \|\nabla f(\boldsymbol{x}_t)\|^2 \right)$$
$$\leq \frac{2}{c\eta T} \sum_{t=0}^{T-1} \left(\frac{c\eta}{2} \|\nabla f(\boldsymbol{x}_t)\|^2 - R(c,\eta,\beta)\|\boldsymbol{\Delta}_t\|^2 \right) \leq \frac{2(f(\boldsymbol{x}_0) - f^\star)}{c\eta T} + \frac{\sigma^2}{T} \sum_{t=0}^{T-1} \frac{1}{b_t}.$$

Since $2\langle \boldsymbol{a}, \boldsymbol{b} \rangle \leq \|\boldsymbol{a}\| + \|\boldsymbol{b}\|$ leads to

$$\|\nabla f(\boldsymbol{y}_t)\|^2 \leq 2\|\nabla f(\boldsymbol{x}_t)\|^2 + 2\|\nabla f(\boldsymbol{y}_t) - \nabla f(\boldsymbol{x}_t)\|^2 \leq 2\|\nabla f(\boldsymbol{x}_t)\|^2 + 2L^2(1-\beta)^2\|\boldsymbol{\Delta}_t\|^2$$
$$\leq \max\left\{ \frac{4}{c\eta}, \frac{2L^2(1-\beta)^2}{-R(c,\eta,\beta)} \right\} \left\{ \frac{c\eta}{2} \|\nabla f(\boldsymbol{x}_t)\|^2 - R(c,\eta,\beta)\|\boldsymbol{\Delta}_t\|^2 \right\},$$

we have

$$\min_{t\in[0:T-1]} \mathbb{E}\left[\|\nabla f(\boldsymbol{y}_t)\|^2\right] \leq \frac{1}{T} \sum_{t=0}^{T-1} \|\nabla f(\boldsymbol{y}_t)\|^2 \leq \max\left\{ \frac{4}{c\eta}, \frac{2L^2(1-\beta)^2}{-R(c,\eta,\beta)} \right\} \frac{1}{T} \sum_{t=0}^{T-1} \left\{ \frac{c\eta}{2} \|\nabla f(\boldsymbol{x}_t)\|^2 - R(c,\eta,\beta)\|\boldsymbol{\Delta}_t\|^2 \right\}$$
$$\leq \max\left\{ \frac{4}{c\eta}, \frac{2L^2(1-\beta)^2}{-R(c,\eta,\beta)} \right\} \left\{ \frac{f(\boldsymbol{x}_0) - f^\star}{T} + \frac{c\eta\sigma^2}{2T} \sum_{t=0}^{T-1} \frac{1}{b_t} \right\}.$$

Moreover, the above calculations lead to

$$\min_{t\in[0:T-1]} \mathbb{E}\left[\|\nabla f(\boldsymbol{x}_t)\|^2 + \|\nabla f(\boldsymbol{y}_t)\|^2\right] \leq \frac{1}{T} \sum_{t=0}^{T-1} (\|\nabla f(\boldsymbol{x}_t)\|^2 + |\nabla f(\boldsymbol{y}_t)\|^2)$$
$$\leq \frac{2}{c\eta} \left\{ \frac{f(\boldsymbol{x}_0) - f^\star}{T} + \frac{c\eta\sigma^2}{2T} \sum_{t=0}^{T-1} \frac{1}{b_t} \right\} + \max\left\{ \frac{4}{c\eta}, \frac{2L^2(1-\beta)^2}{-R(c,\eta,\beta)} \right\} \left\{ \frac{f(\boldsymbol{x}_0) - f^\star}{T} + \frac{c\eta\sigma^2}{2T} \sum_{t=0}^{T-1} \frac{1}{b_t} \right\},$$

which implies

$$\min_{t\in[0:T-1]} \mathbb{E}\left[\|\nabla f(\boldsymbol{x}_t) - \nabla f(\boldsymbol{y}_t)\|^2\right] \leq 2 \min_{t\in[0:T-1]} \mathbb{E}\left[\|\nabla f(\boldsymbol{x}_t)\|^2 + \|\nabla f(\boldsymbol{y}_t)\|^2\right]$$
$$\leq 2\left(\frac{2}{c\eta} + \max\left\{ \frac{4}{c\eta}, \frac{2L^2(1-\beta)^2}{-R(c,\eta,\beta)} \right\} \right) \left\{ \frac{f(\boldsymbol{x}_0) - f^\star}{T} + \frac{c\eta\sigma^2}{2T} \sum_{t=0}^{T-1} \frac{1}{b_t} \right\}.$$

Here, under $\eta \in (0, \delta/2)$, the maximum equals $4/(c\eta)$. We verify that $4/(c\eta) > (2L^2(1-\beta)^2)/(-R(c,\eta,\beta))$, i.e., $-R(c,\eta,\beta) - \{c\eta L^2(1-\beta)^2\}/2 > 0$. From $\eta < \delta/2 \leq 1/(2L\sqrt{1-\beta})$, we have $4\eta^2(1-c)L^2(1-\beta)^2 < 2\eta L\sqrt{1-\beta}$. This inequality, $\eta < \delta \iff \eta(4\max\{1+L, L^2(1-\beta)\} + 2L\sqrt{1-\beta}) < \epsilon$, and

$$Lc^2 + 2(1-c)(1 + Lc + L^2(1-\beta)^2) \leq Lc + 4(1-c)\max\{1+L, L^2(1-\beta)\}$$
$$\leq 4c\max\{1+L, L^2(1-\beta)\} + 4(1-c)\max\{1+L, L^2(1-\beta)\} = 4\max\{1+L, L^2(1-\beta)\}$$

yield

$$
\begin{aligned}
-R(c,\eta,\beta) - \frac{c\eta L^2(1-\beta)^2}{2} &= \frac{c}{4\eta(1-c)}\{c - \eta\{Lc^2 + 2(1-c)(1+Lc+L^2(1-\beta)^2)\} - 4\eta^2(1-c)L^2(1-\beta)^2\} \\
&> \frac{c}{4\eta(1-c)}\{c - \eta(4\max\{1+L, L^2(1-\beta)\} + 2L\sqrt{1-\beta})\} \\
&> \frac{c}{4\eta(1-c)}(c-\epsilon) > 0.
\end{aligned}
$$

Therefore, under $\eta \in (0, \delta/2)$, we obtain

$$
\min_{t\in[0:T-1]} \mathbb{E}\left[\|\nabla f(\boldsymbol{y}_t)\|^2\right] \leq \frac{4}{c\eta}\left\{\frac{f(\boldsymbol{x}_0) - f^\star}{T} + \frac{c\eta\sigma^2}{2T}\sum_{t=0}^{T-1}\frac{1}{b_t}\right\} = O\left(\frac{1}{T} + 2\frac{\sigma^2}{b}\right),
$$

$$
\min_{t\in[0:T-1]} \mathbb{E}\left[\|\nabla f(\boldsymbol{x}_t) - \nabla f(\boldsymbol{y}_t)\|^2\right] \leq \frac{12}{c\eta}\left\{\frac{f(\boldsymbol{x}_0) - f^\star}{T} + \frac{c\eta\sigma^2}{2T}\sum_{t=0}^{T-1}\frac{1}{b_t}\right\} = O\left(\frac{1}{T} + 6\frac{\sigma^2}{b}\right).
$$

The convergence rates follow by setting $b_t = b$. □

A.5 Comparison with standard SGD

Figure 5 compares the test accuracy of Constant, Uniform, and standard SGD on CIFAR-100 with a fixed batch size of 128. Both Constant and Uniform achieve higher test accuracy than standard SGD. In particular, Constant continues to improve during training and attains the highest final test accuracy among the three methods, while Uniform also consistently outperforms SGD. These results indicate that both Constant and Uniform averaging can improve test accuracy compared with standard SGD under this experimental setting.

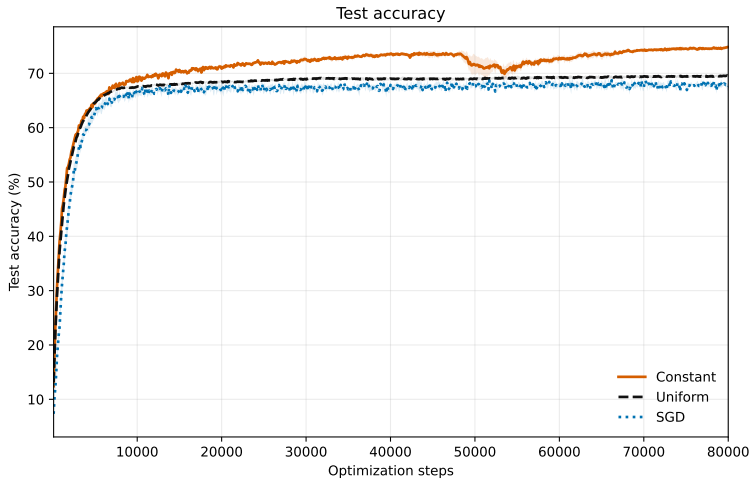

Figure 5: Comparison of test accuracy with SGD on CIFAR-100 using a fixed batch size of 128.

A.6 Signed difference of gradient norms on ImageNet

Theorem 1 provides different upper bounds for the two parameter sequences, with a larger upper bound for $G^2(\boldsymbol{y}_T)$ than for $G^2(\boldsymbol{x}_T)$. Although this difference in the upper bounds does not imply an ordering of the actual gradient norms, Figure 6 shows that, in the fixed-batch ImageNet experiment, the running minimum gradient norm at $\boldsymbol{y}_t$ remains larger than that at $\boldsymbol{x}_t$ over the recorded evaluation points. This empirical tendency may be qualitatively consistent with the difference between the theoretical upper bounds.

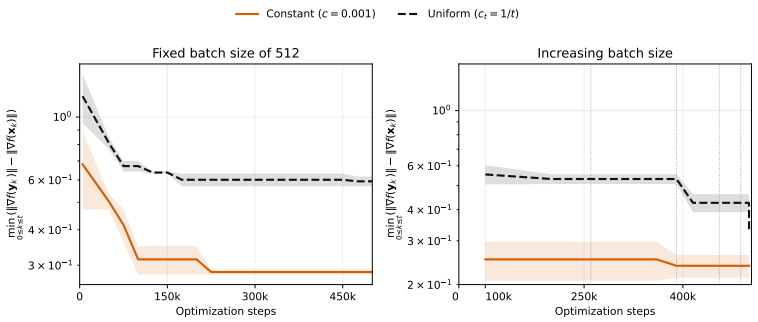

Figure 6: Signed difference of gradient norms on ImageNet-1K. The left panel shows the fixed batch size of 512, and the right panel shows the staged increasing batch-size strategy. The curves show the running minimum of $\|\nabla f(\boldsymbol{y}_t)\| - \|\nabla f(\boldsymbol{x}_t)\|$. The first gradient-evaluation point is shifted on the horizontal axis as in the main-text difference panels.

Additional Remarks on the Assumptions

The requirement $f^\star \in \mathbb{R}$ in Assumption 1(A1) excludes pathological cases in which the empirical risk has no lower bound and is natural for training deep neural networks, including those with cross-entropy losses for which the minimum value is attained. Assumption 1(A2) enables the use of the descent lemma, which underlies most convergence analyses of mini-batch SGD. Assumption 1(A3) is satisfied, for example, when Π is uniform. When Π is nonuniform, the stochastic gradient should be importance-weighted as $\nabla f_\xi(\boldsymbol{x})/(N\Pi(\{\xi\}))$ to satisfy 1(A3). However, an abuse of notation, namely, regarding $\nabla f_\xi(\boldsymbol{x})$ as notation for a random variable depending on ξ, enables us to treat the nonuniform case of Π using the same formulas as in Assumption 1(A3).