THE MARGINAL VALUE OF MOMENTUM FOR SMALL LEARNING RATE SGD

Anonymous authors
Paper under double-blind review

ABSTRACT

Momentum is known to accelerate the convergence of gradient descent in strongly convex settings without stochastic gradient noise. In stochastic optimization, such as training neural networks, folklore suggests that momentum may help deep learning optimization by reducing the variance of the stochastic gradient update, but previous theoretical analyses do not find momentum to offer any provable acceleration. Theoretical results in this paper clarify the role of momentum in stochastic settings where the learning rate is small and gradient noise is the dominant source of instability, suggesting that SGD with and without momentum behave similarly in the short and long time horizons. Experiments show that momentum indeed has limited benefits for both optimization and generalization in practical training regimes where the optimal learning rate is not very large, including small- to medium-batch training from scratch on ImageNet and fine-tuning language models on downstream tasks.

1 INTRODUCTION

In modern deep learning, it is standard to combine stochastic gradient methods with *heavy-ball momentum*, or *momentum* for short, to enable a more stable and efficient training of neural networks (Sutskever et al., 2013). The simplest form is *Stochastic Gradient Descent with Momentum* (SGDM). SGDM aims to minimize the training loss $\mathcal{L}(\boldsymbol{x})$ given a noisy gradient oracle $\mathcal{G}(\boldsymbol{x})$, which is usually realized by evaluating the gradient at a randomly sampled mini-batch from the training set. Specifically, let γ, β be the learning rate and momentum coefficient, then SGDM can be stated as:

$$\boldsymbol{g}_k \sim \mathcal{G}(\boldsymbol{x}_k), \qquad \boldsymbol{m}_{k+1} = \beta \boldsymbol{m}_k + \boldsymbol{g}_k, \qquad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \gamma \boldsymbol{m}_{k+1}, \tag{1}$$

where $\boldsymbol{g}_k, \boldsymbol{m}_k, \boldsymbol{x}_k$ are the gradient, momentum buffer, and parameter vector at step k.

For typical choices of $\beta \in (0, 1)$, the momentum buffer can be interpreted as an exponential moving average of past gradients, i.e., $\boldsymbol{m}_k = \sum_{j=0}^{k} \beta^{k-j} \boldsymbol{g}_j$. Based on this interpretation, Polyak (1964; 1987); Rumelhart et al. (1987) argued that momentum is able to cancel out oscillations along high-curvature directions and add up contributions along low-curvature directions. More concretely, for strongly convex functions without any noise in gradient estimates, Polyak (1964; 1987) showed that adding momentum can stabilize the optimization process even when the learning rate is so large that can make vanilla gradient descent diverge, and thus momentum accelerates the convergence to minimizers by allowing using a larger learning rate.

In deep learning, however, the random sampling of mini-batches inevitably introduces a large amount of stochastic gradient noise. In addition to the high-curvature directions causing the parameter to oscillate back and forth, this large gradient noise can further destabilize training and hinder the use of large learning rates. Given the aforementioned convergence results that solely analyze the noiseless case, it remains unclear whether momentum can likewise stabilize the stochastic optimization process in deep learning. Still, it is intuitive that averaging past stochastic gradients could reduce the variance of the noise in the updates, as long as the parameter does not move drastically fast at each step. Several prior studies indeed cited this reduction of noise in SGDM as a possible advantage that may encourage a more rapid decrease in loss (Bottou et al., 2018; Defazio, 2020; You et al., 2020). To approach this more rigorously, Cutkosky and Orabona (2019) proposed a variant of SGDM that provably accelerates training by leveraging the reduced variance in the updates. They further speculated that the advantage of the original SGDM might be related in some way.

Nevertheless, for SGDM without any modification, past theoretical analyses in the stochastic optimization of convex and non-convex functions usually conclude with a comparable convergence rate as vanilla SGD, rather than a faster one (Yan et al., 2018; Yu et al., 2019; Liu et al., 2020; Sebbouh et al., 2021; Li et al., 2022a). Besides, there also exist simple and concrete instances of convex optimization where momentum does not speed up the convergence rate of SGD, even though it is possible to optimize faster with some variants of SGDM (Kidambi et al., 2018). Despite these failures in theory, it has been empirically confirmed that SGDM continues to stabilize large learning rate training even in the presence of gradient noise. Kidambi et al. (2018); Shallue et al. (2019); Smith et al. (2020) observed that for large-batch training, SGDM can successfully perform training with a large learning rate, in which regime vanilla SGD may exhibit instability that degrades the training speed and generalization. This naturally raises the following question on the true role of momentum:

Does noise reduction in SGDM updates really benefit neural network training?

To address this question, this paper delves into the training regime where the learning rate is small enough to prevent oscillations along high-curvature directions, yet the gradient noise is large enough to induce instability. This setting enables us to concentrate exclusively on the interplay between momentum and gradient noise. More importantly, this training regime is of practical significance as in many situations, such as small-batch training from scratch or fine-tuning a pre-trained model, the optimal learning rate is indeed relatively small (Liu et al., 2019; Malladi et al., 2023).

Main Contributions. In this paper, we present analyses of the training trajectories of SGD with and without momentum, in the regime of small learning rate. We provide theoretical justifications of a long-held belief that SGDM with learning rate γ and momentum β performs comparably to SGD with learning rate $\frac{\gamma}{1-\beta}$ (Tugay and Tanik, 1989; Orr, 1996; Qian, 1999; Yuan et al., 2016; Smith et al., 2020). This finding offers negative evidence for the usefulness of noise reduction in momentum. Additionally, this also motivates us to reformulate SGDM in Definition 2.3 so SGDM and SGD perform comparably under the same learning rate η, which in turn simplifies our analysis.

More specifically, given a run of SGDM, we show that vanilla SGD can closely track its trajectory in the following two regimes with different time horizon:

Regime I. Training with SGD and SGDM for $O(1/\eta)$ steps where the scaling of gradient noise covariance can be as large as $O(1/\eta)$. Specifically, Theorem 3.5 shows that SGD and SGDM are $O(\sqrt{\eta/(1-\beta)})$-close to each other in the sense of weak approximation, where η, β are the learning rate and momentum coefficient under the notation of Definition 2.3. Our analysis not only includes the classical result that both SGD and SGDM converge to Gradient Flow in $O(1/\eta)$ steps where the stochastic gradient is sampled from a bounded distribution independent of η, but also covers the regime of applying Linear Scaling Rule (Goyal et al., 2017), where one decreases the learning rate and batch size at the same rate, so the noise covariance increases inversely proportional to η, and in this case both SGD and SGDM converge to a Stochastic Differential Equation. Our results improve over previous analysis (Yuan et al., 2016; Liu et al., 2018) by avoiding underestimating the role of noise when scaling down the learning rate, and provide rigorous theoretical supports to the scaling claims in Smith et al. (2020); Cowsik et al. (2022). Technically we introduce an auxiliary dynamics $\boldsymbol{y}_k$ (**??**) that bridges SGDM and SGD.

Regime II. Training with SGD and SGDM for $O(1/\eta^2)$ steps for overparametrized models where the minimizers of the loss connect as a manifold and after reaching such a manifold, the gradient noise propels the iterates to move slowly along it. Theorem 4.5 shows that SGD and SGDM follow the same dynamics along the manifold of minimizers and thus have the same implicit bias. The implicit bias result of SGD is due to Katzenberger (1991); Li et al. (2021b), but their analysis does not apply to SGDM because its dynamic depends non-homogeneously on η. Our proof of Theorem 4.5 is highly non-trivial, which carefully bounds various error terms in the decomposition.

In Section 5, we show empirically that momentum indeed has limited benefits for both optimization and generalization in practical training regimes, including small- to medium-batch training from scratch on ImageNet and fine-tuning RoBERTa-large on downstream tasks. We also look into a large-batch training case on CIFAR-10 where SGDM indeed outperforms vanilla SGD, and show that reducing training instability induced by high curvature by running an SDE simulation method called SVAG (Li et al., 2021a) can shrink or eliminate the performance gain.

Finally, we highlight that our results can also have practical significance beyond just understanding the role of momentum. In recent years, the GPU memory capacity sometimes becomes a bottleneck in training large models. As the momentum buffer costs as expensive as storing the entire model, it has raised much interest in when it is safe to remove momentum (Shazeer and Stern, 2018). Our work sheds light on this question by formally proving that momentum only provides marginal values in small learning rate SGD. Furthermore, our results imply that within reasonable range of scales the final performance is insensitive to the momentum hyperparametrization, thereby provide support to save the effort in the extensive hyperparameter grid search.

2 PRELIMINARIES

Consider optimizing a loss function $\mathcal{L} = \frac{1}{\Xi}\sum_{i=1}^{\Xi}\mathcal{L}_i$ where $\mathcal{L}_i : \mathbb{R}^d \to \mathbb{R}$ corresponds to the loss on the i-th sample. We use $\boldsymbol{\theta}$ to indicate parameters along a general trajectory. In each step, we sample a random minibatch $\mathcal{B} \subseteq [\Xi]$, and compute the gradient of the minibatch loss $\mathcal{L}_{\mathcal{B}} = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\mathcal{L}_i$ to get the following noisy estimate of $\nabla\mathcal{L}(\boldsymbol{\theta})$, i.e., $\nabla\mathcal{L}_{\mathcal{B}}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\nabla\mathcal{L}_i(\boldsymbol{\theta})$. It is easy to check that the noise covariance matrix of $\nabla\mathcal{L}_{\mathcal{B}}(\boldsymbol{\theta})$, namely $\mathbb{E}_{\mathcal{B}}(\nabla\mathcal{L}_{\mathcal{B}}(\boldsymbol{\theta}) - \nabla\mathcal{L}(\boldsymbol{\theta}))(\nabla\mathcal{L}_{\mathcal{B}}(\boldsymbol{\theta}) - \nabla\mathcal{L}(\boldsymbol{\theta}))^\top$, scales proportionally to $\frac{1}{|\mathcal{B}|}$. Motivated by this, Malladi et al. (2022) abstracts $\nabla\mathcal{L}_{\mathcal{B}}(\boldsymbol{\theta})$ as sampled from a noisy gradient oracle where the noise covariance depends on a scale parameter.

Definition 2.1 (NGOS, Malladi et al. (2022))**.** A *Noisy Gradient Oracle with Scale Parameter* (NGOS) is characterized by a tuple $\mathcal{G}_\sigma = (\mathcal{L}, \boldsymbol{\Sigma}, \mathcal{Z}_\sigma)$. For a scale parameter $\sigma > 0$, $\mathcal{G}_\sigma$ takes as input $\boldsymbol{\theta}$ and returns $\boldsymbol{g} = \nabla\mathcal{L}(\boldsymbol{\theta}) + \sigma\boldsymbol{v}$, where $\nabla\mathcal{L}(\boldsymbol{\theta})$ is the gradient of $\mathcal{L}$ at $\boldsymbol{\theta}$, $\boldsymbol{v}$ is the gradient noise drawn from the probability distribution $\mathcal{Z}_\sigma(\boldsymbol{\theta})$ with mean zero and covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$. The matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ is independent of the noise scale σ. Slightly abusing the notation, we also use $\mathcal{G}_\sigma(\boldsymbol{\theta})$ to denote the distribution of $\boldsymbol{g}$ given σ and $\boldsymbol{\theta}$.

In our work we invoke NGOS with different σ for different magnitudes of the learning rate, so that we can augment the noise level when the learning rates are set smaller. The scaling is discussed after Lemma 2.4. We now instantiate the SGD and SGDM trajectories under this noise oracle.

Definition 2.2 (Vanilla SGD)**.** Given a stochastic gradient oracle $\mathcal{G}_\sigma$, SGD with the learning rate schedule $\{\bar{\eta}_k\}$ updates the parameters $\boldsymbol{z}_k \in \mathbb{R}^d$ from initialization $\boldsymbol{z}_0$, as

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k - \bar{\eta}_k\boldsymbol{g}_k, \qquad \boldsymbol{g}_k \sim \mathcal{G}_\sigma(\boldsymbol{z}_k). \tag{2}$$

Definition 2.3 (SGD with momentum/SGDM)**.** Given oracle $\mathcal{G}_\sigma$, SGDM with the hyperparameter schedule $\{(\eta_k, \beta_k)\}$, where $\beta_k \in (0, 1)$, updates the parameters $\boldsymbol{x}_k \in \mathbb{R}^d$ from $(\boldsymbol{m}_0, \boldsymbol{x}_0)$, as

$$\boldsymbol{m}_{k+1} = \beta_k\boldsymbol{m}_k + (1-\beta_k)\boldsymbol{g}_k, \qquad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta_k\boldsymbol{m}_{k+1}, \qquad \boldsymbol{g}_k \sim \mathcal{G}_\sigma(\boldsymbol{x}_k). \tag{3}$$

Notice that the formulation of SGDM in Definition 2.3 is different from (1). An easy conversion is given by rewriting Equation (3) as:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \eta_k(1-\beta_k)\boldsymbol{g}_k + \beta_k\frac{\eta_k}{\eta_{k-1}}(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}).$$

Then setting $\eta_k = \frac{\gamma}{1-\beta}$ and $\beta_k = \beta$ recovers the form of (1). η_k is arguably a more natural parameterization that is under the same scale of the learning rates of SGD for comparison.

Modeling the gradient noise as an NGOS gives us the flexibility to scale the noise in our theoretical setting to make the effect of noise non-vanishing in small learning rate training. This is motivated by the following variant of the standard descent lemma, which highlights noise-induced and curvature-induced factors that prevent the loss to decrease:

Lemma 2.4 (Descent Lemma for SGD)**.** *Given $\boldsymbol{z}_k$, the expected change of loss in the next step is*

$$\mathbb{E}[\mathcal{L}(\boldsymbol{z}_{k+1})|\boldsymbol{z}_k] - \mathcal{L}(\boldsymbol{z}_k) =$$
$$\underbrace{-\eta\left\|\nabla\mathcal{L}(\boldsymbol{z}_k)\right\|^2}_{\textit{descent force}} + \underbrace{\frac{1}{2}(\sigma\eta)^2\operatorname{tr}((\nabla^2\mathcal{L})\boldsymbol{\Sigma}(\boldsymbol{z}_k))}_{\textit{noise-induced}} + \underbrace{\frac{1}{2}\eta^2(\nabla\mathcal{L}^\top(\nabla^2\mathcal{L})\nabla\mathcal{L}(\boldsymbol{z}_k))}_{\textit{curvature-induced}} + o(\eta^2, (\sigma\eta)^2).$$

When scaling down the learning rate, the descent force scales with $O(\eta)$ and the noise-induced impact scales with $O(\sigma^2\eta^2)$, therefore we need to set $\sigma = 1/\sqrt{\eta}$ to ensure that noises maintain the same scale of effect across different scales of η. When $\eta \to 0$ under this scaling, if we consider running the updates for $O(1/\eta)$ steps, only the curvature-induced impact is vanishing among the three.

3 WEAK APPROXIMATION OF SGDM BY SGD IN $O(1/\eta)$ STEPS

In the following two sections, we will present our main theoretical results on SGDM with small learning rates. In this section, we show that in $O(1/\eta)$ steps, SGD approximates SGDM in the sense of Definition 3.4. The next section studies SGDM over a longer training time (i.e., $O(1/\eta^2)$ steps) to characterize generalization and finds that the limiting dynamics of SGDM and SGD coincide.

3.1 A WARM UP EXAMPLE: THE VARIANCE REDUCTION EFFECT OF MOMENTUM

Intuitively, momentum makes the SGD update direction less noisy by averaging past stochastic gradients, which seems at first glance to contradict our result that the distribution of SGD and SGDM at any time point are approximately the same. However, the apparent discrepancy can be explained by the following effect: by carrying the noise at a step to subsequent steps, the updates of SGDM have long-range correlations.

To illustrate this, let us consider the case where the stochastic gradients are i.i.d. gaussian as $\boldsymbol{g}_k \sim \mathcal{N}(\boldsymbol{c}, \boldsymbol{I})$ for a constant vector $\boldsymbol{c}$. We compare SGD and SGDM trajectories with hyperparameter $\eta_k = \eta$ and $\beta_k = \beta$, and initialization $\boldsymbol{z}_0 = \boldsymbol{x}_0$ and $\boldsymbol{m}_0 \sim \mathcal{N}(\boldsymbol{c}, \frac{1-\beta}{1+\beta}\boldsymbol{I})$. The single-step updates are

$$\boldsymbol{z}_{k+1} - \boldsymbol{z}_k = -\eta \boldsymbol{g}_k \sim \mathcal{N}(-\eta \boldsymbol{c}, \eta^2 \boldsymbol{I}).$$
$$\boldsymbol{x}_{k+1} - \boldsymbol{x}_k = -\eta \boldsymbol{m}_{k+1} = -\eta(\beta^{k+1}\boldsymbol{m}_0 + \sum_{s=0}^{k} \beta^{k-s}(1-\beta)\boldsymbol{g}_s) \sim \mathcal{N}(-\eta \boldsymbol{c}, \frac{1-\beta}{1+\beta}\eta^2 \boldsymbol{I}).$$

Therefore, the variance of each single-step update is reduced by a factor of $\frac{1-\beta}{1+\beta}$, which implies larger momentum generates a smoother trajectory. Furthermore, measuring the turbulence over a fixed interval via the path length $\sum_k \|\boldsymbol{z}_{k+1} - \boldsymbol{z}_k\|_2$ or the loss variation $\sum_k |\mathcal{L}(\boldsymbol{z}_{k+1}) - \mathcal{L}(\boldsymbol{z}_k)|$ indeed suggests that adding momentum smooths the path.

However, we are usually more interested in tracking the final loss distributions induced by each trajectory. The distributions of after k steps are

$$\boldsymbol{z}_k \sim \mathcal{N}(\boldsymbol{z}_0 - k\eta \boldsymbol{c}, k\eta^2 \boldsymbol{I});$$
$$\boldsymbol{x}_k = \boldsymbol{z}_0 - \eta\beta\frac{1-\beta^k}{1-\beta}\boldsymbol{m}_0 - \eta\sum_{s=0}^{k-1}(1-\beta^{k-s})\boldsymbol{g}_s \sim \mathcal{N}\left(\boldsymbol{z}_0 - k\eta \boldsymbol{c}, k\eta^2 \boldsymbol{I} - 2\beta\eta^2\frac{1-\beta^k}{1-\beta^2}\boldsymbol{I}\right).$$

Notice that the variance of the final endpoint is only different by $|2\beta\eta^2\frac{1-\beta^k}{1-\beta^2}| \leq \frac{2\eta^2}{1-\beta}$, which is bounded regardless of k. The variance is increased at rate η^2 per step, which is significantly larger than the per step update variance $\frac{1-\beta}{1+\beta}\eta^2$. As such, the turbulence of the path may not faithfully reflect the true stochasticity of the iterates.

3.2 MAIN RESULTS ON WEAK APPROXIMATIONS OF SGDM

We study the setting where the magnitude of the hyperparameters are controlled. Let η be the scale of the learning rate so that $\eta_k = O(\eta)$. Furthermore, we set an index $\alpha \geq 0$ so that the decay rate of the momentum is controlled as $1 - \beta_k = O(\eta^\alpha)$. $\alpha = 0$ corresponds to a constant decay schedule while $\alpha > 0$ corresponds to a schedule where we make β_k closer to 1 when the learning rates are getting smaller. More formally, we associate a hyperparameter schedule $(\eta_k^\eta, \beta_k^\eta)$ for each scale η such that the following assumption is satisfied.

Definition 3.1 (Hyperparameter Schedule Scaling)**.** A family of hyperparameter schedules $\{\eta_k^{(n)}, \beta_k^{(n)}\}_{k\geq 1}$ is scaled by $\eta^{(n)}$ with index α if there are constants $\eta_{\max}, \lambda_{\min}$ and $\lambda_{\max}$ independent of n, such that for all n

$$0 \leq \frac{\eta_k^{(n)}}{\eta^{(n)}} < \eta_{\max}, \quad 0 < \lambda_{\min} \leq \frac{1-\beta_k^{(n)}}{(\eta^{(n)})^\alpha} \leq \lambda_{\max} < 1.$$

We need some boundedness of the initial momentum for the SGDM trajectory to start safely.

Assumption 3.2 (Boundedness of the Initial Momentum)**.** For each $m \geq 1$, there is constant $C_m \geq 0$ that $\mathbb{E}(\|\boldsymbol{m}_0\|_2^m) \leq C_m$;

Following Malladi et al. (2022), we further assume that the NGOS satisfies the below conditions, which make the trajectory amenable to analysis. We say a function $g(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}^m$ has polynomial growth if there are constants $k_1, k_2 > 0$ such that $\|g(\boldsymbol{x})\|_2 \leq k_1(1 + \|\boldsymbol{x}\|_2^{k_2}), \forall \boldsymbol{x} \in \mathbb{R}^d$.

Assumption 3.3. The NGOS $\mathcal{G}_\sigma = (\mathcal{L}, \boldsymbol{\Sigma}, \mathcal{Z}_\sigma)$ satisfies the following conditions.

1. **Well-Behaved**: $\nabla\mathcal{L}$ is Lipschitz and $\mathcal{C}^\infty$-smooth; $\boldsymbol{\Sigma}^{1/2}$ is bounded, Lipschitz, and $\mathcal{C}^\infty$-smooth; all partial derivatives of $\nabla\mathcal{L}$ and $\boldsymbol{\Sigma}^{1/2}$ up to and including the third order have polynomial growth.
2. **Bounded Moments**: For all integers $m \geq 1$ and all noise scale parameters σ, there exists a constant C_{2m} (independent of σ) such that $(\mathbb{E}_{\boldsymbol{v}\sim\mathcal{Z}_\sigma(\boldsymbol{\theta})}[\|\boldsymbol{v}\|_2^{2m}])^{\frac{1}{2m}} \leq C_{2m}(1 + \|\boldsymbol{\theta}\|_2), \forall \boldsymbol{\theta} \in \mathbb{R}^d$.

Besides, to rigorously discuss the closeness between dynamics of different algorithms, we introduce the following notion of approximation between two discrete trajectories, inspired by (Li et al., 2019).

Definition 3.4 (Order-γ Weak Approximation)**.** Two families of discrete trajectories $\boldsymbol{x}_k^\eta$ and $\boldsymbol{y}_k^\eta$ are weak approximations of each other, if there is $\eta_{\text{thr}} > 0$ that for any $T > 0$, any function h of polynomial growth, and any $\eta \leq \eta_{\text{thr}}$, there is a constant $C_{h,T}$ independent of η such that,

$$\max_{k=0,\ldots,\lfloor T/\eta \rfloor} |\mathbb{E}h(\boldsymbol{x}_k^\eta) - \mathbb{E}h(\boldsymbol{y}_k^\eta)| \leq C_{h,T} \cdot \eta^\gamma.$$

Weak approximation implies that $\boldsymbol{x}_k^\eta$ and $\boldsymbol{y}_k^\eta$ have similar distributions at any step k, and specifically in the deep learning setting it implies that both training (or testing) curves are similar. Given the above definitions, we are ready to establish our main result.

Theorem 3.5 (Weak Approximation of SGDM by SGD)**.** *Fix the initial point $\boldsymbol{x}_0$, $\alpha \in [0, 1)$, and an NGOS satisfying Assumption 3.3. Consider the SGDM update $\boldsymbol{x}_k^\eta$ with hyperparameter schedule $\{\eta_k, \beta_k\}_{k\geq 1}$ scaled by η with index α, noise scaling $\sigma \leq \eta^{-1/2}$ and initialization $(\boldsymbol{m}_0, \boldsymbol{x}_0)$ satisfying Assumption 3.2, then $\boldsymbol{x}_k^\eta$ is an order-$(1-\alpha)/2$ weak approximation (Definition 3.4) of the trajectories $\boldsymbol{z}_k^\eta$ with initialization $\boldsymbol{z}_0^\eta = \boldsymbol{x}_0$, noise scaling σ and an averaged learning rate schedule $(\bar{\eta}_k = \sum_{s=k}^\infty \eta_s \prod_{\tau=k+1}^s \beta_\tau(1-\beta_k))$.*

Specifically, for a constant schedule where $(\eta_k = \eta, \beta_k = \beta)$ and $\bar{\eta}_k = \eta$, SGD and SGDM with the same learning rate weakly approximate each other with distance $O(\sqrt{\eta/(1-\beta)})$.

The theorem shows that when the learning rate has a small scale η and the momentum is constant (*e.g.*, 0.9), then the trajectory of SGDM and SGD will be close in distribution over $O(1/\eta)$ steps, when the gradient noise is amplified at a scale no more than $\eta^{-1/2}$. Specifically when we consider the limit $\eta \to 0$, then the trajectories of SGDM and SGD will have the same distribution. Following (Li et al., 2019), the limiting distribution can be described by the law of the solution $\boldsymbol{X}_t$ to an SDE $\mathrm{d}\boldsymbol{X}_t = -\lambda_t\nabla\mathcal{L}(\boldsymbol{X}_t)\mathrm{d}t + \lambda_t\boldsymbol{\Sigma}^{1/2}(\boldsymbol{X}_t)\mathrm{d}\boldsymbol{W}_t$ for some rescaled learning rate schedule λ_t.

Our theorem requires $\alpha \in [0, 1)$, and the approximation grows weaker as α approaches 1. When $\alpha = 1$, the two trajectories are no longer weak approximations of each other, and their trajectories will have different limiting distributions. Furthermore, when $\alpha > 1$, yet another behaviour of the SGDM trajectory occurs over a longer range of $O(\eta^{-\frac{1+\alpha}{2}})$ steps. This is often undesirable in practice as optimization is slower.

4 The Limit of SGDM and SGD are Identical in $O(1/\eta^2)$ Steps

In this section, we follow the framework from (Li et al., 2021b) to study the dynamics of SGDM when the iterates are close to some manifold of local minimizers of $\mathcal{L}$. Former analyses (e.g., (Yan et al., 2018)) suggest that SGDM and SGD will get close to a local minimizer in $O(1/\eta)$ steps, at which point the loss function plateaus and the trajectory follows a diffusion process near the local minimizer. If the local minimizers form an manifold in the parameter space, then the diffusion accumulates into a drift within the manifold in $O(1/\eta^2)$ steps. (Li et al., 2021b) shows that the drift induces favorable generalization properties after the training loss reaches its minimum under certain circumstances.

Therefore, we hope to study the generalization effect of SGDM by investigating its dynamics in such a regime. In particular, we show that when $\eta \to 0$ the limiting diffusion of SGDM admits the same form as that of SGD, thus suggesting that momentum provides no generalization benefits.

4.1 Preliminaries on manifold of local minimizers

We consider the case where the local minimizers of the loss $\mathcal{L}$ form a manifold.

Assumption 4.1. $\mathcal{L}$ is smooth. Γ is a $(d-M)$-dimensional submanifold of $\mathbb{R}^d$ for some integer $0 \le M \le d$. Moreover, every $\boldsymbol{x} \in \Gamma$ is a local minimizer of $\mathcal{L}$, satisfying $\nabla \mathcal{L}(\boldsymbol{x}) = 0$ and $\mathrm{rank}(\nabla^2 \mathcal{L}(\boldsymbol{x})) = M$.

We consider a neighborhood O_Γ of Γ that is an attraction set under $\nabla \mathcal{L}$. Specifically, we define the gradient flow under $\nabla \mathcal{L}$ by $\phi(\boldsymbol{x}, t) = \boldsymbol{x} - \int_0^t \nabla \mathcal{L}(\phi(\boldsymbol{x}, s))\mathrm{d}s$ for any $\boldsymbol{x} \in \mathbb{R}^d$ and $t \ge 0$. We further define gradient projection map associated with $\nabla \mathcal{L}$ as $\Phi(\boldsymbol{x}) := \lim_{t\to\infty} \phi(\boldsymbol{x}, t)$ when the limit exists. Formally, we make the following assumption:

Assumption 4.2. For any initialization $\boldsymbol{x} \in O_\Gamma$, the gradient flow governed by $\nabla \mathcal{L}$ converges to some point in Γ, i.e., $\Phi(\boldsymbol{x})$ is well-defined and $\Phi(\boldsymbol{x}) \in \Gamma$.

It can be shown that for every $\boldsymbol{x} \in \Gamma$, $\partial\Phi(\boldsymbol{x})$ is the orthogonal projection onto the tangent space of Γ at $\boldsymbol{x}$. Moreover, (Li et al., 2021b) proved that for any initialization $\boldsymbol{x}_0 \in O_\Gamma$, a fixed learning rate schedule $\eta_k \equiv \eta$, and any $t > 0$, and time-rescaled SGD iterates $\boldsymbol{z}_{\lfloor t/\eta^2 \rfloor}$ converges in distribution to $\boldsymbol{Z}_t$, the solution to the following slow SDE on Γ:

$$\boldsymbol{Z}_t = \Phi(\boldsymbol{x}_0) + \int_0^t \partial\Phi(\boldsymbol{Z}_s)\boldsymbol{\Sigma}^{1/2}(\boldsymbol{Z}_s)\mathrm{d}W_s + \int_0^t \frac{1}{2}\partial^2\Phi(\boldsymbol{Z}_s)[\boldsymbol{\Sigma}(\boldsymbol{Z}_s)]\mathrm{d}s.$$

4.2 Analysis of SGDM via slow SDE

In this regime, for a fixed $\alpha \ge 0$, we choose a series of learning rate scales $\eta^{(0)} > \eta^{(1)} > \cdots > 0$ with $\lim_{n\to\infty} \eta^{(n)} = 0$. For each n, we assign a hyperparameter schedules $\{(\eta_k^{(n)}, \beta_k^{(n)})\}_{k\ge 1}$, such that $\{(\eta_k^{(n)}, \beta_k^{(n)})\}_{k\ge 1}$ is scaled by $\eta^{(n)}$ in the sense of Definition 3.1.

For SGD with a fixed learning rate, as shown in (Li et al., 2021b), it suffices to consider a fixed time rescaling by looking at $\boldsymbol{z}_{\lfloor t/\eta^2 \rfloor}$ to derive the limiting dynamics, i.e., one unit of time for the slow SDE on Γ corresponds to $\lfloor 1/\eta^2 \rfloor$ SGD steps. However, the varying learning rate case requires more care to align the discrete iterates with the slow dynamics on Γ. As such, we consider learning rate schedules over time horizon T, which corresponds to $K^{(n)} = \lfloor T/(\eta^{(n)})^2 \rfloor$ steps of discrete updates of the process $\{\boldsymbol{z}_k^{(n)}\}$ and $\{\boldsymbol{x}_k^{(n)}\}$. To show the dynamics of SGDM and SGD have a limit at $n \to \infty$, it is necessary that the hyperparameter schedules have a limit as $n \to \infty$, which we formalize below.

Assumption 4.3 (Converging hyperparameter scheduling)**.** There exists learning rate schedule $\lambda_t : [0, T] \to \mathbb{R}^+$ with finite variation such that

$$\lim_{n\to\infty} \eta^{(n)} \sum_{k=0}^{K^{(n)}} |\eta_k^{(n)} - \eta^{(n)} \cdot \lambda_{k(\eta^{(n)})^2}| = 0.$$

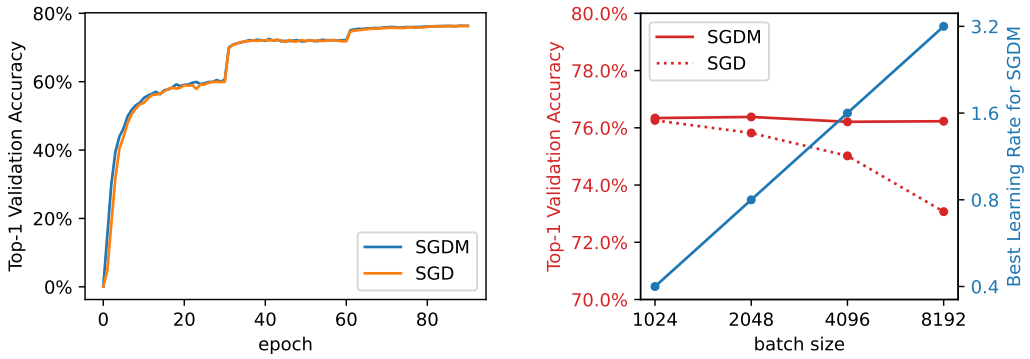
In the special case $\eta_k^{(n)} \equiv \eta^{(n)}$, it is clear that $\lambda_t \equiv 1$, which recovers the regime in (Li et al., 2021b). We furthermore assume that the hyperparameter schedules admit some form of continuity:

Assumption 4.4 (Bounded variation)**.** There is constant Q independent of n such that

$$\sum_{k=1}^{K^{(n)}} |\eta_k^{(n)} - \eta_{k-1}^{(n)}| \le Q\eta^{(n)}, \quad \sum_{k=1}^{K^{(n)}} |\beta_k^{(n)} - \beta_{k-1}^{(n)}| \le Q(\eta^{(n)})^\alpha$$

In this general regime, we define the slow SDE on Γ to admit the following description:

$$\boldsymbol{X}_t = \Phi(\boldsymbol{x}_0) + \int_0^t \lambda_t \partial\Phi(\boldsymbol{X}_s)\boldsymbol{\Sigma}^{1/2}(\boldsymbol{X}_s)\mathrm{d}W_s + \int_0^t \frac{\lambda_t^2}{2}\partial^2\Phi(\boldsymbol{X}_s)[\boldsymbol{\Sigma}(\boldsymbol{X}_s)]\mathrm{d}s. \tag{4}$$



(a) SGDM and SGD with batch size 1024

(b) SGDM v.s. SGD across batch sizes

Figure 1: SGDM performs comparably to SGD in training ResNet-50 on ImageNet with smaller batch sizes (e.g., 1024), and outperforms SGD significantly at larger batch sizes.

Indeed, we show that both SGDM and SGD, under the corresponding hyperparameter schdules, converge to the above slow SDE on Γ, as summarized in the following theorem.

Theorem 4.5. *Fix the initialization $\boldsymbol{x}_0 = \boldsymbol{z}_0 \in O_\Gamma$ and any $\alpha \in (0,1)$, and suppose the initial momentum $\boldsymbol{m}_0$ satisfies Assumption 3.2. For $n \geq 1$, let $\{(\eta_k^{(n)}, \beta_k^{(n)})\}_{k\geq 1}$ be any hyperparameter schedule scaled by $\eta^{(n)}$ satisfying Assumptions 4.3 and 4.4. Further fix the noise scale $\sigma^{(n)} \equiv 1$. Under Assumptions 4.1 and 4.2, consider the SGDM trajectory $\{\boldsymbol{x}_k^{(n)}\}_{k\geq 1}$ with hyperparameter schedule $\{(\eta_k^{(n)}, \beta_k^{(n)})\}_{k\geq 1}$ and initialization $(\boldsymbol{x}_0, \boldsymbol{m}_0)$, and the SGD trajectory $(\boldsymbol{z}_k^{(n)})$ and initialization $\boldsymbol{z}_0 = \boldsymbol{x}_0$. Suppose the slow SDE defined in (4) has a global solution $\{\boldsymbol{X}_t\}_{t\geq 0}$, then as $n \to \infty$ with $\eta^{(n)} \to 0$, both $\boldsymbol{x}^{(n)}_{\lfloor t/(\eta^{(n)})^2 \rfloor}$ and $\boldsymbol{z}^{(n)}_{\lfloor t/(\eta^{(n)})^2 \rfloor}$ converge in distribution to $\boldsymbol{X}_t$.*

The proof of Theorem 4.5 is inspired by (Calzolari and Marchetti, 1997). In this regime, the momentum process $\boldsymbol{m}_k^{(n)}$ behaves like an Uhlenbeck-Ornstein process with $O(\eta^\alpha)$ mixing variance, so the per-step variance will be significantly smaller than that of SGD, analogous to Section 3.1. Therefore a more careful expansion of the per-step change $\Phi(\boldsymbol{x}_{k+1}) - \Phi(\boldsymbol{x}_k)$ is needed. Tools from the semi-martingale analysis and weak limit results of stochastic integrals complete our proof.

5 Experiments

Our theoretical results in the previous sections mostly work for learning rates that are asymptotically small. In this section, we verify that momentum indeed has limited benefits in practical training regimes where the optimal learning rate is not very large. Additional details are in the appendix.

5.1 Momentum may indeed have marginal value in practice

ImageNet Experiments. First, we train ResNet-50 on ImageNet across batch sizes. Following the experimental setup in Goyal et al. (2017), we use a learning rate schedule that starts with a 5-epoch linear warmup to the peak learning rate and decays it at epoch #30, #60, #80. For SGDM (1), we use the default value of $\beta = 0.9$, and grid search for the best learning rate γ over 0.1×2^k ($k \in \mathbb{Z}$). Then we check whether vanilla SGD with learning rate $\frac{\gamma}{1-\beta}$ can achieve the same performance as SGDM. Consistent with previous empirical studies (Shallue et al., 2019; Smith et al., 2020), we observed that for training with smaller batch sizes, the optimal learning rate of SGDM is small enough so that SGD can perform comparably, though SGDM can indeed outperform SGD at larger batch sizes.

Language Model Experiments. In fine-tuning a pre-trained model, a small learning rate is also preferable to retain the model's knowledge learned during pre-training. Indeed, we observe that SGD and SGDM behave similarly in this case. We fine-tune RoBERTa-large (Liu et al., 2019) on 5 diverse tasks (SST-2 (Socher et al., 2013), SST-5 (Socher et al., 2013), SNLI (Bowman et al., 2015), TREC (Voorhees and Tice, 2000), and MNLI (Williams et al., 2018)) using SGD and SGDM. We follow the few shot setting described in (Gao et al., 2021; Malladi et al., 2023), using a grid for SGD

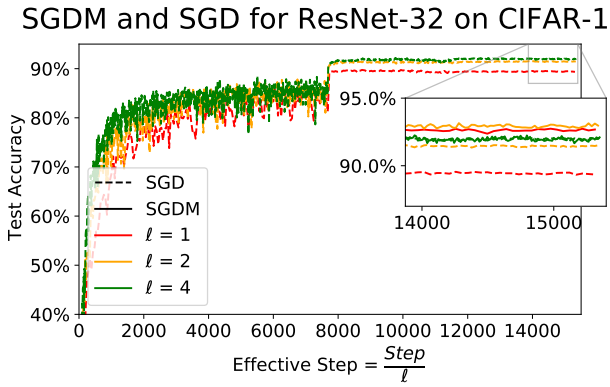


Figure 2: Standard SGDM achieves higher test performance than SGD (see $\ell = 1$), but the two trajectories get closer when reducing the curvature-induced term with SVAG (i.e., increasing the value of ℓ, see Definition 5.1 and Lemma 2.4). These experiments confirm our theoretical findings that SGD and SGDM approximate each other when the gradient noise is the primary source of instability. We use batch size $B = 512$ with two learning rate decays by a factor of 0.1 at epochs 80 and 120. We grid search to find the best learning rate for SGDM ($\eta = 0.2$) and then use it to run SGD and SGDM with SVAG. We use $\beta = 0.9$ for SGDM. Additional experimental details are in the appendix.

based on (Malladi et al., 2023) and sampling 512 examples per class (Table 1). Additional settings and trajectories are in **??**.

Table 1: SGD and SGDM for fine-tuning RoBERTa-large on 5 tasks using 512 examples from each class (Gao et al., 2021; Malladi et al., 2023). Results are averaged over 5 random subsets of the full dataset. These findings confirm that SGD and SGDM approximate each other in noisy settings.

Task	**SST-2**	**SST-5**	**SNLI**	**TREC**	**MNLI**
Zero-shot	79.0	35.5	50.2	51.4	48.8
SGD	94.0 (0.4)	55.2 (1.1)	87.7 (0.3)	97.2 (0.2)	84.0 (0.3)
SGDM	94.0 (0.5)	55.0 (1.0)	88.4 (0.6)	97.2 (0.4)	83.7 (0.8)

5.2 Investigating the benefit of momentum in large-batch training

The ImageNet experiments demonstrate that momentum indeed offers benefits in large-batch training when the optimal learning rate is relatively large. We now use large-batch training experiments on CIFAR-10 to provide empirical evidence that this benefit may not be due to the noise reduction effect. We apply SVAG (Li et al., 2021a) to control the noise scale in our experiments and reduce the curvature-induced training instability (Lemma 2.4) while leaving the noise-induced term unchanged.

Definition 5.1 (SVAG)**.** With any $\ell > 0$, SVAG transforms the NGOS $\mathcal{G}_\sigma = (f, \mathbf{\Sigma}, \mathcal{Z}_\sigma)$ (Definition 2.1) into another NGOS $\widehat{\mathcal{G}}_{\sqrt{\ell}\sigma} = (f, \mathbf{\Sigma}, \widehat{\mathcal{Z}}_{\sqrt{\ell}\sigma})$ with scale $\sqrt{\ell}\sigma$. For an input $\boldsymbol{\theta}$, $\widehat{\mathcal{G}}_{\ell\sigma}$ returns $\widehat{\boldsymbol{g}} = r_1(\ell)\boldsymbol{g}_1 + r_2(\ell)\boldsymbol{g}_2$ where $\boldsymbol{g}_1, \boldsymbol{g}_2 \sim \mathcal{G}_\sigma(\boldsymbol{\theta})$ and $r_i(\ell) = \frac{1}{2}(1 + (-1)^i\sqrt{2\ell - 1})$. $\widehat{\mathcal{Z}}_{\sqrt{\ell}\sigma}$ is defined to ensure $\widehat{\boldsymbol{g}}$ has the same distribution as $\nabla f(\boldsymbol{\theta}) + \sqrt{\ell}\sigma\boldsymbol{z}$ when $\boldsymbol{z} \sim \widehat{\mathcal{Z}}_{\sqrt{\ell}\sigma}(\boldsymbol{\theta})$.

In our experiments, we divide the learning rate η by ℓ after applying SVAG so $\lambda' = \lambda/\sqrt{\ell}$, and run ℓ times the original iterate steps. This ensures that way the noise-induced impact and the descent force stay the same scale in Lemma 2.4, while the curvature-induced impact is reduced by a factor of ℓ.

We train a ResNet-32 (He et al., 2016) on CIFAR-10 (Krizhevsky et al.) with batch size $B = 512$. In order to control the curvature-induced impact, we apply SVAG (Li et al., 2021a; Malladi et al., 2022) to the NGOS (Definition 2.1) for SGD and SGDM. We first grid search to find the best learning rate for the standard SGDM ($\ell = 1$), and then we perform SGD and SGDM with that same learning rate for different levels of ℓ. The results are summarized in Figure 2. We see that standard SGDM outperforms standard SGD, but when we increase the noise level ℓ, the two trajectories become closer.

6 Related Works

The role of momentum in optimization. The accelerating effect of some variants of momentum has been observed in convex optimization (Kidambi et al., 2018) and linear regression (Jain et al., 2018) with under specialized parametrizations. Smith (2018) pointed out that momentum can help stabilize training, but the optimal choice of momentum is closely related to the choice of learning rate. Plattner (2022) later empirically established that momentum enlarges the learning rate but does not boost performance. Arnold et al. (2019) argued using a quadratic example that momentum might not reduce variance as the gradient noise in each would actually be carried over to future iterates due to momentum. Tondji et al. (2021) showed that the application of a multi-momentum strategy can achieve variance reduction in deep learning.

Defazio (2020) proposed a stochastic primal averaging formulation for SGDM which facilitates a Lyapunov analysis for SGDM, and one particular insight from their analysis is that momentum may help reduce noise in the early stage of training but is no longer helpful when the iterates are close to local minima. Xie et al. (2021) showed that under SDE approximation, the posterior of SGDM is the same as that of SGD. Jelassi and Li (2022) proved the generalization benefit of momentum in GD in a specific setting of binary classification, by showing that GD+M is able to learn small margin data from the historical gradients in the momentum. A stronger implicit regularization effect of momentum in GD is also proved in Ghosh et al. (2023).

Convergence of momentum methods. Momentum-based methods do not tend to yield faster convergence rates in theory. Yu et al. (2019) showed that distributed SGDM can achieve the same linear speedup as ditributed SGD in the non-convex setting. Also in the non-convex setting, Yan et al. (2018) showed that the gradient norm converges at the same rate for SGD, SGDM and stochastic Nestrov's accelerated gradient descent, and they used stability analysis to argue that momentum helps generalization when the loss function is Lipschitz. Under the formulation of quasi-hyperbolic momentum (Ma and Yarats, 2019), Gitman et al. (2019) proposed another unified analysis for momentum methods.Liu et al. (2020) proved that SGDM converges as fast as SGD for strongly convex and non-convex objectives even without a bounded gradient assumption. Using a iterate-averaging formulation, Sebbouh et al. (2021) proved last-iterate convergence of SGDM in both convex and non-convex settings. Later, (Li et al., 2022a) showed that constant momentum can lead to suboptimal last-iterate convergence rate and increasing momentum resolves the issue. Smith (2018); Liu et al. (2018) provided evidence that momentum helps escape saddle points.

Characterizing implicit bias near manifold of local minimizers A recent line of work has studied the implicit bias induced by gradient noise in SGD-type algorithms, when iterates are close to some manifold of local minimizers (Blanc et al., 2020; Damian et al., 2021; Li et al., 2021b). In particular, Li et al. (2021b) developed a framework for describing the dynamics of SGD via a slow SDE on the manifold of local minimizers in the regime of small learning rate (see **??** for an introduction). Similar methodology has become a powerful tool for analyzing algorithmic implicit bias and has been extended to many other settings, including SGD/GD for models with normalization layers (Lyu et al., 2022; Li et al., 2022b), GD in the edge of stability regime (Arora et al., 2022), Local SGD (Gu et al., 2023), sharpness-aware minimization (Wen et al., 2022), and pre-training for language models (Liu et al., 2022). Notably, Cowsik et al. (2022) utilized the similar idea to study the slow SDE of SGDM study the optimal scale of the momentum parameter with respect to the learning rate, which has a focus different from our paper.

7 Conclusions

This work provides theoretical characterizations of the role of momentum in stochastic gradient methods. We formally show that momentum does not introduce optimization and generalization benefits when the learning rates are small, and we further exhibit empirically that the value of momentum is marginal for gradient-noise-dominated learning settings with practical learning rate scales. Hence we conclude that momentum does not provide a significant performance boost in the above cases. Our results further suggest that model performance is agnostic to the choice of momentum parameters over a range of hyperparameter scales.

REFERENCES

Sébastien Arnold, Pierre-Antoine Manzagol, Reza Babanezhad Harikandeh, Ioannis Mitliagkas, and Nicolas Le Roux. Reducing the variance in online optimization by transporting past gradients. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Sanjeev Arora, Zhiyuan Li, and Abhishek Panigrahi. Understanding gradient descent on the edge of stability in deep learning. In *International Conference on Machine Learning*, pages 948–1024. PMLR, 2022.

Guy Blanc, Neha Gupta, Gregory Valiant, and Paul Valiant. Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 483–513. PMLR, 09–12 Jul 2020.

Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018. doi: 10.1137/16M1080173.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://aclanthology.org/D15-1075.

Antonella Calzolari and Federico Marchetti. Limit motion of an ornstein–uhlenbeck particle on the equilibrium manifold of a force field. *Journal of Applied Probability*, 34(4):924–938, 1997.

Aditya Cowsik, Tankut Can, and Paolo Glorioso. Flatter, faster: scaling momentum for optimal speedup of sgd. *arXiv preprint arXiv:2210.16400*, 2022.

Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Alex Damian, Tengyu Ma, and Jason D. Lee. Label noise SGD provably prefers flat global minimizers. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

Aaron Defazio. Momentum via primal averaging: theoretical insights and learning rate schedules for non-convex optimization. *arXiv preprint arXiv:2010.00406*, 2020.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, 2021. doi: 10.18653/v1/2021.acl-long.295. URL https://aclanthology.org/2021.acl-long.295.

Avrajit Ghosh, He Lyu, Xitong Zhang, and Rongrong Wang. Implicit regularization in heavy-ball momentum accelerated stochastic gradient descent. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ZzdBhtEH9yB.

Igor Gitman, Hunter Lang, Pengchuan Zhang, and Lin Xiao. Understanding the role of momentum in stochastic gradient methods. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Xinran Gu, Kaifeng Lyu, Longbo Huang, and Sanjeev Arora. Why (and when) does local SGD generalize better than SGD? In *The Eleventh International Conference on Learning Representations*, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Prateek Jain, Sham M. Kakade, Rahul Kidambi, Praneeth Netrapalli, and Aaron Sidford. Accelerating stochastic gradient descent for least squares regression. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 545–604. PMLR, 06–09 Jul 2018.

Samy Jelassi and Yuanzhi Li. Towards understanding how momentum improves generalization in deep learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 9965–10040. PMLR, 17–23 Jul 2022.

Gary Shon Katzenberger. Solutions of a stochastic differential equation forced onto a manifold by a large drift. *The Annals of Probability*, pages 1587–1628, 1991.

Rahul Kidambi, Praneeth Netrapalli, Prateek Jain, and Sham M. Kakade. On the insufficiency of existing momentum schemes for stochastic optimization. In *International Conference on Learning Representations*, 2018.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. CIFAR-10 (Canadian Institute for Advanced Research). URL http://www.cs.toronto.edu/~kriz/cifar.html.

Qianxiao Li, Cheng Tai, and Weinan E. Stochastic modified equations and dynamics of stochastic gradient algorithms i: Mathematical foundations. *Journal of Machine Learning Research*, 20(40): 1–47, 2019.

Xiaoyu Li, Mingrui Liu, and Francesco Orabona. On the last iterate convergence of momentum methods. In Sanjoy Dasgupta and Nika Haghtalab, editors, *Proceedings of The 33rd International Conference on Algorithmic Learning Theory*, volume 167 of *Proceedings of Machine Learning Research*, pages 699–717. PMLR, 29 Mar–01 Apr 2022a.

Zhiyuan Li, Sadhika Malladi, and Sanjeev Arora. On the validity of modeling SGD with stochastic differential equations (sdes). *Advances in Neural Information Processing Systems*, 34:12712–12725, 2021a.

Zhiyuan Li, Tianhao Wang, and Sanjeev Arora. What happens after SGD reaches zero loss?–a mathematical framework. In *International Conference on Learning Representations*, 2021b.

Zhiyuan Li, Tianhao Wang, and Dingli Yu. Fast mixing of stochastic gradient descent with normalization and weight decay. *Advances in Neural Information Processing Systems*, 35:9233–9248, 2022b.

Hong Liu, Sang Michael Xie, Zhiyuan Li, and Tengyu Ma. Same pre-training loss, better downstream: Implicit bias matters for language models. *arXiv preprint arXiv:2210.14199*, 2022.

Tianyi Liu, Zhehui Chen, Enlu Zhou, and Tuo Zhao. A diffusion approximation theory of momentum sgd in nonconvex optimization. *arXiv preprint arXiv:1802.05155*, 2018.

Yanli Liu, Yuan Gao, and Wotao Yin. An improved analysis of stochastic gradient descent with momentum. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18261–18271. Curran Associates, Inc., 2020.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. URL https://arxiv.org/pdf/1907.11692.pdf.

Kaifeng Lyu, Zhiyuan Li, and Sanjeev Arora. Understanding the generalization benefit of normalization layers: Sharpness reduction. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=xp5VOBxTxZ.

Jerry Ma and Denis Yarats. Quasi-hyperbolic momentum and adam for deep learning. In *International Conference on Learning Representations*, 2019.

Sadhika Malladi, Kaifeng Lyu, Abhishek Panigrahi, and Sanjeev Arora. On the SDEs and scaling rules for adaptive gradient algorithms. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. A kernel-based view of language model fine-tuning, 2023.

Genevieve Beth Orr. *Dynamics and Algorithms for Stochastic Search*. PhD thesis, USA, 1996. UMI Order No. GAX96-08998.

Maximilian Plattner. On sgd with momentum. page 60, 2022. URL http://infoscience.epfl.ch/record/295398.

Boris T. Polyak. *Introduction to Optimization*. Optimization Software, Inc., 1987.

B.T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964. ISSN 0041-5553.

Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1): 145–151, 1999.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Learning Internal Representations by Error Propagation*, pages 318–362. 1987.

Othmane Sebbouh, Robert M Gower, and Aaron Defazio. Almost sure convergence rates for stochastic gradient descent and stochastic heavy ball. In Mikhail Belkin and Samory Kpotufe, editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 3935–3971. PMLR, 15–19 Aug 2021.

Christopher J. Shallue, Jaehoon Lee, Joseph Antognini, Jascha Sohl-Dickstein, Roy Frostig, and George E. Dahl. Measuring the effects of data parallelism on neural network training. *Journal of Machine Learning Research*, 20(112):1–49, 2019.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR, 10–15 Jul 2018.

Leslie N Smith. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018.

Samuel Smith, Erich Elsen, and Soham De. On the generalization benefit of noise in stochastic gradient descent. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9058–9067. PMLR, 13–18 Jul 2020.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. 2013. URL https://aclanthology.org/D13-1170.pdf.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

Lionel Tondji, Sergii Kashubin, and Moustapha Cisse. Variance reduction in deep learning: More momentum is all you need. *arXiv preprint arXiv:2111.11828*, 2021.

Mehmet Ali Tugay and Yalçin Tanik. Properties of the momentum lms algorithm. *Signal Processing*, 18(2):117–127, 1989. ISSN 0165-1684.

Ellen M Voorhees and Dawn M Tice. Building a question answering test collection. In *the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, 2000.

Kaiyue Wen, Tengyu Ma, and Zhiyuan Li. How does sharpness-aware minimization minimize sharpness? *arXiv preprint arXiv:2211.05729*, 2022.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. 2018. URL https://aclanthology.org/N18-1101.pdf.

Zeke Xie, Li Yuan, Zhanxing Zhu, and Masashi Sugiyama. Positive-negative momentum: Manipulating stochastic gradient noise to improve generalization. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11448–11458. PMLR, 18–24 Jul 2021.

Yan Yan, Tianbao Yang, Zhe Li, Qihang Lin, and Yi Yang. A unified analysis of stochastic momentum methods for deep learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 2955–2961. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training BERT in 76 minutes. In *International Conference on Learning Representations*, 2020.

Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7184–7193. PMLR, 09–15 Jun 2019.

Kun Yuan, Bicheng Ying, and Ali H Sayed. On the influence of momentum acceleration on online learning. *The Journal of Machine Learning Research*, 17(1):6602–6667, 2016.