Dynamic Momentum Recalibration in Online Gradient Learning

Zhipeng Yao [1, 2] ⋆ Rui Yu [2] † Guisong Chang [3] Ying Li [1] Yu Zhang [1] Dazhou Li [1] †
[1]Shenyang University of Chemical Technology [2]University of Louisville [3]Northeastern University
yiucp@outlook.com, rui.yu@louisville.edu, gschang@mail.neu.edu.cn
Gooddayli12358@outlook.com, zhangy@syuct.edu.cn, lidazhou@syuct.edu.cn
https://github.com/LilYau350/SGDF-Optimizer

Abstract

Stochastic Gradient Descent (SGD) and its momentum variants form the backbone of deep learning optimization, yet the underlying dynamics of their gradient behavior remain insufficiently understood. In this work, we reinterpret gradient updates through the lens of signal processing and reveal that fixed momentum coefficients inherently distort the balance between bias and variance, leading to skewed or suboptimal parameter updates. To address this, we propose SGDF (SGD with Filter), an optimizer inspired by the principles of Optimal Linear Filtering. SGDF computes an online, time-varying gain to dynamically refine gradient estimation by minimizing the mean-squared error, thereby achieving an optimal trade-off between noise suppression and signal preservation. Furthermore, our approach could extend to other optimizers, showcasing its broad applicability to optimization frameworks. Extensive experiments across diverse architectures and benchmarks demonstrate SGDF surpasses conventional momentum methods and achieves performance on par with or surpassing state-of-the-art optimizers.

1. Introduction

In deep learning optimization, the optimizer plays a pivotal role in refining model parameters to capture underlying data patterns, while strategically navigating complex loss landscapes [17] to identify regions that promote strong generalization [37]. Its choice profoundly affects training efficiency, convergence speed, generalization, and robustness to data shifts [2], with suboptimal selections potentially causing slow or failed convergence, whereas effective ones accelerate learning and bolster model resilience [59]. Thus, the design and refinement of optimizers remain essential challenges in enhancing the capabilities of models.

Stochastic Gradient Descent (SGD) [50] and its variants, including momentum-based methods [56, 63] and adaptive techniques like Adam [38] and RMSprop [30], which have advanced training efficiency [6]. However, these approaches face challenges in high-dimensional, non-convex settings [26], where adaptive methods often yield rapid convergence but suffer from poor generalization [36]. Efforts to mitigate this have led to Adam variants [7, 44, 47, 79] that refine adaptive learning rates, yet they fall short of fully closing the generalization gap, highlighting the ongoing need for innovative optimization strategies that better balance estimation accuracy and practical performance.

Actually, the issues that arise from the optimizer during training, particularly in terms of optimization and generalization, are inherently tied to the trade-off between bias and variance [23, 51]. High bias leads to underfitting, while high variance results in overfitting. Similarly, the gradients used by the optimizer to update weights also face this challenge. Intuitively, high bias in the gradients may lead to convergence at a suboptimal plateau [69, 75], while high variance can lead to instability in the optimization path, causing oscillations that hinder convergence [5, 19]. Therefore, a good optimizer should strike a balance between the bias and variance in its gradient estimates.

From a statistical signal processing perspective, we analyze the mechanism behind optimizer updates. Specifically, we decompose the optimizer's gradients used for updating model weight into bias and variance components. Then, We identify a key limitation in momentum-based optimization techniques supplemented with examining the statistical distribution of gradients within the model: they struggle to balance bias and variance components in gradients, often introducing a gradient shift phenomenon, which we term *bias gradient estimate*. This bias estimate, arising from fixed momentum coefficients, accumulates over time, leading to bias. As a result, the model may struggle to adapt to variations in curvature across different layers, resulting in suboptimal or directionally skewed updates [16, 76].

To address this issue, we introduce SGDF, a novel method that uses principles from Optimal Linear Filter to adjust gradient estimation dynamically. SGDF derives an optimal,

⋆ Work initiated at [1] and further developed at [2], † Equal advising.

time-varying gain to minimize mean-squared error in gradient estimation, balancing noise reduction with signal preservation. This filter mechanism provides a more accurate first-order gradient estimate and avoids the limitations of fixed momentum parameters, allowing SGDF to adjust dynamically throughout training. Additionally, SGDF's flexibility extends to other optimization frameworks, which enhance performance across a range of tasks. Through extensive empirical validation across diverse model architectures and visual tasks, we demonstrate that SGDF consistently outperforms traditional momentum-based and variance reduction methods, achieving competitive or superior results relative to state-of-the-art optimizers.

The main contributions can be summarized as follows:

- We quantify the bias-variance trade-off in momentum-based gradient estimation (EMA and CM) using a unified SDE framework, revealing their static limitations.
- We introduce SGDF, an optimizer that combines historical and current gradient data to estimate the gradient, addressing the trade-off between bias and variance in the momentum method.
- We theoretically analyze the convergence property of SGDF in both convex optimization and non-convex stochastic optimization (Sec. 3.3), and empirically verify the effectiveness of SGDF (Sec. 4).
- We preliminarily explore the extension of SGDF's first-moment filter estimation to other optimization framework (*e.g.*, Adam), which shows a promising enhancement in their performance (Sec. 4.2), surpassing traditional momentum-based methods.

2. The Gradient Estimation Dilemma

2.1. Bias and Variance

Stochastic gradient-based optimization lies at the core of modern machine learning. We revisit this and found that it grapples with a fundamental challenge: the trade-off between gradient bias and variance. To dissect this dilemma, we begin by unifying two prominent momentum strategies under a single framework. The proof of this section is in Supplemental Materials Appendix A.

Definition 2.1. The unified momentum update rule is defined as:

$$m_t = \beta m_{t-1} + u g_t, \quad \theta_t = \theta_{t-1} - \alpha m_t, \tag{1}$$

where α is the learning rate, $\beta \in [0, 1)$ is the momentum coefficient, and $u \geq 1-\beta$ scales the current gradient. For all $\theta \in \mathbb{R}^d$, $f_t(\theta) = f(\theta; \xi_t)$ denotes the stochastic objective at iteration t with data sample $\xi_t \sim \mathcal{D}$. The expected objective is $f(\theta) = \mathbb{E}_\xi[f(\theta; \xi)]$, and $g_t = \nabla f_t(\theta_t) + \epsilon_t$ (where $\epsilon_t \sim \mathcal{N}(0, \sigma^2 I)$) is the stochastic gradient. Specific cases include:

- $u = 1 - \beta$: Exponential Moving Average (EMA),
- $u = 1$: Classical Momentum (CM) [56, 63].

This formulation encapsulates EMA and CM, two cornerstones of gradient estimation, differing in how they weight the current gradient against historical trends. EMA through a balanced mean update, while CM aggressively incorporates the gradient direction. We dissect the nature of the two methods, quantified by the mean square error.

Lemma 2.2. *For any gradient estimator $\hat{g}_t = \mathcal{A}(g_1, ..., g_t)$, the estimation of the mean square error decomposes as:*

$$\mathbb{E}[(\hat{g}_t - \nabla f(\theta_t))^2] = \underbrace{(\mathbb{E}[\hat{g}_t] - \nabla f(\theta_t))^2}_{\text{Bias}^2} + \underbrace{(\hat{g}_t - \mathbb{E}[\hat{g}_t])^2}_{\text{Variance}} \tag{2}$$

Lemma 2.2 establishes that the error in gradient estimation arises from two sources: bias, reflecting systematic deviation from the true gradient, and variance, capturing sensitivity to stochastic fluctuations. To explore how EMA and CM navigate this trade-off, we extend prior work on stochastic differential equations (SDEs) for vanilla SGD [62], reformulating momentum in continuous time.

Theorem 2.3. *Consider the unified momentum estimator $m(t)$ defined by the stochastic differential equation (SDE) of Lemma A.4, with the solution given in Lemma A.6 (see the Supplement Materials). Let the bias be defined relative to the expected true gradient:* $\text{Bias}(m(t)) = \mathbb{E}[m(t)] - \mathbb{E}[\nabla f(\theta(t))]$. *Assuming that the gradient $\nabla f(\theta(t))$ is bounded and Lipschitz continuous, the asymptotic bounds (as $t \to \infty$) for the bias and variance of $m(t)$ as an estimator are given by:*

$$\|\text{Bias}(m(t))\|^2 \leq \left(\frac{u^2 \alpha L G}{(1-\beta)^3} + \frac{u^2 \alpha \sigma L}{\sqrt{2}(1-\beta)^{2.5}} + \left(\frac{u}{1-\beta} - 1 \right) G \right)^2, \tag{3}$$

where L is the Lipschitz constant, G bounds the gradient norm $\|\nabla f(\theta(t))\|$, and the second term inside the parenthesis explicitly captures the parameter-shift bias induced by the stochastic noise σ.

$$\text{Var}(m(t)) \leq \frac{u^2 \sigma^2}{1-\beta} + \frac{2u^2 V^2}{(1-\beta)^2}, \tag{4}$$

where σ^2 is the total variance of the stochastic gradient noise, and V^2 conservatively bounds the variance of the true gradient sequence, i.e., $\text{Var}(\nabla f(\theta(t))) \leq V^2$.

Prior analyses typically regarded momentum-based gradient estimators as unbiased under the assumption that the parameter θ_t remains stationary [10, 38, 63]. However, as θ_t evolves over training, an additional *parameter-shift bias* arises. Theorem 2.3 explicitly quantifies this effect to fill a critical gap left by prior analyses, revealing that the stochastic noise σ itself exacerbates this shift.

Table 1. Bias and variance bounds for different momentum estimators. As $\beta \to 1$, static estimators suffer from either diverging bias or diverging variance, highlighting the fundamental trade-off.

Method	Bias Bound	Variance Bound	Limit as $\beta \to 1$
SGD ($\beta = 0$)	0 (Assuming $\alpha \to 0$)	$\sigma^2 + 2V^2$	N/A
EMA ($u = 1 - \beta$)	$\left(\frac{\alpha L G}{1-\beta} + \frac{\alpha\sigma L}{\sqrt{2(1-\beta)}}\right)^2$	$(1-\beta)\sigma^2 + 2V^2$	Bias $\to \infty$, Var $\to 2V^2$
CM ($u = 1$)	$\left(\frac{\alpha L G}{(1-\beta)^3} + \frac{\alpha L\sigma}{\sqrt{2}(1-\beta)^{2.5}} + \frac{\beta G}{1-\beta}\right)^2$	$\frac{\sigma^2}{1-\beta} + \frac{2V^2}{(1-\beta)^2}$	Bias $\to \infty$, Var $\to \infty$

To illustrate the implications of this theorem, Tab. 1 summarizes the bias and variance bounds for standard momentum formulations and their asymptotic behaviors. For EMA ($u = 1 - \beta$), the initialization bias strictly vanishes. EMA effectively acts as a low-pass filter: as $\beta \to 1$, it successfully damps high-frequency stochastic noise (variance drops), but at the severe expense of unbounded bias driven by outdated gradients and accumulated noise-drift. Consequently, EMA is unbiased only under strict, often unrealistic conditions in deep learning, such as the learning rate or curvature approaching zero ($\alpha, L \to 0$). Conversely, CM ($u = 1$) exhibits an aggressive momentum nature. As shown in Tab. 1, both its bias and variance diverge sharply as $\beta \to 1$, amplifying systematic lag and noise susceptibility (CM are further discussed in the Supplemental Materials Appendix E.12).

Consider the other extreme: when $\beta = 0$, both methods reduce to vanilla SGD, minimizing momentum-induced bias but retaining the full base variance. These bounds expose a fundamental limitation of conventional optimization paradigms: static choices of u and β lock the estimator into a rigid, predetermined trade-off, rendering it ill-suited to the dynamic noise and curvature of objective functions.

This analysis reveals an inherent dilemma in momentum methods: structurally reducing variance inevitably amplifies bias, while minimizing bias exposes the estimator to higher variance. This naturally raises a fundamental question: Can we design an adaptive gain that dynamically reduces the dependence on momentum to minimize bias during low-variance phases, while heavily utilizing the momentum update to aggressively filter noise when variance is high?

3. Method

In the Sec. 2, we analyzed a challenge in momentum-based methods: how to effectively balance the bias and variance in gradient estimation. The analysis suggests that introducing a variable gain could help mitigate this dilemma by adaptively adjusting the contribution of past and current gradients. Motivated by this, and inspired by the minimum mean square error (MMSE) principle [35] in Optimal Linear Filter [34], we design a novel online gradient estimator tailored for stochastic optimization. In the following, we describe the estimator's design and theoretical grounding.

Algorithm 1 SGDF: Online Filter Estimate Gradient (element-wise).

Input: θ_0: initial parameter, $f(\theta)$: stochastic objective function
Parameter: $\{\alpha_t\}_{t=1}^T$: step size, $\{\beta_1, \beta_2\}$: attenuation coefficients, $\{\varepsilon\}$: numerical stability,$\{\gamma\}$: power scaling.
Output: θ_T: resulting parameters.

Initialize: $m_0 \leftarrow 0$, $s_0 \leftarrow 0$
while $t \leq T$ **do**
  $g_t \leftarrow \nabla f_t(\theta_{t-1})$
  $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$
  $s_t \leftarrow \beta_2 s_{t-1} + (1-\beta_2)(g_t - m_t)^2$
  $\widehat{m}_t \leftarrow \dfrac{m_t}{1-\beta_1^t}, \widehat{s}_t \leftarrow \dfrac{(1-\beta_1)(1-\beta_1^{2t})\, s_t}{(1+\beta_1)(1-\beta_2^t)}$
  $K_t \leftarrow \dfrac{\widehat{s}_t}{\widehat{s}_t + (g_t - \widehat{m}_t)^2 + \varepsilon}$
  $\widehat{g}_t \leftarrow \widehat{m}_t + K_t^{\gamma}(g_t - \widehat{m}_t)$
  $\theta_t \leftarrow \theta_{t-1} - \alpha_t \widehat{g}_t$
end while
return θ_T

3.1. SGDF General Introduction

In this subsection, we introduce the SGDF by building on principles of optimal gradient estimation. The detailed derivation of the SGDF is provided in supplement Appendix B.1. The complete algorithm is summarized in Algorithm 1. Below, we summarize its key steps and motivation.

In stochastic gradient descent (SGD), the core challenge is to estimate a reliable gradient direction under noise. Suppose we are given a sequence of stochastic gradients $\{g_i\}_{i=1}^t$, our goal is to estimate a smoothed direction $\hat{g}_t$ that effectively combines the current observation g_t and past gradients.

Inspired by the principle of minimum mean squared error (MMSE), we begin with a naive average:

$$\hat{g}_t = \frac{1}{t}\sum_{i=1}^{t-1} g_i + \frac{1}{t} g_t = \left(1 - \frac{1}{t}\right) \bar{g}_{1:t-1} + \frac{1}{t} g_t, \quad (5)$$

where $\bar{g}_{1:t-1} = \frac{1}{t-1}\sum_{i=1}^{t-1} g_i$ denotes the averaging of the gradient under different model parameters. Then, we rewrite this as a linear interpolation:

$$\hat{g}_t = (1 - K_t)\bar{g}_{1:t-1} + K_t g_t, \quad \text{where} \quad K_t = \frac{1}{t}. \quad (6)$$

To enable efficient computation, we replace $\bar{g}_{1:t-1}$ with a bias-corrected momentum estimate $\widehat{m}_t$, giving:

$$\hat{g}_t = \widehat{m}_t + K_t(g_t - \widehat{m}_t). \quad (7)$$

This form mirrors the update structure of an Optimal Linear Filter, which recursively refines estimates by weighting

prediction and observation with gain derived from their respective uncertainties. Here, K_t acts as a gain to balance trust between the prior $\widehat{m}_t$ and the observation g_t.

To find an optimal gain, we minimize the variance of $\hat{g}_t$, assuming $\widehat{m}_t$ and g_t are independent:

$$\mathrm{Var}(\hat{g}_t) = (1 - K_t)^2 \mathrm{Var}(\widehat{m}_t) + K_t^2 \mathrm{Var}(g_t). \tag{8}$$

Solving $\frac{\mathrm{d}}{\mathrm{d}K_t}\mathrm{Var}(\hat{g}_t) = 0$ yields the optimal gain:

$$K_t^\star = \frac{\mathrm{Var}(\widehat{m}_t)}{\mathrm{Var}(\widehat{m}_t) + \mathrm{Var}(g_t)}. \tag{9}$$

This form naturally down-weights noisy gradients and shifts the estimate toward the more reliable direction.

To estimate $\mathrm{Var}(\widehat{m}_t)$ in practice, we follow [79] by computing the second-order moment s_t as the exponential moving average of $(g_t - m_t)^2$, and apply Adam's bias correction [38] to obtain $\widehat{m}_t$ and $\widehat{s}_t$. To further refine this estimate, we introduce a variance correction factor, $(1-\beta_1)(1-\beta_1^{2t})/(1+\beta_1)$ (derived in Appendix B.2 of the Supplementary Materials), which improves $\widehat{s}_t$ under independent gradients with bounded variance. Its empirical benefits and alignment with our theoretical framework are validated in the Supplementary Materials Appendix E.10. Finally, to improve responsiveness in noisy regimes, we scale K_t by $\gamma = \frac{1}{2}$, an operation formally justified in Supplementary Materials Appendix B.4 as mathematically equivalent to modulating the effective observation variance.

3.2. Fusion of Gaussian Distributions

Building on the MMSE-based principle in Sec. 3.1, we now seek a deeper statistical view of the SGDF. Specifically, we interpret the gain-controlled interpolation between the momentum estimate $\widehat{m}_t$ and the current gradient g_t as the fusion of two uncertain sources. This subsection presents both a variance-weighted linear view and a Bayesian interpretation, showing that SGDF performs optimal Gaussian fusion under the assumption of independent noise.

Linear combination view. Starting from the interpolation formula Eq. (7), we write the following:

$$\hat{g}_t = (1 - K_t)\widehat{m}_t + K_t g_t. \tag{10}$$

We assume that the stochastic gradients can be modeled as Gaussian distributions [3, 44], *i.e.*, $\widehat{m}_t \sim \mathcal{N}(\mu_m, \sigma_m^2)$ and $g_t \sim \mathcal{N}(\mu_g, \sigma_g^2)$, with independence between the two sources.

Under these assumptions, the fused mean and variance are

$$\mathbb{E}[\hat{g}_t] = (1 - K_t)\mu_m + K_t\mu_g = \frac{\sigma_g^2\mu_m + \sigma_m^2\mu_g}{\sigma_m^2 + \sigma_g^2}, \tag{11}$$

$$\mathrm{Var}(\hat{g}_t) = (1 - K_t)^2\sigma_m^2 + K_t^2\sigma_g^2 = \frac{\sigma_m^2\sigma_g^2}{\sigma_m^2 + \sigma_g^2}. \tag{12}$$

Bayesian fusion view. An equivalent result is obtained by multiplying the two Gaussian densities and normalising [21]:

$$p(\hat{g}_t) \propto \exp\left[-\frac{(\hat{g}_t - \mu_m)^2}{2\sigma_m^2} - \frac{(\hat{g}_t - \mu_g)^2}{2\sigma_g^2}\right], \tag{13}$$

which yields the posterior

$$\widehat{g}_t \sim \mathcal{N}\left(\frac{\sigma_g^2\mu_m + \sigma_m^2\mu_g}{\sigma_m^2 + \sigma_g^2}, \frac{\sigma_m^2\sigma_g^2}{\sigma_m^2 + \sigma_g^2}\right). \tag{14}$$

The fused mean $\mu_{\hat{g}_t}$ is a variance-weighted average of μ_m and μ_g, assigning greater weight to the source with lower variance to reflect higher confidence in more stable estimates. Similarly, the fused variance $\sigma_{\hat{g}_t}^2$ is smaller than both σ_m^2 and σ_g^2, indicating reduced uncertainty in the gradient estimate. This reduction results from the Optimal Linear Filter's optimality in minimizing the mean squared error. The full derivation is provided in supplement material Appendix B.3.

3.3. Convex and Non-convex Convergence Analysis

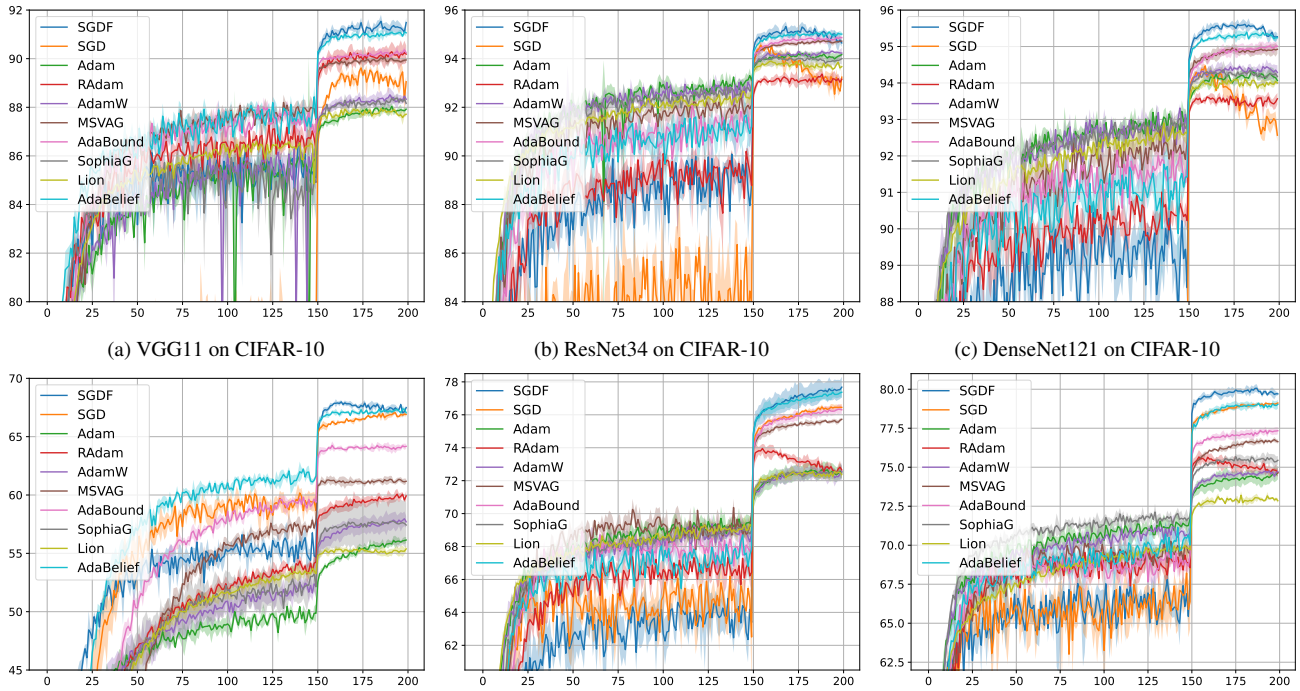
Finally, we provide the convergence property of SGDF as shown in Theorem 3.1 and Theorem 3.2. The assumptions are common and standard when analyzing the convergence of convex and non-convex functions via SGD-based methods [8, 38]. Proofs for convergence in convex and non-convex cases are provided in the supplement material Appendix C and Appendix D, respectively.

Theorem 3.1 (Convergence in Convex Optimization)**.** *Assume that the objective functions f_t are convex. Let the gradients be bounded such that $\|\nabla f_t\|_\infty \le G_\infty$, and the optimization domain be bounded with $\|\theta_m - \theta_n\|_\infty \le D_\infty$. Suppose the momentum coefficient $\beta_1, \beta_2 \in [0, 1)$ is constant, the power scaling factor follows $\gamma_t = \gamma_0/\sqrt{t}$ for some $\gamma_0 > 0$. For a learning rate $\alpha_t = \alpha/\sqrt{t}$, SGDF achieves the following cumulative regret bound for all $T \ge 1$:*

$$R(T) \le \sum_{i=1}^{d}\left(\frac{D_\infty^2}{2\alpha} + \alpha G_\infty^2\frac{1+\beta_1}{1-\beta_1}\right)\sqrt{T} + \sum_{i=1}^{d}\frac{\beta_1 G_\infty D_\infty}{1-\beta_1}\left(2 + \sum_{t=1}^{T-1}|K_{t,i} - K_{t+1,i}|\right) \tag{15}$$

In Adam-type optimizers, decaying $\beta_{1,t}$ to zero is crucial for convex analysis [38, 79]. By contrast, SGDF maintains constant coefficients (β_1, β_2) and delegates the stabilization role to the dynamic power-scaled gain $K_t^{\gamma_t}$. With $\gamma_t = \mathcal{O}(1/\sqrt{t})$ that $\sum|K_{t,i} - K_{t+1,i}| \le \mathcal{O}(1) + \mathcal{O}(\sqrt{T})$. Therefore, in the convex case, Theorem 3.1 establishes that the regret of SGDF is upper bounded by $\mathcal{O}(\sqrt{T})$.

Theorem 3.2. *(Convergence for Non-convex Stochastic Optimization) Assume Assumptions 1–4 hold and the step size is $\alpha_t = \alpha/\sqrt{t}$. For all $T \ge 1$, SGDF satisfies:*



(a) VGG11 on CIFAR-10 (b) ResNet34 on CIFAR-10 (c) DenseNet121 on CIFAR-10

(d) VGG11 on CIFAR-100 (e) ResNet34 on CIFAR-100 (f) DenseNet121 on CIFAR-100

Figure 1. Test accuracy ($[\mu \pm \sigma]$) on CIFAR.

1. ***Bounded Variables:*** $\|\theta - \theta^*\|_2 \leq D$ *(or* $\|\theta_i - \theta_i^*\|_2 \leq D_i$ *for each dimension* i*) for all* θ, θ^*.
2. ***Unbiased Noise:*** *The noise* $\zeta_t = g_t - \nabla f(\theta_t)$ *satisfies* $\mathbb{E}[\zeta_t] = 0$, $\mathbb{E}[\|\zeta_t\|_2^2] \leq \sigma^2$, *and* ζ_{t_1}, ζ_{t_2} *are independent for* $t_1 \neq t_2$.
3. ***Bounded Gradients:*** *Both the true and noisy gradients are uniformly bounded, i.e.,* $\|\nabla f(\theta_t)\|_2 \leq G$ *and* $\|g_t\|_2 \leq G$ *for all* $t \geq 1$.
4. ***Function Properties:*** f *is differentiable, lower bounded, and* L*-smooth, i.e.,* $\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2$ *for all* x, y.

The convergence guarantee is given by:

$$\mathbb{E}(T) \leq \frac{C_7 \alpha^2 (\log T + 1) + C_8}{\alpha \sqrt{T}}, \tag{16}$$

where $\mathbb{E}(T) = \min_{t=1,\ldots,T} \mathbb{E}_t[\|\nabla f(\theta_t)\|_2^2]$ *is the minimum expected squared gradient norm,* α *is the initial learning rate, and* C_7, C_8 *are constants independent of* T *(*C_7 *also independent of* d*). The expectation is taken over all randomness in* $\{g_t\}$.

Theorem 3.2 shows that the convergence rate of SGDF in the non-convex setting is $\mathcal{O}(\log T/\sqrt{T})$, which matches the rates established for Adam-type optimizers [8, 57]. In our analysis, the terms involving the estimated gain K_t were upper bounded by their maximal possible values to simplify the derivation of the final bound. We adopted the general L-smoothness assumption to obtain this rate. Furthermore, if a fixed decay schedule $\alpha/\sqrt{T}$ is used, where T represents the total number of iterations, instead of $\alpha/\sqrt{t}$ for infinite iterations, the convergence rate improves to $\mathcal{O}(1/\sqrt{T})$ [65].

4. Experiments

4.1. Empirical Evaluation

In this study, we focus on the following tasks: **Image Classification.** We employed the VGG [61], ResNet [29], and DenseNet [31] models for image classification tasks on the CIFAR dataset [39]. The major difference between these three network architectures is the residual connectivity, which we will discuss in Sec. 4.2. We evaluated and compared the performance of SGDF with other optimizers such as SGD, Adam, RAdam [44], AdamW [45], MSVAG [1], Adabound [47], Sophia [43], Lion [9], and AdaBelief [79], all of which were implemented based on the official PyTorch. Additionally, we further tested the performance of SGDF on the ImageNet dataset [14] using the ResNet model. **Object Detection.** Object detection was performed on the PASCAL VOC dataset [20] using Faster-RCNN [58] integrated with FPN. **Post-training in ViT.** We test the performance of transformer architecture networks by post-training ViT [15] on six benchmark dataset. **More experimental results are in the Supplemental Materials Appendix E.**

Table 2. Top-1, 5 accuracy (%) of ResNet18 on ImageNet. * † ‡ is reported in [7, 44, 79].

Method	SGDF	SGD	AdaBelief	PAdam	AdaBound	Yogi	MSVAG	Adam	RAdam	AdamW
Top-1	**70.51**$_{\pm 0.05}$	70.23†	70.08*	70.07†	68.13†	68.23†	65.99*	63.79† (66.54‡)	67.62‡	67.93†
Top-5	**89.69**$_{\pm 0.16}$	89.40†	-	89.47†	88.55†	88.59†	-	85.61†	-	88.47†

Table 3. Comparison of top-1 accuracy (%) across different model variants and optimizers on the ImageNet classification.

Model	VGG11	VGG13	ResNet34	ResNet50	DenseNet121	DenseNet161
SGD[1]	70.37	71.58	73.31	76.13	74.43	77.13
SGDF	**71.34**$_{\pm 0.21}$	**72.74**$_{\pm 0.07}$	**74.07**$_{\pm 0.21}$	**76.72**$_{\pm 0.09}$	**75.75**$_{\pm 0.09}$	**78.34**$_{\pm 0.08}$

Hyperparameter tuning. Following [79], we delved deep into the optimal hyperparameter settings for our experiments.

- *SGDF:* We followed to Adam's original parameter values except learning rate: $\alpha = 0.5$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$. The learning rate was searched same as SGD research set.
- *SGD:* We set the momentum 0.9, which is the default for networks like ResNet and DenseNet. The learning rate was searched in the set $\{10.0, 1.0, 0.5, 0.1, 0.01, 0.001\}$.
- *Adam, RAdam, MSVAG, AdaBound, AdaBelief:* Exploring the hyperparameter space, we tested β_1 values in $\{0.5, 0.6, 0.7, 0.8, 0.9\}$, examined α as in SGD, while keeping other parameters to their literary defaults.
- *AdamW, SophiaG, Lion:* Following Adam's parameter search pattern, we fixed weight decay at 5×10^{-4}; yet for AdamW, whose optimal decay often exceeds norms [45], we ranged weight decay over $\{10^{-4}, 5 \times 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$.
- *SophiaG, Lion:* We searched for the learning rate among $\{10^{-3}, 10^{-4}, 10^{-5}\}$ and adjusted Lion's learning rate [43]. Following [9, 43], we set β_1=0.965, 0.9 and default parameters β_2=0.99.

CIFAR-10/100 Experiments. We trained on the CIFAR-10 and CIFAR-100 datasets using the VGG, ResNet and DenseNet models and access the performance of the SGDF optimizer. In these experiments, we employ basic data augmentation techniques such as random horizontal flip and random cropping. The results represent the mean and standard deviation of 3 runs by fixing random seeds $\{0, 1, 2\}$.

As Fig. 1 shows, it is evident that the SGDF optimizer exhibited convergence speeds comparable to adaptive optimization algorithms. Additionally, SGDF's final test set accuracy was better than others. These consistent results across multiple architectures indicate that SGDF can effectively adapt to networks of varying depths and complexities. We summarize the numerical results for the mean best test accuracies, standard deviations, and parameter details in Appendix E.1 of the Supplemental Materials.

ImageNet Experiments. We applied standard data augmentation techniques, including random cropping and random horizontal flipping [79], using random seeds $\{0, 1, 2\}$. To eliminate the effect of stepwise learning rate schedules, we adopted cosine learning rate decay as suggested by [9, 77]. Following [7, 79], ResNet18 was trained for 100 epochs to compare with popular optimizers using their best-reported results [7, 44, 79].

Following the GAM [77] setup, we further benchmarked SGDF against SGD on VGG11/13 (with BN), ResNet34/50, and DenseNet121/161 for 90 epochs. As shown in Tab. 2 and Tab. 3, SGDF consistently outperforms SGD, demonstrating robust scalability, smooth convergence, and strong generalization across varying network capacities. Detailed training/test curves and hyperparameters are provided in Supplemental Materials Appendix E.2.

Table 4. The mAP on PASCAL VOC. * † is reported in [73, 79].

Method	SGDF	AdaBelief	EAdam	SGD	Adam	AdamW	RAdam
mAP	**83.81**	81.02*	80.62†	80.43	78.67	78.48	75.21

Object Detection. We conducted object detection experiments on the PASCAL VOC dataset [20]. The model used in these experiments was pre-trained on the COCO dataset [42], obtained from the official website. We trained this model on the VOC2007 and VOC2012 train-val dataset (17K) and evaluated it on the VOC2007 test dataset (5K). The utilized model was Faster-RCNN [58] with FPN, and the backbone was ResNet50 [29]. Results are summarized in Tab. 4. To facilitate result reproduction, we provide the parameter table for this subpart in the supplementary material (Appendix E.3). As expected, SGDF outperforms other methods in detection accuracy and stability. These results demonstrate the efficiency and robustness of our method in complex detection tasks, highlighting its consistent optimization behavior across vision architectures.

[1]Results from PyTorch official pre-trained models.

Table 5. Post-training in ViT. We report Top-1 accuracy (%).

Model	Method	CIFAR-10	CIFAR-100	Oxford-IIIT-Pets	Oxford Flowers-102	Food101	ImageNet
ViT-B/32	SGD	$98.71_{\pm 0.03}$	$90.62_{\pm 0.07}$	$89.71_{\pm 0.32}$	$96.79_{\pm 0.29}$	$88.56_{\pm 0.05}$	$81.42_{\pm 0.04}$
	SGDF	$\mathbf{98.74}_{\pm 0.10}$	$\mathbf{91.44}_{\pm 0.13}$	$\mathbf{92.68}_{\pm 0.04}$	$\mathbf{97.17}_{\pm 0.47}$	$\mathbf{89.35}_{\pm 0.09}$	$\mathbf{81.52}_{\pm 0.02}$
ViT-L/32	SGD	$98.73_{\pm 0.05}$	$91.30_{\pm 0.17}$	$85.21_{\pm 0.39}$	$96.52_{\pm 0.15}$	$89.13_{\pm 0.20}$	$81.28_{\pm 0.04}$
	SGDF	$\mathbf{98.83}_{\pm 0.04}$	$\mathbf{92.20}_{\pm 0.14}$	$\mathbf{91.96}_{\pm 0.18}$	$\mathbf{96.79}_{\pm 0.12}$	$\mathbf{90.04}_{\pm 0.08}$	$\mathbf{81.38}_{\pm 0.01}$

Post-training in ViT. To evaluate SGDF on a critical benchmark where SGD with momentum traditionally excels, we followed the standard ViT transfer learning protocol [15]. We tested ImageNet-21K pre-trained ViT-B/32 and ViT-L/32 across six datasets: CIFAR-10/100, Oxford-IIIT-Pets [54], Oxford Flowers-102 [52], Food101 [4], and ImageNet-1K. Strictly replicating the original setup, we froze the backbone, replaced the MLP head, and trained for 10 epochs (seeds $\{0, 1, 2\}$). Input images were upsampled to 384×384 with 2D-interpolated position encodings. We utilized a batch size of 512, cosine learning rate decay, zero weight decay, and gradient clipping (norm 1). As shown in Tab. 5, SGDF effectively validates its performance against established SGD baselines. Detailed hyperparameters are provided in Supplementary Materials Appendix E.6.

4.2. Extensibility of Filter-Estimated Gradients

Beyond vanilla SGD, our optimal linear filter serves as a plug-and-play module that enhances first-moment gradient estimates across diverse optimization frameworks. To demonstrate its broad compatibility, we integrate it with Adam, Sign-based optimizers [3], and the Muon [33].

Table 6. Accuracy comparison between Adam and Filter-Adam.

Model	VGG11	ResNet34	DenseNet121
Filter-Adam	**62.64**	**73.98**	**74.89**
Vanilla-Adam	56.73	72.34	**74.89**

Integration with Adam. We first evaluated our filter's integration with Adam by substituting the first-moment gradient estimates in the vanilla optimizer with our filtered counterparts. As shown in Tab. 6 (with detailed test curves in Supplementary Materials Appendix E.11), experiments on the CIFAR-100 dataset reveal distinct behaviors across network architectures. For networks without Batch Normalization (BN) like VGG, the filter significantly improves performance by providing more accurate gradient estimates and reducing noise-induced errors. For architectures like ResNet and DenseNet that inherently promote stable gradient updates through BN and residual/dense connections, the filter still maintains highly competitive performance, albeit with less pronounced marginal gains. This structural dynamic effectively explains the variance in improvements across architectures.

Extension to Sign-based Optimizers. To further evaluate the robustness of our gradient estimation in different update paradigms, we tested a sign-based variant. Specifically, we updated the parameters using the sign of our filtered gradient ($\text{sign}(\widehat{g}_t)$) [3]. We evaluated this *Sign* SGDF on diffusion models, specifically DiT/SiT-Base (130M parameters) [49, 55] in ImageNet. For a fair comparison, we aligned all hyperparameters with Adam and used an unscaled K_t. As shown in Fig. 2, Sign SGDF converges significantly faster and achieves a better FID score than the Adam baseline, proving that our filter effectively captures the true gradient direction even when the magnitude is discarded.

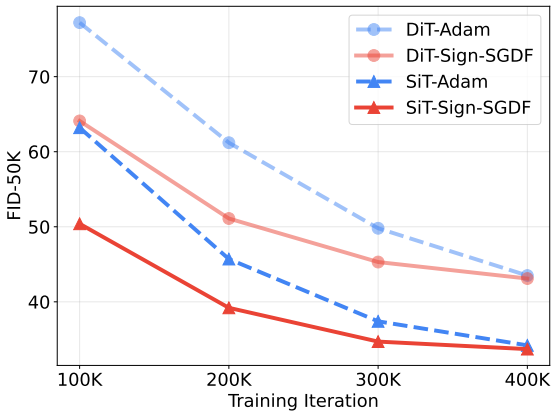


Figure 2. Convergence comparison between Sign SGDF and Adam.

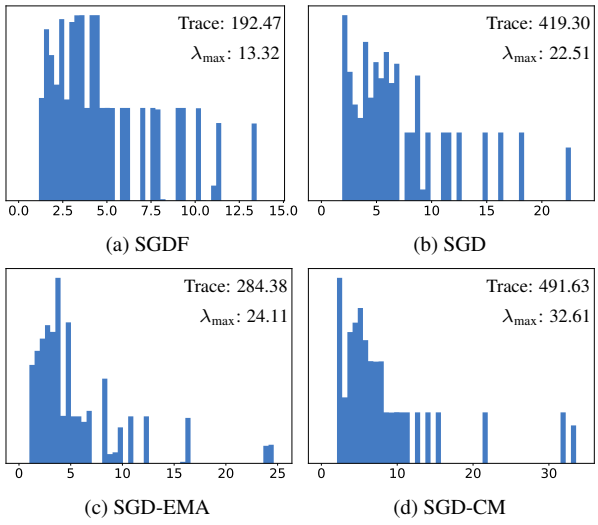
Compatibility with Muon. Finally, we explored the compatibility of our filter with Muon, a recently proposed optimizer that relies on orthogonal momentum. We conducted a preliminary experiment on DiT-B/4 by replacing Muon's standard momentum formulation with our filter-estimated gradient. As presented in Tab. 7, this integration yields superior results compared to the standard Adam baseline across multiple generation metrics at 400K training iterations. This underscores the potential of our optimal linear filter to serve as a fundamental gradient refinement step for state-of-the-art training recipes.

Table 7. Compatibility of filter-estimated gradient with Muon.

Method	FID↓	sFID↓	IS↑	Pre.↑	Rec.↑
Adam	68.32	13.63	20.51	0.36	0.53
Muon + SGDF	**64.24**	**12.43**	**22.26**	**0.37**	**0.59**

4.3. Top Eigenvalues of Hessian and Hessian Trace

The success of optimization algorithms in deep learning depends on both minimizing training loss and the quality of the solutions they find. So we numerically verified the hessian matrix properties between the different methods. We computed the Hessian spectrum of ResNet-18 trained on the CIFAR-100 dataset for 200 epochs. These experiments ensure that all methods achieve similar results on the training set. We employed power iteration [70] to compute the top eigenvalues of Hessian and Hutchinson's method [71] to compute the Hessian trace. Histograms illustrating the distribution of the top 50 Hessian eigenvalues for each optimization method are presented in Fig. 3. SGDF brings lower eigenvalue and trace of the hessian matrix, which explains the fact that SGDF demonstrates better performance than SGD as the categorization category increases.



(a) SGDF (b) SGD

(c) SGD-EMA (d) SGD-CM

Figure 3. Histogram of Top 50 Hessian Eigenvalues. Lower values indicate better performance on the test dataset.

5. Related Works

Early optimization efforts focused on variance reduction [1, 13, 32, 60] and adaptive learning rates [16, 18, 74]. However, standard adaptive optimizers often converge to sharp minima with weak generalization in high-dimensional non-convex landscapes [11, 28, 46, 66, 67]. This motivated geometry-aware regularization, such as Sharpness-Aware Minimization (SAM) and its variants [22, 77, 80], alongside momentum tuning strategies like Adaptive Inertia [68], to explicitly penalize sharpness and escape sharp basins.

Beyond loss geometry, recent studies highlight the critical role of dynamically balancing gradient bias and variance for stable learning [24, 27]. To achieve this, parallel works manipulate updates directly: Quasi-Hyperbolic Momentum (QHM) [48] introduces a static interpolation between current gradients and momentum, while Grokfast [40] heuristically amplifies low-frequency gradient signals. Alternatively, methods based on second-order approximations or Kalman filtering [12, 25, 34, 43, 53, 64, 72] refine updates using local curvature or Fisher information. While principled, these techniques significantly inflate computational overhead and parameter complexity, limiting their practical scalability.

In contrast, we formulate momentum fundamentally as an online filtering process to rigorously resolve the bias-variance dilemma without the heavy overhead of curvature estimation. Unlike QHM's static interpolation or Grokfast's heuristic frequency scaling, SGDF fuses past and current gradients via an optimal, time-varying gain that adaptively minimizes mean-squared estimation error. Consequently, SGDF achieves noise-aware, stable updates through a lightweight statistical design, ensuring principled gradient refinement at no additional computational cost.

6. Discussion and Future Work

Our SDE framework readily accommodates existing continuous-time analyses [41, 62, 68]. A compelling theoretical direction is coupling first-order gradient statistics with zeroth-order loss moments via Itô's lemma, establishing unified bounds that link local stability to global generalization in non-convex settings. Practically, while SGDF currently requires a memory footprint comparable to standard Adam, integrating block-wise or reduced-state approximations, which are inspired by recent memory-efficient designs like Adam-mini [78], presents a highly promising avenue to minimize overhead. Together, these theoretical and empirical extensions will rigorously refine adaptive optimization for large-scale learning systems.

7. Conclusion

In this work, we introduced SGDF, a novel optimizer rooted in statistical signal processing. By dynamically minimizing the mean-squared estimation error, SGDF balances gradient bias and variance, overcoming the suboptimal trade-offs of traditional momentum to yield highly accurate first-moment estimates. Extensive experiments across diverse architectures confirm that this principled approach achieves a superior tradeoff between convergence speed and generalization compared to current state-of-the-art optimizers.

Acknowledgement

The authors gratefully acknowledge the financial support provided by the Scientific Research Project of Liaoning Provincial Department of Education under Grant LJ212510149013. Rui Yu received no funding in support of this work. Zhipeng Yao thanks to Aram Davtyan and Prof. Dr. Paolo Favaro in the Computer Vision Group at the University of Bern for discussing and improving the paper.

References

[1] Lukas Balles and Philipp Hennig. Dissecting adam: The sign, magnitude and variance of stochastic gradients. In *International Conference on Machine Learning*, pages 404–413. PMLR, 2018. 5, 8

[2] Yoshua Bengio and Yann Lecun. Scaling learning algorithms towards ai. 2007. 1

[3] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pages 560–569. PMLR, 2018. 4, 7

[4] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part VI 13*, pages 446–461. Springer, 2014. 7

[5] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM review*, 60(2):223–311, 2018. 1

[6] Nisha Chandramoorthy, Andreas Loukas, Khashayar Gatmiry, and Stefanie Jegelka. On the generalization of learning algorithms that do not converge. *Advances in Neural Information Processing Systems*, 35:34241–34257, 2022. 1

[7] Jinghui Chen, Dongruo Zhou, Yiqi Tang, Ziyan Yang, Yuan Cao, and Quanquan Gu. Closing the generalization gap of adaptive gradient methods in training deep neural networks. *arXiv preprint arXiv:1806.06763*, 2018. 1, 6

[8] Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. In *International Conference on Learning Representations*, 2019. 4, 5

[9] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Yao Liu, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, et al. Symbolic discovery of optimization algorithms. *arXiv preprint arXiv:2302.06675*, 2023. 5, 6

[10] Ashok Cutkosky and Harsh Mehta. Momentum improves normalized sgd. In *International conference on machine learning*, pages 2260–2268. PMLR, 2020. 2

[11] Yann Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *MIT Press*, 2014. 8

[12] Aram Davtyan, Sepehr Sameni, Llukman Cerkezi, Givi Meishvili, Adam Bielski, and Paolo Favaro. Koala: A kalman optimization algorithm with loss adaptivity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6471–6479, 2022. 8

[13] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014. 8

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 5

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 5, 7

[16] Timothy Dozat. Incorporating nesterov momentum into adam. *International Conference on Learning Representations Workshop, 2016*, 2016. 1, 8

[17] Simon Du and Jason Lee. On the power of over-parametrization in neural networks with quadratic activation. In *International conference on machine learning*, pages 1329–1338. PMLR, 2018. 1

[18] Duchi, John, Hazan, Elad, Singer, and Yoram. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 2011. 8

[19] John Duchi and Hongseok Namkoong. Variance-based regularization with convex objectives. *Journal of Machine Learning Research*, 20(68):1–55, 2019. 1

[20] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 5, 6

[21] Ramsey Faragher. Understanding the basis of the kalman filter via a simple and intuitive derivation [lecture notes]. *IEEE Signal processing magazine*, 29(5):128–132, 2012. 4

[22] Pierre Foret et al. Sharpness-aware minimization for efficiently improving generalization. In *ICLR*, 2021. spotlight. 8

[23] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 2014. 1

[24] Arna Ghosh, Yuhan Helena Liu, Guillaume Lajoie, Konrad Kording, and Blake Aaron Richards. How gradient estimator variance and bias impact learning in neural networks. In *The Eleventh International Conference on Learning Representations*, 2022. 8

[25] Damien Martins Gomes, Yanlei Zhang, Eugene Belilovsky, Guy Wolf, and Mahdi S Hosseini. Adafisher: Adaptive second order optimization via fisher information. *arXiv preprint arXiv:2405.16397*, 2024. 8

[26] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016. 1

[27] Mingming Ha, Xuewen Tao, Wenfang Lin, Qiongxu Ma, Wujiang Xu, and Linxun Chen. Fine-grained dynamic framework for bias-variance joint optimization on data missing not at random. *Advances in Neural Information Processing Systems*, 37:104010–104034, 2024. 8

[28] Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. *Mathematics*, 2015. 8

[29] 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. 5, 6

[30] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012. 1

[31] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 5

[32] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26, 2013. 8

[33] Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. 7

[34] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 1960. 3, 8

[35] Steven M Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, Inc., 1993. 3

[36] Nitish Shirish Keskar and Richard Socher. Improving generalization performance by switching from adam to sgd. *arXiv preprint arXiv:1712.07628*, 2017. 1

[37] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2022. 1

[38] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1, 2, 4

[39] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[40] Jaerin Lee, Bong Gyun Kang, Kihoon Kim, and Kyoung Mu Lee. Grokfast: Accelerated grokking by amplifying slow gradients. *arXiv preprint arXiv:2405.20233*, 2024. 8

[41] Qianxiao Li, Cheng Tai, and E Weinan. Stochastic modified equations and dynamics of stochastic gradient algorithms i: Mathematical foundations. *Journal of Machine Learning Research*, 20(40):1–47, 2019. 8

[42] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. *European Conference on Computer Vision (ECCV)*, 2014. 6

[43] Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. *arXiv preprint arXiv:2305.14342*, 2023. 5, 6, 8

[44] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations*, 2020. 1, 4, 5, 6

[45] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5, 6

[46] Aurelien Lucchi, Frank Proske, Antonio Orvieto, Francis Bach, and Hans Kersting. On the theoretical properties of noise correlation in stochastic optimization. *Advances in Neural Information Processing Systems*, 35:14261–14273, 2022. 8

[47] Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. *arXiv preprint arXiv:1902.09843*, 2019. 1, 5

[48] Jerry Ma and Denis Yarats. Quasi-hyperbolic momentum and adam for deep learning. *arXiv preprint arXiv:1810.06801*, 2018. 8

[49] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pages 23–40. Springer, 2024. 7

[50] Robbins Sutton Monro. a stochastic approximation method. *Annals of Mathematical Statistics*, 22(3):400–407, 1951. 1

[51] Han Nguyen, Hai Pham, Sashank J Reddi, and Barnabás Póczos. On the algorithmic stability and generalization of adaptive optimization methods. *arXiv preprint arXiv:2211.03970*, 2022. 1

[52] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 7

[53] Yann Ollivier. The extended kalman filter is a natural gradient descent in trajectory space. *arXiv: Optimization and Control*, 2019. 8

[54] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 7

[55] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 7

[56] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964. 1, 2

[57] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018. 5

[58] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Neural Information Processing Systems (NIPS)*, 2015. 5, 6

[59] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 1

[60] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1):83–112, 2017. 8

[61] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Computer Science*, 2014. 5

[62] Mandt Stephan, Matthew D Hoffman, David M Blei, et al. Stochastic gradient descent as approximate bayesian inference. *Journal of Machine Learning Research*, 18(134):1–35, 2017. 2, 8

[63] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013. 1, 2

[64] James Vuckovic. Kalman gradient descent: Adaptive variance reduction in stochastic optimization. *ArXiv*, 2018. 8

[65] Bohan Wang, Jingwen Fu, Huishuai Zhang, Nanning Zheng, and Wei Chen. Closing the gap between the upper bound and lower bound of adam's iteration complexity. *Advances in Neural Information Processing Systems*, 36:39006–39032, 2023. 5

[66] Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. *Advances in neural information processing systems*, 30, 2017. 8

[67] Zeke Xie, Qian Yuan Tang, Yunfeng Cai, Mingming Sun, and Ping Li. On the power-law spectrum in deep learning: A bridge to protein science. *arXiv preprint arXiv:2201.13011*, 2, 2022. 8

[68] Zeke Xie, Xinrui Wang, Huishuai Zhang, Issei Sato, and Masashi Sugiyama. Adaptive inertia: Disentangling the effects of adaptive learning rate and momentum. In *International conference on machine learning*, pages 24430–24459. PMLR, 2022. 8

[69] Ning Yang, Chao Tang, and Yuhai Tu. Stochastic gradient descent introduces an effective landscape-dependent regularization favoring flat solutions. *Physical Review Letters*, 130 (23):237101, 2023. 1

[70] Zhewei Yao, Amir Gholami, Qi Lei, Kurt Keutzer, and Michael W Mahoney. Hessian-based analysis of large batch training and robustness to adversaries. *Advances in Neural Information Processing Systems*, 31, 2018. 8

[71] Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W. Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *International Conference on Big Data*, 2020. 8

[72] Zhewei Yao, Amir Gholami, Sheng Shen, Kurt Keutzer, and Michael W Mahoney. Adahessian: An adaptive second order optimizer for machine learning. *arXiv preprint arXiv:2006.00719*, 2020. 8

[73] Wei Yuan and Kai-Xin Gao. Eadam optimizer: How ϵ impact adam. *arXiv preprint arXiv:2011.02150*, 2020. 6

[74] Matthew D. Zeiler. Adadelta: An adaptive learning rate method. *arXiv e-prints*, 2012. 8

[75] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021. 1

[76] Jian Zhang and Ioannis Mitliagkas. Yellowfin and the art of momentum tuning. *arXiv preprint arXiv:1706.03471*, 2017. 1

[77] Xingxuan Zhang, Renzhe Xu, Han Yu, Hao Zou, and Peng Cui. Gradient norm aware minimization seeks first-order flatness and improves generalization. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 6, 8

[78] Yushun Zhang, Congliang Chen, Ziniu Li, Tian Ding, Chenwei Wu, Diederik P Kingma, Yinyu Ye, Zhi-Quan Luo, and Ruoyu Sun. Adam-mini: Use fewer learning rates to gain more. *arXiv preprint arXiv:2406.16793*, 2024. 8

[79] Juntang Zhuang, Tommy Tang, Yifan Ding, Sekhar C Tatikonda, Nicha Dvornek, Xenophon Papademetris, and James Duncan. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. *Advances in neural information processing systems*, 33:18795–18806, 2020. 1, 4, 5, 6

[80] Juntang Zhuang, Boqing Gong, Liangzhe Yuan, Yin Cui, Hartwig Adam, Nicha Dvornek, Sekhar Tatikonda, James Duncan, and Ting Liu. Surrogate gap minimization improves sharpness-aware training. *arXiv preprint arXiv:2203.08065*, 2022. 8