LOVA: COMBATING LOCAL VARIANCE DOMINANCE FOR EFFICIENT NEURAL NETWORK TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Adam is one of the most dominant optimizers for training neural networks. Nevertheless, it has been challenged by matrix-based optimizers recently. In this work, we identify that the signal is often dominated by the random noise in Adam, an issue that becomes increasingly severe in large-batch training. The consequence is catastrophic—Adam overdampens the updates and heavily impedes the training efficiency. To address this issue, we propose *Lova (**Lo**cal **V**ariance **A**daptation)*, a memory-efficient optimizer that preserves the signal drift while retaining the adaptivity of coordinate-wise learning rate. It should be highlighted that Lova tracks only the first-moment buffer, and the memory footprint is as low as stochastic gradient descent (SGD) with momentum. While Adam is not guaranteed to converge for smooth nonconvex functions, theoretical analysis suggests that Lova, under standard assumptions, achieves a convergence rate of $\mathcal{O}(T^{-1/4}\ln T)$. Importantly, this rate holds for an arbitrary constant $\beta_2 \in (0,1)$ and eliminates the polynomial dependence on the reciprocal of a small constant for numerical stability, $1/\epsilon$. Extensive experiments on various architectures, datasets, and tasks demonstrate the superiority of Lova over AdamW and other popular optimizers.

1 INTRODUCTION

Adaptive optimization algorithms (Wen et al., 2026; Zhang et al., 2026; Li et al., 2026) play a fundamental role in deep learning, among which Adam (Kingma & Ba, 2015) remains the golden standard in practice. Different from stochastic gradient descent (SGD), Adam rescales the coordinate-wise learning rates with historical gradient statistics to navigate complex, non-convex loss landscapes. Mathematically, for a learning rate η_t at timestep t, Adam is governed by the following equations,

$$\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1-\beta_1)\mathbf{g}_t, \quad \mathbf{v}_t^2 = \beta_2 \mathbf{v}_{t-1}^2 + (1-\beta_2)\mathbf{g}_t^2, \quad \boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \eta_t \frac{\mathbf{m}_t}{\sqrt{\mathbf{v}_t^2 + \epsilon}}. \quad (1)$$

Here, $\boldsymbol{\theta}_t \in \mathbb{R}^d$ stands for the learnable parameters, $\mathbf{g}_t$ the mini-batch stochastic gradient, $\mathbf{m}_t, \mathbf{v}_t^2$ the exponential moving average (EMA) of the first and second moment of $\mathbf{g}_t$, and β_1, β_2 are the smoothing factors. Note that for convenience here we omit the bias correction terms, and a small constant $\epsilon > 0$ is introduced to prevent division by zero.

Previous studies (Kingma & Ba, 2015; Balles & Hennig, 2018) have shown that after a sufficiently long training, when the updates are stationary, $\mathbf{m}_t$ estimates the expected gradient $\mathbb{E}[\mathbf{g}_t]$ and $\mathbf{v}_t^2$ estimates the expected second moment of $\mathbb{E}[\mathbf{g}_t^2]$. According to the expectation-variance decomposition,

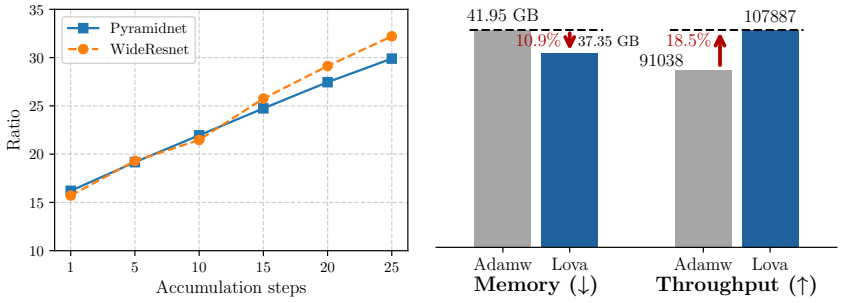
$$\mathbb{E}[\mathbf{g}_t^2] = \mathbb{E}[\mathbf{g}_t]^2 + \mathbb{E}[(\mathbf{g}_t - \mathbb{E}[\mathbf{g}_t])^2],$$

it naturally yields the following approximation,

$$\mathbf{v}_t^2 \approx \mathbf{m}_t^2 + (\mathbf{g}_t - \mathbf{m}_t)^2,$$

where $(\mathbf{g}_t - \mathbf{m}_t)^2$ can be treated as the local variance, estimating the instantaneous stochastic noise at timestep t. As shown in Figure 1(a), however, the local variance $(\mathbf{g}_t - \mathbf{m}_t)^2$ often exceeds the signal $\mathbf{m}_t^2$, a phenomenon which we refer to as *local variance dominance*. It causes Adam to overdampen updates even when the gradient direction is consistent, an issue that becomes particularly severe in large-batch training. Furthermore, Adam requires tracking the states of $\mathbf{m}_t$ and $\mathbf{v}_t^2$, which poses a great challenge for large language model (LLM) training. To reduce memory footprint, Adafactor

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



(a) (b)

Figure 1: (a) Evolution of the ratio between $(\mathbf{g}_t - \mathbf{m}_t)^2$ and $\mathbf{m}_t^2$ against the accumulation steps that effectively increase the mini-batch size. Both networks, including PyramidNet (Han et al., 2017) and WideResNet (Zagoruyko & Komodakis, 2016), are trained on CIFAR-100 using the AdamW (Loshchilov & Hutter, 2019) optimizer with a cosine learning rate decaying scheduler. The value of the ratio is averaged across all steps of the training process. (b) Memory and throughput (# tokens per second) for pre-training LLaMA-124M with the same mini-batch size on 8×NVIDIA-PRO5000-48GB GPUs. Lova takes less memory (the reduced amount is equal to the model size) and reaches higher throughput. Note that throughout this paper, all experiments are conducted using AdamW (Loshchilov & Hutter, 2019) as it generally performs better than the vanilla Adam.

(Shazeer & Stern, 2018) proposes to factorize $\mathbf{v}_t^2$ into two rank-1 factors, while Adam-mini (Zhang et al., 2025b) argues that more than 99% of the elements of $\mathbf{v}_t^2$ can be harmlessly removed if the parameters are carefully partitioned into blocks. But these memory-efficient variants often suffer from performance degradation due to the erroneous update (Luo et al., 2023; Zhang et al., 2025a).

In this paper, we present *Lova (**Lo**cal **V**ariance **A**daptation)*, an adaptive optimization framework that unifies $\mathbf{m}_t$ and $\mathbf{g}_t - \mathbf{m}_t$ into a single second-order statistic,

$$\mathbf{v}_t^2 = \beta_2 \mathbf{m}_t^2 + (1 - \beta_2)(\mathbf{g}_t - \mathbf{m}_t)^2,$$

where β_2 is the smoothing factor. By interpolating between two extremes, Lova provides a principled way to counteract the local variance dominance: when the noise dominates, the $(\mathbf{g}_t - \mathbf{m}_t)^2$ term suppresses the update, mirroring AdaBelief (Zhuang et al., 2020); when the signal dominates, the $\mathbf{m}_t^2$ term preserves the update, recovering SIGNUM (Bernstein et al., 2018). In brief, our contributions are summarized as follows:

- We introduce Lova, an adaptive optimizer that interpolates between momentum-based sign updates and variance-aware scaling. It tracks only the first-moment buffer, and its memory footprint is as low as SGD with momentum acceleration (Polyak, 1964), which, as a result, allows larger batch sizes and increases throughput (see Figure 1(b)).
- We show that Lova achieves an $1/\epsilon$-independent convergence bound for smooth nonconvex stochastic optimization, eliminating the polynomial dependence on $1/\epsilon$ that appears in most Adam-type analyses (Zaheer et al., 2018; Zou et al., 2019; Défossez et al., 2020; Guo et al., 2021). Since ϵ is often set to 1.0×10^{-8} or smaller for numerical stability, this dependence can dominate the theoretical bound in practice. The key ingredient is the structural property $\mathbf{v}_t^2 \geq \beta_2 \mathbf{m}_t^2$, which bounds the effective update without resorting to ϵ. More importantly, the final rate $\mathcal{O}(\ln T / T^{1/4})$ is attained under standard assumptions and holds for arbitrary constant $\beta_2 \in (0, 1)$.
- Through extensive experiments on image classification, LLM pretraining, reinforcement learning, and image generation, we demonstrate that Lova achieves superior practical performance to AdamW and other popular optimizers such as Lion (Chen et al., 2023) and MARS (Yuan et al., 2025).

The remainder is organized as follows. Section 2 reviews the related works, and Section 3 elaborates the details of the proposed algorithm. Before concluding in Section 6, the convergence analysis of Lova is presented in Section 4, and the experimental results are provided in Section 5.

2 Related Works

In this section, we provide a brief review of the previous studies that are most relevant to our work from two aspects.

Connection to adaptive gradient methods. Adaptive gradient methods adjust learning rates based on second-order statistics of past gradients. Apart from the early optimizers such as AdaGrad (Duchi et al., 2011) and RMSProp (Tieleman & Hinton, 2017), subsequent variants of Adam primarily attempt to address various failure modes or refine the variance estimation. For example, the most famous AdamW (Loshchilov & Hutter, 2019) decouples the weight decay from the traditional ℓ_2 regularization and significantly improves the generalization performance. To address the convergence issue of Adam, AMSGrad (Reddi et al., 2018) is proposed to ensure the monotonicity of the effective learning rates. It is further analyzed under the framework of PAdam (Chen et al., 2018), which imposes a partial adaptive parameter to control the degree of adaptiveness. To improve the robustness against noisy gradients, AdaBelief (Zhuang et al., 2020) replaces $\mathbf{g}_t^2$ with $(\mathbf{g}_t - \mathbf{m}_t)^2$ to adapt the learning rates according to the "belief" in the current gradient observation. To understand the underlying mechanisms, Balles & Hennig (2018) interpret Adam from the sign and magnitude aspects and propose SVAG by transferring the variance adaptation directly to SGD. Several other works, including Adam$^+$ (Liu et al., 2020), SuperAdam (Huang et al., 2021) and MARS (Yuan et al., 2025), also explore the application of variance reduction techniques. Different from these optimizers, the second-order statistic of Lova is memoryless and thus significantly reduces the memory footprint.

Connection to sign-based optimizers. Sign-based algorithms, such as SignSGD and SIGNUM (Bernstein et al., 2018), restrict update magnitudes by applying the element-wise sign operator to gradients or momentum vectors, which were initially designed to reduce communication overhead. Later on, Lion (Chen et al., 2023) demonstrated that tracking only the sign of momentum yields strong generalization performance across large-scale vision and language tasks. More recently, C-Lion (Liang et al., 2026) proposes updating the parameters only when the update direction and the current gradients are aligned. Theoretically, several studies (Chen et al., 2024; Sfyraki & Wang, 2026) also connect Lion to constrained optimization and ℓ_∞ geometry interpretations. However, sign-based optimizers are highly sensitive to large learning rates and are not guaranteed to converge. These issues are addressed by Lova since the effective update is now also modulated by the local variance.

3 Methodology

In training a neural network, we often consider the minimization of the objective function $f : \mathbb{R}^d \mapsto \mathbb{R}$ with respect to the learnable parameters $\boldsymbol{\theta} \in \mathbb{R}^d$. And Adam (Kingma & Ba, 2015) is the cornerstone of many learning algorithms to find the minimum of such a function. Nevertheless, Adam only tracks the EMA of $\mathbf{g}_t^2$; it cannot distinguish whether the magnitude is due to the signal or the noise. This blindness is detrimental, particularly for large-batch training where the local variance significantly dominates over the signal. Before proceeding, we first present several assumptions that are standard in stochastic optimization.

Assumption 3.1 (L-smoothness)**.** The objective function $f(\boldsymbol{\theta})$ is L-smooth on $\mathbb{R}^d$, *i.e.*, f is differentiable and the gradient of f is L-Lipschitz.

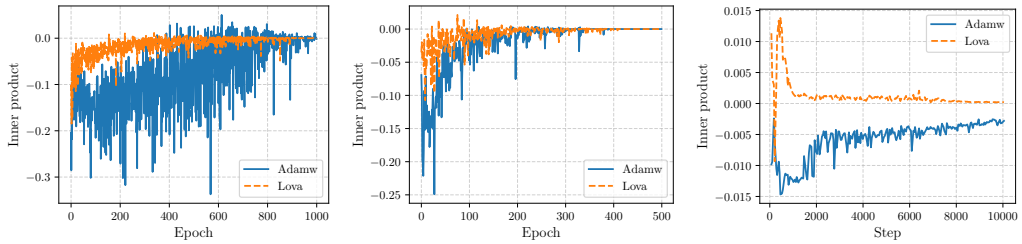
Assumption 3.2 (Bounded variance)**.** Let $\mathcal{F}_{t-1} = \sigma(\mathbf{g}_1, \ldots, \mathbf{g}_{t-1})$ be the σ-algebra representing the history up to step $t-1$. Assume the stochastic gradient $\mathbf{g}_t$ follows the decomposition, $\mathbf{g}_t = \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_{t-1}) + \boldsymbol{\xi}_t$, where $\mathbb{E}[\boldsymbol{\xi}_t \mid \mathcal{F}_{t-1}] = 0$ and $\mathbb{E}\left[\|\boldsymbol{\xi}_t\|^2 \mid \mathcal{F}_{t-1}\right] \leq \sigma^2$. Furthermore, assume $\{\boldsymbol{\xi}_t\}$ are mutually independent across steps, $\mathbb{E}[\langle \boldsymbol{\xi}_i, \boldsymbol{\xi}_j \rangle] = 0$ for all $i \neq j$.

Assumption 3.3 (Bounded gradient)**.** Assume that there exists a constant $G > 0$ such that $\|\mathbf{g}_t\|_2 \leq G$ for all $t \in [T]$.

3.1 Orthogonality of Signal and Local Variance

To justify the decomposition $\mathbf{v}_t^2 \approx \mathbf{m}_t^2 + (\mathbf{g}_t - \mathbf{m}_t)^2$, we must ensure that the inner-product between $\mathbf{m}_t$ and $\mathbf{g}_t - \mathbf{m}_t$ is negligible. Indeed, expanding the squared norm gives

$$\|\mathbf{g}_t\|^2 = \|\mathbf{m}_t\|^2 + \|\mathbf{g}_t - \mathbf{m}_t\|^2 + 2\langle \mathbf{m}_t, \mathbf{g}_t - \mathbf{m}_t \rangle.$$



(a) ResNet-56 on CIFAR-10 (b) WideResnet on CIFAR-100 (c) LLaMA-124M on FineWeb

Figure 2: Evolution of the inner product between $\mathbf{m}_t$ and $\mathbf{g}_t - \mathbf{m}_t$ of AdamW and Lova during training.

If the cross term vanishes asymptotically, the second moment decomposes into the squared momentum and the local variance. The following theorem shows that this is indeed the case for Adam's iterates.

Theorem 3.4. *Under Assumptions 3.1-3.3, the following holds for Adam with* $\eta_t = \eta/t^\delta$, $1-\beta_{1,t} = 1/t^\tau$, $0 < \tau < \delta \leq 1$,

$$|\mathbb{E}[\langle \mathbf{m}_t, \mathbf{g}_t - \mathbf{m}_t \rangle]| \leq \mathcal{O}(\frac{LG^2}{\epsilon} t^{-\kappa}),$$

where $\kappa = \min(\delta - \tau, \tau)$ *and achieves the supremum of* $1/2$ *by letting* $\delta = 1$ *and* $\tau = 1/2$. *Taking the limit as* $t \to \infty$, *it follows that*

$$\mathbb{E}[\|\mathbf{g}_t\|^2] \to \mathbb{E}[\|\mathbf{m}_t\|^2] + \mathbb{E}[\|\mathbf{g}_t - \mathbf{m}_t\|^2].$$

Remark 3.5. Theorem 3.4 holds for any constant $\beta_2 \in (0, 1)$ and is derived from the generic Adam with bias corrections. The small constant $\epsilon > 0$ is also employed as in practice. If we further assume $\beta_{1,t}^2 \leq \beta_2$ to bound the update $\mathbf{m}_t / \left(\sqrt{\mathbf{v}_t^2} + \epsilon\right)$ using Cauchy–Schwarz inequality (Zou et al., 2019; Wang et al., 2023), the upper bound would reduce to $\mathcal{O}(LG\sqrt{d}t^{-\kappa})$. But this requires careful adjustment of β_2 because $\beta_{1,t}$ is monotonically increasing and tends to be 1 as long as $t \to \infty$. Figure 2 visualizes how the inner product between $\mathbf{m}_t$ and $\mathbf{g}_t - \mathbf{m}_t$ evolves in realistic neural networks during training. While this value deviates far away from zero at the beginning, it finally stays around zero.

Algorithm 1 Lova algorithm

Input: learning rate $\{\eta_t\}$, initial parameter $\boldsymbol{\theta}_0$, objective function f
Input: $(\beta_1, \beta_2) = (0.9, 0.99)$, weight decay λ, $\epsilon > 0$
Initialize: $\mathbf{m}_0 \leftarrow 0, t \leftarrow 0$
for $t = 1$ to T **do**
  $\mathbf{g}_t \leftarrow \nabla_{\boldsymbol{\theta}} f(\boldsymbol{\theta}_{t-1})$
  $\mathbf{m}_t \leftarrow \beta_1 \mathbf{m}_{t-1} + (1-\beta_1)\mathbf{g}_t$
  $\mathbf{v}_t^2 \leftarrow \beta_2 \mathbf{m}_t^2 + (1-\beta_2)\left(\mathbf{g}_t - \mathbf{m}_t\right)^2$
  $\boldsymbol{\theta}_t \leftarrow (1-\eta_t\lambda)\boldsymbol{\theta}_{t-1} - \eta_t \frac{\mathbf{m}_t}{\sqrt{\mathbf{v}_t^2}+\epsilon}$
end for
return $\boldsymbol{\theta}_t$

The orthogonality indicates that, after sufficiently long training, the gradient square $\mathbf{g}_t^2$ is essentially composed of the momentum square $\mathbf{m}_t^2$ and the local variance $(\mathbf{g}_t - \mathbf{m}_t)^2$. It is known that $\mathbf{v}_t^2$ approximates $\mathbf{g}_t^2$ when the updates are stationary (Kingma & Ba, 2015). Consequently, we obtain

$$\mathbf{v}_t^2 \approx \mathbf{m}_t^2 + (\mathbf{g}_t - \mathbf{m}_t)^2,$$

which allows us to directly separate the signal from the local variance.

3.2 Algorithm

As Figure 2 suggests, the inner product between $\mathbf{m}_t$ and $\mathbf{g}_t - \mathbf{m}_t$ converges to zero very slowly in Adam. This is undesirable when the budget is limited. To accelerate the isolation between the signal and the noise, we propose a new algorithm, Lova, by estimating the second-order statistic $\mathbf{v}_t^2$ with the following equation:

$$\mathbf{v}_t^2 = \beta_2 \mathbf{m}_t^2 + (1-\beta_2)(\mathbf{g}_t - \mathbf{m}_t)^2,$$

where $0 < \beta_2 < 1$ should be interpreted as a smoothing factor that controls the signal and the noise. The full procedure is summarized in Algorithm 1. Similar to Theorem 3.4, the following result suggests that $\mathbf{m}_t$ and $\mathbf{g}_t - \mathbf{m}_t$ for updates of Lova are asymptotically orthogonal as well.

Theorem 3.6. *Under Assumptions 3.1-3.3, the following holds for Lova with $\eta_t = \eta/t^\delta$, $1-\beta_{1,t} = 1/t^\tau$, $0 < \beta_2 \leq 1$,$0 < \tau < \delta \leq 1$,*

$$|\mathbb{E}[\langle \mathbf{m}_t, \mathbf{g}_t - \mathbf{m}_t \rangle]| \leq \mathcal{O}(\frac{LG\sqrt{d}}{\sqrt{\beta_2}} t^{-\kappa}),$$

where $\kappa = \min(\delta - \tau, \tau)$ and achieves the supremum of $1/2$ by letting $\delta = 1$ and $\tau = 1/2$.

Remark 3.7. Compared to Theorem 3.4, this theorem implies that under the same assumptions, the inner product between $\mathbf{m}_t$ and $\mathbf{g}_t - \mathbf{m}_t$ decreases to zero at a faster rate because $\sqrt{d} \ll 1/\epsilon$ in practice. It is also empirically validated from Figure 2 that the value indeed quickly decays to zero.

Since $\mathbf{v}_t^2$ only consists of $\mathbf{m}_t$ and $\mathbf{g}_t$, Lova actually only needs to track the momentum buffer $\mathbf{m}_t$. This formulation allows Lova to track the long-term and consistent signal while being swift to sudden gradient changes. It is worth noting that when $\beta_2 = 1$ and $\epsilon = 0$, Lova recovers the sign-based optimizer SIGNUM (Bernstein et al., 2018). In contrast, when $\beta_2 = 0$, Lova only uses the local variance to adjust the learning rates and behaves like AdaBelief (Zhuang et al., 2020) except that the latter uses the EMA.

4 Convergence Analysis

It is well-known that Adam does not converge in theory, an issue that was first pointed out by Reddi et al. (2018). The central challenge lies in the "non-increasing" nature of effective learning rates. That is, the quantity $\Gamma(t) = \mathbf{v}_{t+1}/\eta_{t+1} - \mathbf{v}_t/\eta_t \geq 0$ does not necessarily hold for Adam, where we use $\mathbf{v}_t = \sqrt{\mathbf{v}_t^2}$ without further specification. To overcome this issue, AMSGrad (Reddi et al., 2018) and subsequent variants such as AdaBelief (Zhuang et al., 2020) explicitly impose another assumption that $\{\mathbf{v}_t\}$ is a monotonically increasing sequence. In the following theorem, we show that Lova under this condition achieves the same convergence rate as Adam.

Theorem 4.1 (Convex setting)**.** *Let $\boldsymbol{\theta} \in \Theta$, where $\Theta \subset \mathbb{R}^d$ is a convex feasible set. Assume that $f(\boldsymbol{\theta})$ is convex, lower-bounded and $\boldsymbol{\theta}^*$ is the optimal point such that $\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_\infty \leq D_\infty$. Consider the sequences $\{\mathbf{g}_t\}$ and $\{\mathbf{v}_t\}$ generated by Lova, let $\mathbf{v}_t = \max(\mathbf{v}_{t-1}, \mathbf{v}_t)$ for all $t \in [T]$ and $\eta_t = \eta/\sqrt{t}$. Then, under Assumption 3.3, for the sequence $\{\boldsymbol{\theta}_t\}$ generated by the algorithm, we have the following bound on the regret,*

$$\sum_{t=1}^{T} [f_t(\boldsymbol{\theta}_t) - f_t(\boldsymbol{\theta}^*)] \leq \left(\frac{D_\infty^2}{2\eta} + \frac{\beta_1 \eta}{(1-\beta_1)\beta_2} \right) d \left[\sqrt{4 - 3\beta_2} G + \epsilon \right] \sqrt{T} + \frac{\beta_1}{1-\beta_1} dG D_\infty.$$

Remark 4.2. While this theorem achieves the same regret $\mathcal{O}(\sqrt{T})$ as AMSGrad (Reddi et al., 2018) and AdaBelief (Zhuang et al., 2020), it is worth noting that our result holds for arbitrary constants β_1, β_2 without requiring $\beta_1^2 \leq \beta_2$ or decaying β_1.

In practice, however, we need to track the maximum of the historical $\mathbf{v}_t$ to ensure the monotonicity is valid, which inevitably incurs a larger memory footprint. Therefore, it is tempting to remove the monotonicity assumption. With properly chosen hyperparameters, we can show that Lova converges to a stationary point in the nonconvex setting.

Theorem 4.3 (Noncovex setting)**.** *Under Assumptions 3.1-3.3, we further assume the objective function $f(\boldsymbol{\theta})$ is lower bounded. The following inequality hold for $\eta_t = \eta/t^{3/4}, 1-\beta_{1,t} = 1/t^{1/2}$, and arbitrary constant $0 < \beta_2 < 1$,*

$$\min_{t=1,\cdots,T} \left\{ \mathbb{E}\left[\|\nabla f(\boldsymbol{\theta}_t)\|^2 \right] \right\} \leq \mathcal{O}\left(\frac{2\left[f(\boldsymbol{\theta}_0) - f(\boldsymbol{\theta}_T)\right]}{T^{1/4}} + \frac{8L^2\eta^2 d}{\beta_2 T^{1/4}} + \frac{\sigma^2}{T^{1/4}} + \frac{d \ln T}{\beta_2 T^{1/4}} + \frac{Ld}{\beta_2 T^{3/4}} \right).$$

Remark 4.4. Note that the final convergence rate $\mathcal{O}(\ln T/T^{1/4})$ does not match the lower bound of $\mathcal{O}(1/T^{1/2})$ for first-order optimizers (Arjevani et al., 2023) and is even slower than the rate of $\mathcal{O}(\ln T/T^{1/2})$ for Adam-type analyses (Reddi et al., 2018; Zou et al., 2019; Zhang et al., 2025a). This limitation is fundamental due to the fact that the denominator $\mathbf{v}_t$ is memoryless. As such, we cannot use cumulative adaptive scaling to compress the per-step sum into logarithmic growth. However, it should be highlighted that this bound is free of the polynomial dependence on $1/\epsilon$, which in practice is as small as 1.0×10^{-8} and can dominate Adam's bound. Moreover, it holds under standard assumptions without requiring other conditions such as symmetric noise or affine variance.

5 EXPERIMENTS

In this section, we present the experimental results on several popular tasks such as image classification and reinforcement learning. The code to reproduce the experimental results will be released at `https://anonymous.4open.science/r/lova2026-2D32`.

5.1 IMAGE CLASSIFICATION FROM SCRATCH

We start with realistic image datasets like CIFAR-10/100 (Krizhevsky et al., 2009) and ImageNet-1K (Deng et al., 2009). The baselines include AdamW (Loshchilov & Hutter, 2019), SIGNUM (Bernstein et al., 2018), Lion (Chen et al., 2023), MARS (Yuan et al., 2025), AdamS (Zhang et al., 2025a), and AdaBelief (Zhuang et al., 2020). All runs are repeated with three random seeds, and the hyperparameters of each optimizer generally follow the recommended settings. Without further specification, all models are trained with a batch size of 256 and the learning rate is decayed to zero using a cosine scheduler.

For CIFAR-10/100, we adopt three different architectures—ResNet-56 (He et al., 2016), WRN-28-10 (Zagoruyko & Komodakis, 2016), PyramidNet-110 (Han et al., 2017) and train them up to 200 epochs to ensure the features are fully learned. Table 1 shows that AdaBelief consistently performs better than AdamW on CIFAR-10/100. This is because AdaBelief takes a large step when the gradient closely aligns with the momentum. While MARS is designed to reduce gradient noise, it nevertheless worsens the performance for ResNet-56 on CIFAR-100. This implies that the gradient noise may be beneficial for generalization when the model cannot fully overfit the training data. In contrast, Lova improves over AdamW on CIFAR-10/100 for all architectures. Particularly, it improves the test accuracy of ResNet-56 by 2% on CIFAR-100.

For the large-scale ImageNet-1K dataset, we only apply basic data augmentations, namely, resizing and cropping images to a 224-pixel resolution and then normalizing them. Both models, including ResNet-18 and ResNet-50, are trained up to 100 epochs with eight GPUs. To simulate large-batch training, the accumulation step is set to 10, and the global batch size reaches 20480. The results, as depicted in Table 1, further validate the efficiency of Lova. It achieves a top-1 validation accuracy of 71.7% on ResNet-50, which surpasses AdamW by 5.8% and SIGNUM by 1.24%. These improvements are substantial for ImageNet-1K. Meanwhile, we can also observe that MARS and AdaBelief perform even worse than AdamW. These results imply that sign-based updates are useful in large-batch training and that, with careful design like Lova, it may perform even better.

Lastly, it is known that AdamW is easy to get trapped in sharp regions in the presence of noise. In this scenario, we demonstrate that Lova still performs better than AdamW. For this purpose, we trained ResNet-56 on CIFAR-10 corrupted with different severities of symmetric, asymmetric, and instance-dependent noise (IDN) (Wang et al., 2025). As shown in Table 2, Mars performs the worst because it reduces the gradient noise, which is particularly useful in the noisy setting. In contrast, Lion generally generalizes better than AdamW, and Lova achieves the highest test accuracy on different kinds of corruption. Notably, at the corruption level $\tau = 0.5$, Lova improves the test accuracy from 71.51% to 76.42% for the instance-dependent noise. Meanwhile, it also improves the test accuracy from 58.98% to 64.50% for the symmetric noise at $\tau = 0.7$. These results indicate that Lova is more robust to the noise.

5.2 NATURAL LANGUAGE PROCESSING

LLM Pretraining on Fineweb. We now evaluate the performance of Lova on LLM pretraining. Due to hardware constraints (8×NVIDIA-PRO5000-48G), we only train a LLaMA-like transformer with 124M parameters (Semenov et al., 2025) on the 100B tokens subset of FineWeb (Penedo et al., 2024). Throughout all runs, we use a sequence length of 512, and the micro-batch size is 64 with 4 gradient accumulations. Consequently, the batch size per step amounts to $512 \times 64 \times 4 \times 8 \approx$ 1M tokens. To observe how the performance varies under different durations, we train the model for {10k, 30k, 100k} iterations that correspond to {10B, 30B, 100B} tokens in total. Apart from AdamW, we also include Muon (Jordan et al., 2024) and D-Muon (Liu et al., 2025) optimizers as two baselines. Slightly different from previous studies (Semenov et al., 2025; Wen et al., 2026), we fix the peak learning rate and the weight decay to be 3.0×10^{-4} and 0.1 for all optimizers. This setting is consistent with the industrial training of Kimi-K2 (Team et al., 2025) and DeepSeek-V4-

Table 1: Test accuracy (%) on the task of image classification. Note that CIFAR-10/100 are trained using a batch size of 256, while the batch size of ImageNet-1K is 20480 by gradient accumulation and no scaling rule is applied to the learning rate to better illustrate how AdamW deteriorates in large-batch training.

		AdamW	MARS	AdamS	SIGNUM	AdaBelief	Lova
CIFAR-10	ResNet-56	92.36 ± 0.01	92.08 ± 0.05	92.49 ± 0.18	92.65 ± 0.09	92.41 ± 0.01	$\mathbf{92.72 \pm 0.02}$
	WRN-28-10	94.82 ± 0.01	94.95 ± 0.05	95.10 ± 0.08	94.96 ± 0.12	95.29 ± 0.07	$\mathbf{95.36 \pm 0.08}$
	PyramidNet-110	94.81 ± 0.13	94.82 ± 0.08	94.90 ± 0.17	94.77 ± 0.01	95.00 ± 0.01	$\mathbf{95.31 \pm 0.15}$
CIFAR-100	ResNet-56	67.14 ± 0.40	66.52 ± 0.12	68.49 ± 0.28	68.64 ± 0.48	67.69 ± 0.16	$\mathbf{69.11 \pm 0.33}$
	WRN-28-10	77.31 ± 0.01	77.41 ± 0.29	76.59 ± 0.10	76.44 ± 0.61	77.71 ± 0.39	$\mathbf{77.87 \pm 0.31}$
	PyramidNet-110	76.50 ± 0.04	77.18 ± 0.13	75.97 ± 0.08	76.91 ± 0.38	77.35 ± 0.01	$\mathbf{77.79 \pm 0.11}$
ImageNet-1K	ResNet-18	56.82 ± 0.21	40.96 ± 0.13	59.73 ± 0.15	62.35 ± 0.08	52.36 ± 0.12	$\mathbf{64.01 \pm 0.06}$
	ResNet-50	65.91 ± 0.12	51.16 ± 0.22	69.07 ± 0.09	70.47 ± 0.11	60.96 ± 0.35	$\mathbf{71.71 \pm 0.18}$

Table 2: Test accuracy (%) on CIFAR-10 for different levels of corruption ($\tau = 0.1, 0.3, 0.5, 0.7$).

		$\tau = 0.1$	$\tau = 0.3$	$\tau = 0.5$	$\tau = 0.7$
Symmetric	AdamW	85.38 ± 0.17	80.31 ± 0.43	74.58 ± 0.87	58.96 ± 1.15
	AdaBelief	85.29 ± 0.22	80.72 ± 0.29	74.04 ± 0.64	60.18 ± 0.98
	Lion	86.52 ± 0.25	82.57 ± 0.63	76.37 ± 0.79	59.51 ± 0.08
	MARS	83.38 ± 0.28	78.36 ± 0.21	72.21 ± 0.40	58.05 ± 0.17
	Lova (ours)	$\mathbf{86.97 \pm 0.07}$	$\mathbf{83.51 \pm 0.52}$	$\mathbf{77.73 \pm 0.17}$	$\mathbf{64.50 \pm 0.51}$
Asymmetric	AdamW	88.05 ± 0.03	84.30 ± 0.09	77.10 ± 0.36	62.48 ± 0.25
	AdaBelief	86.76 ± 0.07	84.11 ± 0.22	75.87 ± 0.03	62.19 ± 0.67
	Lion	88.21 ± 0.33	85.07 ± 0.14	76.11 ± 0.57	63.20 ± 0.28
	MARS	85.25 ± 0.26	82.83 ± 0.18	75.47 ± 0.01	60.62 ± 0.26
	Lova (ours)	$\mathbf{88.79 \pm 0.24}$	$\mathbf{86.74 \pm 0.23}$	$\mathbf{78.55 \pm 0.72}$	$\mathbf{63.97 \pm 0.21}$
IDN	AdamW	85.02 ± 0.30	80.71 ± 0.53	71.51 ± 0.26	42.69 ± 0.61
	AdaBelief	84.99 ± 0.07	80.57 ± 0.29	71.69 ± 0.98	43.43 ± 0.48
	Lion	86.83 ± 0.11	82.45 ± 0.99	73.36 ± 0.26	41.59 ± 0.17
	MARS	83.44 ± 0.12	78.67 ± 0.40	67.93 ± 0.38	42.36 ± 1.60
	Lova (ours)	$\mathbf{87.15 \pm 0.05}$	$\mathbf{83.66 \pm 0.06}$	$\mathbf{76.43 \pm 0.61}$	$\mathbf{44.72 \pm 0.52}$

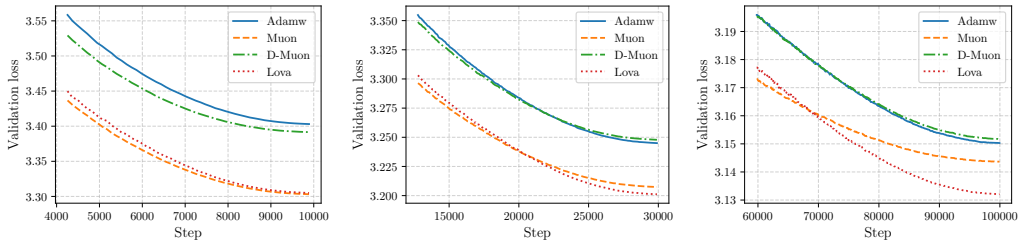
Flash (Xu et al., 2026), where they set the peak learning rate of D-Muon to be 2.0×10^{-4} and 2.7×10^{-4}.

As shown in Figure 3(a), Muon and D-Muon indeed enjoy a faster convergence speed than AdamW when the training tokens do not heavily exceed the critical value of 2.5B by Chinchilla's law (Hoffmann et al., 2022). Surprisingly, as we continue to train with more tokens, D-Muon finally performs even worse than AdamW, though Muon still performs better. But when compared against Lova, particularly in the over-training stage (namely, 30B and 100B tokens), Muon begins to lag behind Lova after $2/3$ of all steps. This observation implies the potential of Lova to surpass Muon in the regime of large-batch over-training.

Memory-Efficient Fine-Tuning. In this part, we further evaluate how LoVA performs in the popular fine-tuning task: RoBERTa-Large (Zhuang et al., 2021) on the GLUE benchmark (Wang et al., 2018). For GLUE fine-tuning, all tasks are trained for 3 epochs with a cosine learning rate scheduler. The maximum sequence length for each task is 256. Table 3 demonstrates the superiority of Lova and, in particular, it surpasses other optimizers by 3% for the task of CoLA.

5.3 Image Generation

Apart from the traditional tasks, we also evaluate Lova on diffusion transformers (DiTs) (Peebles & Xie, 2023) for image generation. Slightly different from diffusion models that use a convolutional U-Net as the backbone, DiTs replace it in latent diffusion models with a transformer. In our experiment, we train two class-conditional models (*i.e.* DiT-B/4 and DiT-L/4) at 256×256 image resolution on the ImageNet-1K dataset, and we only use basic data augmentation like horizontal flip. For AdamW and Lova, we use the same constant learning rate of 1.0×10^{-4}. Neither weight decay nor learning rate warmup is employed.



(a) 10B tokens (b) 30B tokens (c) 100B tokens

Figure 3: The validation loss curves of pre-training LLaMA-124M on FineWeb from 10B to 100B tokens. At each step, the gradient is estimated over one million tokens.

Table 3: Finetuning results of pretrained RoBERTa-Large model on the GLUE benchmark (higher is better). Note that Matthew's correlation coefficient and Pearson correlation are reported for CoLA and STS-B, while the classification accuracy is reported for the remaining tasks.

	CoLA	STS-B	MRPC	RTE	SST2	MNLI	QNLI	QQP
AdamW	64.05 ± 1.40	90.23 ± 0.72	87.98 ± 0.34	83.21 ± 0.25	96.15 ± 0.07	90.59 ± 0.05	94.52 ± 0.16	91.43 ± 0.05
AdamS	64.32 ± 1.42	92.27 ± 0.19	88.11 ± 0.17	84.12 ± 0.48	96.09 ± 0.64	90.56 ± 0.03	94.47 ± 0.08	91.82 ± 0.01
Lion	63.18 ± 0.87	91.38 ± 0.31	89.20 ± 0.68	81.76 ± 0.77	96.25 ± 0.05	89.74 ± 0.14	94.21 ± 0.03	91.57 ± 0.03
MARS	64.32 ± 1.41	91.46 ± 0.37	87.74 ± 1.04	79.24 ± 0.25	96.32 ± 0.16	90.62 ± 0.04	94.48 ± 0.01	91.21 ± 0.04
Lova (ours)	$\mathbf{67.12 \pm 0.16}$	$\mathbf{92.33 \pm 0.39}$	$\mathbf{89.58 \pm 0.52}$	$\mathbf{85.37 \pm 0.25}$	$\mathbf{96.49 \pm 0.07}$	$\mathbf{90.79 \pm 0.11}$	$\mathbf{94.59 \pm 0.14}$	$\mathbf{91.86 \pm 0.09}$

Following the common practice (Peebles & Xie, 2023), all results are reported on an EMA of DiT weights over training with a decay of 0.9999. The evaluation metrics include Fréchet Inception Distance (FID) (Heusel et al., 2017) and Inception Score (Salimans et al., 2016). To make a fair comparison, all values are obtained using the standard TensorFlow evaluation suite (Dhariwal & Nichol, 2021). For DiT-B/4, as shown in Table 4, Lova promotes the inception score from 18.45 to 29.50 and reduces the FID from 75.43 to 49.84. These improvements are significant and even more pronounced for DiT-L/4. The reason might be that the training process of diffusion models is highly noisy, and Lova is more capable of adapting itself to the gradient changes. In Figure 4, we also plot the curves of training loss. It is evident that Lova converges much faster than AdamW and achieves a lower training loss. But it is worth noting that the final performance is largely determined by the model size.

Table 4: Evaluation metrics.

		Inception Score	FID
DiT-B-4	AdamW	18.45	75.43
	Lova	**29.50** (+59.8%)	**49.84** (-33.9%)
DiT-L-4	AdamW	30.96	48.61
	Lova	**49.99** (+61.5%)	**29.54** (-39.2%)

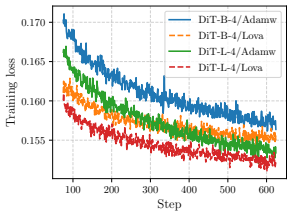
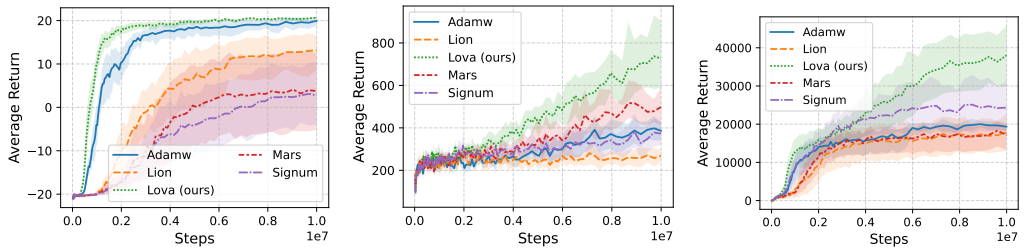


Figure 4: Training curves.

5.4 Deep Reinforcement Learning

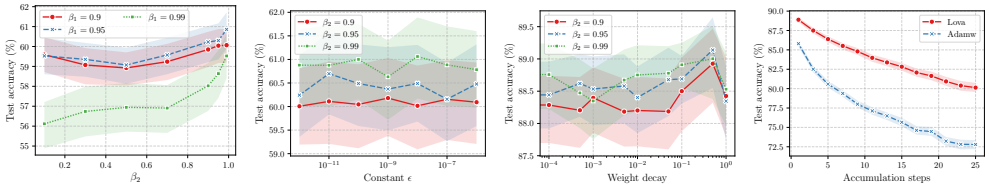
We further train reinforcement learning (RL) agents using the popular proximal policy optimization (PPO) (Schulman et al., 2017; Engstrom et al., 2019) algorithm. As a testbed, we use the non-modular CleanRL (Huang et al., 2022) library that is tailored for quickly prototyping novel features. Three different games (Pong, Gravitar, and RoadRunner) are selected from the Atari environments (Bellemare et al., 2013). Apart from AdamW, Lion (Chen et al., 2023), SIGNUM (Bernstein et al., 2018), and MARS (Yuan et al., 2025) are used as baselines as well. Both agents are trained up to ten million steps with a cosine learning rate decay.

The results in Figure 5 indicate that sign-based optimizers such as Lion and SIGNUM generally perform worse than AdamW on all games because they lose the information of gradient magnitude and cannot adapt to the quickly-changing loss landscape. On the other hand, MARS only achieves a higher return than AdamW for the Gravitar game. This observation is critically important since it implies that using recent gradients for variance reduction could be misleading in highly noisy training environments. In contrast, Lova consistently performs better than other baselines. For example, compared to AdamW, it approximately improves the average return by 75% and 100% for the games of Gravitar and RoadRunner.



(a) Pong (b) Gravitar (c) RoadRunner

Figure 5: Performance comparison of various optimizers on RL agents. The shaded region indicates 95% confidence intervals of ten trials.



(a) (b) (c) (d)

Figure 6: Sensitivity to hyperparameters. (a) investigates the effect of β_2 and (b) shows how the performance changes with the small constant $\epsilon > 0$. These values are reported using the classification results of ResNet-56 on CIFAR-100. In contrast, (c) and (d) illustrate the effects of the weight decay coefficient and the mini-batch size (here we use gradient accumulation to simulate a large batch size). And the values of (c) and (d) are based on the results of ResNet-56 on CIFAR-10.

5.5 Hyperparameter Sensitivity

Like AdamW, we set $\beta_1 = 0.9$ by default. The hyperparameter β_2 plays a dramatically important role in Lova. It controls the strength of the local variance in the denominator. To investigate how β_2 affects the generalization performance, we conduct another experiment with varying β_2 from $\{0.1, 0.2, \cdots, 0.9, 0.95, 0.99\}$. As shown in Figure 6(a), we can observe that $\beta_2 \geq 0.9$ generally performs much better than $\beta_2 < 0.9$. This implies that we need to choose a large value of β_2 so that the local variance can be strongly suppressed. In practice, we would recommend using $\beta_2 = 0.95$ for a large learning rate or small batch size and using $\beta_2 = 0.99$ for large-batch training. Apart from β_2, we also investigate the sensitivity of the small constant $\epsilon > 0$. Figure 6(b) shows that the final test accuracy remains nearly identical for ϵ ranging from 1.0×10^{-12} to 1.0×10^{-6}. Therefore, the influence of ϵ is negligible. Meanwhile, we also investigate the influence of the weight decay coefficient that varies in $\{0.0001, \cdots, 0.1, 0.5, 1.0\}$. Similar to AdamW, Figure 6(b) indicates that Lova also favors a large weight decay. Furthermore, we also evaluate how Lova changes with different batch sizes. Figure 6(c) indicates that under the same peak learning rate, the test accuracy of Lova and AdamW drops by 10% when the accumulation step increases from 1 to 25. Interestingly, we find that the gap between Lova and AdamW increases from 2.5% to 7.5% when more steps are accumulated.

6 Conclusion

In this paper, we developed an adaptive optimizer, Lova, to counteract the phenomenon of local variance dominance in gradient estimation. By interpolating between the squared momentum and the local variance, Lova tracks only a single momentum buffer, matching the memory footprint of SGD with momentum while retaining coordinate-wise adaptivity. Across various tasks such as image classification and reinforcement learning, we tested the practical performance of Lova. However, due to hardware constraints, our current work is limited to small-scale models and datasets. In the future, we will employ Lova in more challenging tasks such as post-training and few-shot learning.

AI USE STATEMENT

During the preparation of this work, we used Gemini-3.6 and ChatGPT-5.6 to check the mathematical proofs and the language grammar. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *Mathematical Programming*, 199(1):165–214, 2023.

Lukas Balles and Philipp Hennig. Dissecting Adam: The sign, magnitude and variance of stochastic gradients. In *International Conference on Machine Learning (ICML)*, pp. 404–413, 2018.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The Arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. SignSGD: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning (ICML)*, pp. 560–569, 2018.

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. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1–9, 2018.

Lizhang Chen, Bo Liu, Kaizhao Liang, and Qiang Liu. Lion secretly solves a constrained optimization: As Lyapunov predicts. In *International Conference on Learning Representations (ICLR)*, pp. 35404–35439, 2024.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, et al. Symbolic discovery of optimization algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 49205–49233, 2023.

Alexandre Défossez, Léon Bottou, Francis Bach, and Nicolas Usunier. A simple convergence proof of Adam and AdaGrad. *Transactions on Machine Learning Research*, pp. 1–30, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 8780–8794, 2021.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7), 2011.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep policy gradients: A case study on PPO and TRPO. In *International Conference on Learning Representations (ICLR)*, pp. 1–14, 2019.

Zhishuai Guo, Yi Xu, Wotao Yin, Rong Jin, and Tianbao Yang. A novel convergence analysis for algorithms of the Adam family. In *Annual Workshop on Optimization for Machine Learning*, pp. 1–13, 2021.

Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5927–5935, 2017.

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

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6626–6637, 2017.

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

Feihu Huang, Junyi Li, and Heng Huang. Super-Adam: faster and universal framework of adaptive gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 9074–9085, 2021.

Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and JoÃĢo GM AraÃšjo. CleanRL: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022.

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. URL `https://kellerjordan.github.io/posts/muon`.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, pp. 1–15, 2015.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Siyuan Li, Jiabao Pan, Yumou Liu, Zhuoli Ouyang, Xin Jin, Xinglong Xu, Jingxuan Wei, Shengye Pang, Jintao Che, Xuanhe Zhou, et al. OmniOpt: Taxonomy, geometry, and benchmarking of modern optimizers. *arXiv preprint arXiv:2607.04033*, 2026.

Kaizhao Liang, Lizhang Chen, Bo Liu, and Qiang Liu. Cautious optimizers: Improving training with one line of code. In *International Conference on Learning Representations (ICLR)*, pp. 106538–106563, 2026.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for LLM training. *arXiv preprint arXiv:2502.16982*, 2025.

Mingrui Liu, Wei Zhang, Francesco Orabona, and Tianbao Yang. Adam$^+$: A stochastic method with adaptive variance reduction. *arXiv preprint arXiv:2011.11985*, 2020.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, pp. 1–18, 2019.

Yang Luo, Xiaozhe Ren, Zangwei Zheng, Zhuo Jiang, Xin Jiang, and Yang You. Came: Confidence-guided adaptive memory efficient optimization. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 4442–4453, 2023.

William Peebles and Saining Xie. Scalable diffusion models with Transformers. In *International Conference on Computer Vision (ICCV)*, pp. 4172–4182, 2023.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 30811–30849, 2024.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of Adam and beyond. In *International Conference on Learning Representations (ICLR)*, pp. 1–23, 2018.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1–9, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Andrei Semenov, Matteo Pagliardini, and Martin Jaggi. Benchmarking optimizers for large language model pretraining. *arXiv preprint arXiv:2509.01440*, 2025.

Maria-Eleni Sfyraki and Jun-Kun Wang. Lions and Muons: Optimization via stochastic Frank-Wolfe under heavy-tailed noises. In *International Conference on Machine Learning (ICML)*, pp. 1–41, 2026.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning (ICML)*, pp. 4596–4604, 2018.

Kimi Team, Yifan Bai, Yiping Bao, Y Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, et al. Kimi K2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Tijmen Tieleman and Geoffrey Hinton. Divide the gradient by a running average of its recent magnitude. *Technical Report*, 2017.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, 2018.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 39006–39032, 2023.

Jialiang Wang, Xianming Liu, Xiong Zhou, Gangfeng Hu, Deming Zhai, Junjun Jiang, and Xiangyang Ji. Joint asymmetric loss for learning with noisy labels. In *International Conference on Computer Vision (ICCV)*, pp. 1947–1956, 2025.

Kaiyue Wen, David Hall, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them. In *International Conference on Learning Representations (ICLR)*, pp. 144731–144838, 2026.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Huizhuo Yuan, Yifeng Liu, Shuang Wu, Xun Zhou, and Quanquan Gu. Mars: Unleashing the power of variance reduction for training large models. In *International Conference on Machine Learning (ICML)*, pp. 73553–73587, 2025.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference (BMVC)*, pp. 1–15, 2016.

Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1–11, 2018.

Huishuai Zhang, Bohan Wang, and Luoxin Chen. AdamS: Momentum itself can be a normalizer for LLM pretraining and post-training. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 10730–10749, 2025a.

Tong Zhang, Jiangning Zhang, Zhucun Xue, Juntao Jiang, Yicheng Xu, Chengming Xu, Teng Hu, Xingyu Xie, Xiaobin Hu, Yabiao Wang, et al. Evolution of optimization methods: Algorithms, scenarios, and evaluations. *arXiv preprint arXiv:2604.12968*, 2026.

Yushun Zhang, Congliang Chen, Ziniu Li, Tian Ding, Chenwei Wu, Diederik Durk Kingma, Yinyu Ye, Zhi-Quan Luo, and Ruoyu Sun. Adam-mini: Use fewer learning rates to gain more. In *International Conference on Learning Representations (ICLR)*, pp. 28033–28063, 2025b.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 18795–18806, 2020.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. A robustly optimized BERT pre-training approach with post-training. In *Chinese National Conference on Computational Linguistics (CNCCL)*, pp. 1218–1227, 2021.

Fangyu Zou, Li Shen, Zequn Jie, Weizhong Zhang, and Wei Liu. A sufficient condition for convergences of Adam and RMSProp. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11119–11127, 2019.

A Appendix

In this section, we provide the missing proofs, and by default the weight decay λ is zero. For simplicity, we first provide several auxiliary lemmas as follows.

Lemma A.1. *For $\alpha \in (0,1)$, it holds that $f(t) = (1-t^{-\alpha})^t \le e^{-1}$ for all $t \ge 1$.*

Proof. It is easy to obtain that

$$\log f(t) = t\log(1-t^{-\alpha}) \le -t^{1-\alpha} \le -1,$$

thus concluding the proof. □

Lemma A.2. *For $\alpha \in [0,1]$, $(1-x)^{-\alpha} \le 1+2\alpha x$ holds for all $x \in [0,1/2]$.*

Proof. Let $g(x) = (1-x)^{-\alpha}$ for $x \in [0,1/2]$. Compute its first and second derivatives with respect to x,

$$g'(x) = \alpha(1-x)^{-\alpha-1} \quad \text{and} \quad g''(x) = \alpha(\alpha+1)(1-x)^{-\alpha-2}.$$

For $\alpha \in [0,1]$ and $x \in [0,1/2]$, we have $\alpha \ge 0$, $\alpha+1 > 0$, and $(1-x) > 0$. Therefore,

$$g''(x) \ge 0 \quad \text{for all } x \in [0,1/2].$$

Thus, $g(x)$ is convex on the interval $[0,1/2]$. Because $g(x)$ is convex on $[0,1/2]$, any point on the graph of $g(x)$ lies below the secant line connecting the endpoints $(0,g(0))$ and $(1/2,g(1/2))$. By the definition of convexity,

$$g(x) \le (1-2x)g(0) + 2x\, g\left(\frac{1}{2}\right) = 1 + 2x(2^{\alpha}-1).$$

Now we just need to show that $2^{\alpha}-1 \le \alpha$ for all $\alpha \in [0,1]$. Consider the function $h(\alpha) = 2^{\alpha}-1-\alpha$ for $\alpha \in [0,1]$, where $h(0) = 0$ and $h(1) = 0$. Since $h(\alpha)$ is convex in α, it lies on or below the line connecting its endpoints $(0,0)$ and $(1,0)$ on the interval $[0,1]$. Therefore,

$$h(\alpha) \le 0 \implies 2^{\alpha} - 1 \le \alpha \quad \text{for all } \alpha \in [0,1].$$

Substituting $2^{\alpha} - 1 \le \alpha$ back completes the proof. □

Lemma A.3. *Let $\tau \in (0,1)$ and $x \ge 0$. Then the following inequality holds*

$$(1+x)^{-\tau} \ge 1 - x\tau.$$

Proof. Define the function $g(x) = (1+x)^{-\tau}$ for $x \ge 0$. Computing the derivatives with respect to x,

$$g'(x) = -\tau(1+x)^{-\tau-1} \quad \text{and} \quad g''(x) = \tau(\tau+1)(1+x)^{-\tau-2}.$$

Since $\tau > 0$, $(\tau + 1) > 0$, and $(1 + x) > 0$ for all $x \geq 0$, we have

$$g''(x) > 0 \quad \text{for all } x \geq 0.$$

Thus, $g(x)$ is strictly convex on $[0, \infty)$. By the first-order condition for convex functions, any strictly convex function lies strictly above its tangent line at any point. The tangent line at $x_0 = 0$ is given by

$$h(x) = g(0) + g'(0)(x - 0) = 1 - \tau x.$$

By convexity, $g(x) \geq h(x)$ for all $x \geq 0$,

$$(1 + x)^{-\tau} \geq 1 - \tau x,$$

concluding the proof. □

Proof of Theorem 3.4. Because $\mathbf{m}_t = \beta_{1,t}\mathbf{m}_{t-1} + (1 - \beta_{1,t})\mathbf{g}_t$ is an exponential moving average of bounded gradients, it is also bounded. Namely, $\|\mathbf{m}_t\| \leq G$ for all $t \in [T]$. Now we subtract $\mathbf{m}_t$ from $\mathbf{g}_t$,

$$\mathbf{g}_t - \mathbf{m}_t = \mathbf{g}_t - (\beta_{1,t}\mathbf{m}_{t-1} + (1 - \beta_{1,t})\mathbf{g}_t) = \beta_{1,t}(\mathbf{g}_t - \mathbf{m}_{t-1}).$$

Substitute it into the inner product,

$$\begin{aligned}
\langle \mathbf{m}_t, \mathbf{g}_t - \mathbf{m}_t \rangle &= \langle (1 - \beta_{1,t})\mathbf{g}_t + \beta_{1,t}\mathbf{m}_{t-1},\ \beta_{1,t}(\mathbf{g}_t - \mathbf{m}_{t-1}) \rangle \\
&= \beta_{1,t}(1 - \beta_{1,t})\langle \mathbf{g}_t, \mathbf{g}_t - \mathbf{m}_{t-1} \rangle + \beta_{1,t}^2 \langle \mathbf{m}_{t-1}, \mathbf{g}_t - \mathbf{m}_{t-1} \rangle \\
&= \beta_{1,t}(1 - \beta_{1,t})\langle \mathbf{g}_t - \mathbf{m}_{t-1} + \mathbf{m}_{t-1}, \mathbf{g}_t - \mathbf{m}_{t-1} \rangle + \beta_{1,t}^2 \langle \mathbf{m}_{t-1}, \mathbf{g}_t - \mathbf{m}_{t-1} \rangle \\
&= \beta_{1,t}(1 - \beta_{1,t})\|\mathbf{g}_t - \mathbf{m}_{t-1}\|^2 + \beta_{1,t}\langle \mathbf{m}_{t-1}, \mathbf{g}_t - \mathbf{m}_{t-1} \rangle. \qquad (2)
\end{aligned}$$

Taking the conditional expectation $\mathbb{E}_{t-1}[\cdot] = \mathbb{E}[\cdot \mid \mathcal{F}_{t-1}]$ on both sides. Since $\mathbb{E}_{t-1}[\mathbf{g}_t] = \nabla f(\boldsymbol{\theta}_{t-1})$ and $\mathbf{m}_{t-1}$ is $\mathcal{F}_{t-1}$-measurable,

$$\begin{aligned}
&\mathbb{E}_{t-1}[\langle \mathbf{m}_t, \mathbf{g}_t - \mathbf{m}_t \rangle] = \beta_{1,t}(1 - \beta_{1,t})\mathbb{E}_{t-1}[\|\mathbf{g}_t - \mathbf{m}_{t-1}\|^2] + \beta_{1,t}\mathbb{E}_{t-1}[\langle \mathbf{m}_{t-1}, \mathbf{g}_t - \mathbf{m}_{t-1} \rangle] \\
&= \beta_{1,t}(1 - \beta_{1,t})\mathbb{E}_{t-1}[\|\mathbf{g}_t - \nabla f(\boldsymbol{\theta}_{t-1}) + \nabla f(\boldsymbol{\theta}_{t-1}) - \mathbf{m}_{t-1}\|^2] + \beta_{1,t}\langle \mathbf{m}_{t-1}, \nabla f(\boldsymbol{\theta}_{t-1}) - \mathbf{m}_{t-1} \rangle \\
&= \beta_{1,t}(1 - \beta_{1,t})\mathbb{E}_{t-1}[\|\mathbf{g}_t - \nabla f(\boldsymbol{\theta}_{t-1})\|^2 + \|\nabla f(\boldsymbol{\theta}_{t-1}) - \mathbf{m}_{t-1}\|^2] + \beta_{1,t}\langle \mathbf{m}_{t-1}, \nabla f(\boldsymbol{\theta}_{t-1}) - \mathbf{m}_{t-1} \rangle \\
&\leq \beta_{1,t}(1 - \beta_{1,t})\|\nabla f(\boldsymbol{\theta}_{t-1}) - \mathbf{m}_{t-1}\|^2 + \beta_{1,t}\langle \mathbf{m}_{t-1}, \nabla f(\boldsymbol{\theta}_{t-1}) - \mathbf{m}_{t-1} \rangle + \beta_{1,t}(1 - \beta_{1,t})\sigma^2.
\end{aligned}$$

Since $\mathbf{m}_{t-1}$ is random over the past $t - 1$ steps, take full expectation yields

$$\begin{aligned}
&\mathbb{E}[\langle \mathbf{m}_t, \mathbf{g}_t - \mathbf{m}_t \rangle] \leq \beta_{1,t}(1 - \beta_{1,t})\mathbb{E}[\|\nabla f(\boldsymbol{\theta}_{t-1}) - \mathbf{m}_{t-1}\|^2] + \beta_{1,t}\mathbb{E}[\langle \mathbf{m}_{t-1}, \nabla f(\boldsymbol{\theta}_{t-1}) - \mathbf{m}_{t-1} \rangle] + \beta_{1,t}(1 - \beta_{1,t})\sigma^2 \\
&\leq \beta_{1,t}(1 - \beta_{1,t})\mathbb{E}[\|\nabla f(\boldsymbol{\theta}_{t-1}) - \mathbb{E}[\mathbf{m}_{t-1}] + \mathbb{E}[\mathbf{m}_{t-1}] - \mathbf{m}_{t-1}\|^2] + \beta_{1,t}\mathbb{E}[\langle \mathbf{m}_{t-1}, \nabla f(\boldsymbol{\theta}_{t-1}) - \mathbf{m}_{t-1} \rangle] + \beta_{1,t}(1 - \beta_{1,t})\sigma^2 \\
&\leq 2\beta_{1,t}(1 - \beta_{1,t})\mathbb{E}[\|\nabla f(\boldsymbol{\theta}_{t-1}) - \mathbb{E}[\mathbf{m}_{t-1}]\|^2 + \|\mathbb{E}[\mathbf{m}_{t-1}] - \mathbf{m}_{t-1}\|^2] \\
&+ \beta_{1,t}\mathbb{E}[\langle \mathbf{m}_{t-1}, \nabla f(\boldsymbol{\theta}_{t-1}) - \mathbf{m}_{t-1} \rangle] + \beta_{1,t}(1 - \beta_{1,t})\sigma^2.
\end{aligned}$$

Expand $\mathbf{m}_{t-1}$ as a sum of past gradients,

$$\mathbf{m}_{t-1} = \sum_{k=1}^{t-1} w_{k,t-1}\mathbf{g}_k, \quad \text{where } w_{k,t-1} = (1 - \beta_{1,k}) \prod_{j=k+1}^{t-1} \beta_{1,j}.$$

Decompose $\mathbf{m}_{t-1}$ into its deterministic expectation $\mathbb{E}[\mathbf{m}_{t-1}]$ and stochastic noise fluctuation,

$$\mathbb{E}[\mathbf{m}_{t-1}] = \sum_{k=1}^{t-1} w_{k,t-1}\nabla f(\boldsymbol{\theta}_{k-1}), \quad \mathbf{m}_{t-1} - \mathbb{E}[\mathbf{m}_{t-1}] = \sum_{k=1}^{t-1} w_{k,t-1}\boldsymbol{\xi}_k.$$

By L-smoothness of f and update rule of Adam,

$$\begin{aligned}
&\|\nabla f(\boldsymbol{\theta}_t) - \nabla f(\boldsymbol{\theta}_k)\| \leq L\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_k\| \leq L \sum_{j=k+1}^{t} \|\boldsymbol{\theta}_j - \boldsymbol{\theta}_{j-1}\| \\
&= L \sum_{j=k+1}^{t} \eta_j \Big\| \frac{\mathbf{m}_j / (1 - \prod_{s=1}^{j} \beta_{1,s})}{\mathbf{v}_j / \sqrt{1 - \beta_2^j} + \epsilon} \Big\| \leq L \sum_{j=k+1}^{t} \eta_j \Big\| \frac{\mathbf{m}_j / (1 - \beta_{1,j}^j)}{\mathbf{v}_j / \sqrt{1 - \beta_2^j} + \epsilon} \Big\| \leq \frac{LG}{\epsilon(1 - e^{-1})} \sum_{j=k+1}^{t} \eta_j, \qquad (3)
\end{aligned}$$

where the last inequality is due to the fact that $1-\beta_{1,j} = j^{-\tau}$ and Lemma A.1. Using the fact that $\sum_{k=1}^{t-1} w_{k,t-1} = 1 - \prod_{j=1}^{t-1} \beta_{1,j}$, we have

$$\begin{aligned}\|\nabla f(\boldsymbol{\theta}_{t-1}) - \mathbb{E}[\mathbf{m}_{t-1}]\| &= \left\| \left(1 - \sum_{k=1}^{t-1} w_{k,t-1}\right) \nabla f(\boldsymbol{\theta}_{t-1}) + \sum_{k=1}^{t-1} w_{k,t-1}(\nabla f(\boldsymbol{\theta}_{t-1}) - \nabla f(\boldsymbol{\theta}_k)) \right\| \\ &\leq \left(\prod_{j=1}^{t-1} \beta_{1,j}\right) \|\nabla f(\boldsymbol{\theta}_{t-1})\| + \sum_{k=1}^{t-1} w_{k,t-1} \|\nabla f(\boldsymbol{\theta}_{t-1}) - \nabla f(\boldsymbol{\theta}_k)\| \\ &\leq \left(\prod_{j=1}^{t-1} \beta_{1,j}\right) G + \frac{LG}{\epsilon(1-e^{-1})} \sum_{k=1}^{t-1} w_{k,t-1} \sum_{j=k}^{t-1} \eta_j.\end{aligned}$$

Note that the first term decays exponentially to zero because $1-\beta_{1,j} = j^{-\tau}$ and thus can be ignored. Denote

$$S_t = \sum_{k=1}^{t-1} w_{k,t-1} \sum_{j=k}^{t-1} \eta_j,$$

and our goal is to bound this double sum. Splitting the inner sum $\sum_{j=k}^{t-1} \eta_j = \left(\sum_{j=k}^{t-2} \eta_j\right) + \eta_{t-1}$,

$$S_t = \sum_{k=1}^{t-1} w_{k,t-1} \sum_{j=k}^{t-2} \eta_j + \eta_{t-1} \sum_{k=1}^{t-1} w_{k,t-1}.$$

Now look at the first term. For $k = t-1$, the inner sum $\sum_{j=t-1}^{t-2} \eta_j = 0$. For $1 \leq k \leq t-2$, $w_{k,t-1} = \beta_{1,t-1} w_{k,t-2}$. Substituting these back gives

$$\sum_{k=1}^{t-1} w_{k,t-1} \sum_{j=k}^{t-2} \eta_j = \beta_{1,t-1} \sum_{k=1}^{t-2} w_{k,t-2} \sum_{j=k}^{t-2} \eta_j = \beta_{1,t-1} S_{t-1}.$$

Using the identity for $\sum_{k=1}^{t-1} w_{k,t-1}$ in the second term, we have $S_1 = 0$ and

$$S_t = \beta_{1,t-1} S_{t-1} + \eta_{t-1} \left(1 - \prod_{i=1}^{t-1} \beta_{1,i}\right) \quad \text{for } t \geq 2.$$

Since $0 < \prod_{i=1}^{t-1} \beta_{1,i} < 1$, we get the fundamental recursive inequality,

$$S_t \leq \beta_{1,t-1} S_{t-1} + \eta_{t-1} = \left(1 - (t-1)^{-\tau}\right) S_{t-1} + \eta (t-1)^{-\delta}.$$

We now prove by induction that $S_t \leq C_S t^{\tau-\delta}$ for some time-independent constant $C_S > 0$. Recall that $S_1 = 0 \leq C_S \cdot 1^{\tau-\delta}$ holds for any $C_S \geq 0$. Using $S_{t-1} \leq C_S (t-1)^{\tau-\delta}$, it follows that

$$S_t \leq C_S (1-(t-1)^{-\tau})(t-1)^{\tau-\delta} + \eta (t-1)^{-\delta} \leq C_S (t-1)^{\tau-\delta} - C_S (t-1)^{-\delta} + 2\eta t^{-\delta}.$$

According to Lemma A.2, $(t-1)^{\tau-\delta} = t^{\tau-\delta}(1-1/t)^{-(\delta-\tau)}$ for $t \geq 2$. Consequently,

$$(t-1)^{\tau-\delta} \leq t^{\tau-\delta} + 2(\delta-\tau) t^{\tau-\delta-1}.$$

Also, note that $(t-1)^{-\delta} \geq t^{-\delta}$. Substituting these in,

$$S_t \leq C_S \left[t^{\tau-\delta} + 2(\delta-\tau) t^{\tau-\delta-1}\right] - C_S t^{-\delta} + 2\eta t^{-\delta} = C_S t^{\tau-\delta} + t^{-\delta} \left[2\eta - C_S + 2C_S(\delta-\tau) t^{\tau-1}\right].$$

Since $0 < \tau < 1$, $t^{\tau-1} \leq 1$ for all $t \geq 2$, choosing $C_S \geq \frac{2\eta}{1-2^{\tau}(\delta-\tau)}$ guarantees that

$$2\eta - C_S + 2C_S(\delta-\tau) t^{\tau-1} \leq 0.$$

Thus, $S_t \leq C_S t^{\tau-\delta}$, completing the induction.

Now, we continue to bound

$$\mathbb{E}[\|\mathbf{m}_{t-1} - \mathbb{E}[\mathbf{m}_{t-1}]\|^2] = \mathbb{E}[\| \sum_{k=1}^{t-1} w_{k,t-1} \boldsymbol{\xi}_k \|^2] = \sum_{k=1}^{t-1} w_{k,t-1}^2 \mathbb{E}\left[\|\boldsymbol{\xi}_k\|^2\right] \leq \sigma^2 \sum_{k=1}^{t-1} w_{k,t-1}^2,$$

because $\boldsymbol{\xi}_k$ are zero-mean and mutually independent across steps ($\mathbb{E}[\langle \boldsymbol{\xi}_i \boldsymbol{\xi}_j \rangle] = 0$ for $i \neq j$). Consider the sum $P_{t+1} = \sum_{k=1}^{t} w_{k,t}^2$,

$$P_{t+1} = \sum_{k=1}^{t-1} w_{k,t}^2 + w_{t,t}^2 = \beta_{1,t}^2 \left(\sum_{k=1}^{t-1} w_{k,t-1}^2 \right) + (1 - \beta_{1,t})^2.$$

This gives the exact recurrence relation,

$$P_{t+1} = \beta_{1,t}^2 P_t + (1 - \beta_{1,t})^2 = (1 - t^{-\tau})^2 P_t + t^{-2\tau}.$$

Since $(1 - t^{-\tau})^2 = 1 - 2t^{-\tau} + t^{-2\tau} \leq 1 - t^{-\tau}$,

$$P_{t+1} \leq (1 - t^{-\tau}) P_t + t^{-2\tau}.$$

Assume $P_t \leq C_P t^{-\tau}$ for some time-independent constant $C_P > 1$. As a result,

$$P_{t+1} \leq (1 - t^{-\tau})(C_P t^{-\tau}) + t^{-2\tau} = C_P t^{-\tau} - C_P t^{-2\tau} + t^{-2\tau} = C_P t^{-\tau} - (C_P - 1) t^{-2\tau}.$$

We want to show that $P_{t+1} \leq C_P (t+1)^{-\tau}$ as well, which requires

$$C_P t^{-\tau} - (C_P - 1) t^{-2\tau} \leq C_P (t+1)^{-\tau}.$$

Rearranging terms, we obtain

$$(C_P - 1) t^{-2\tau} \geq C \left[t^{-\tau} - (t+1)^{-\tau} \right] = C_P t^{-\tau} \left[1 - \left(1 + \frac{1}{t} \right)^{-\tau} \right].$$

Divide both sides by $t^{-\tau}$,

$$(C_P - 1) t^{-\tau} \geq C_P \left[1 - \left(1 + \frac{1}{t} \right)^{-\tau} \right].$$

By Lemma A.3, we have

$$C_P \left[1 - \left(1 + \frac{1}{t} \right)^{-\tau} \right] \leq \frac{C_P \tau}{t}.$$

So it is sufficient to satisfy

$$(C_P - 1) t^{-\tau} \geq \frac{C_P \tau}{t} \implies C_P - 1 \geq C_P \tau t^{\tau - 1}.$$

Since $\tau < 1$, we have $\tau - 1 < 0$, meaning $t^{\tau - 1} \leq 1$ for all $t \geq 1$. Thus, we simply need

$$C_P - 1 \geq C_P \tau \implies C_P (1 - \tau) \geq 1 \implies C_P \geq \frac{1}{1 - \tau}.$$

Since $\tau \in (0, 1)$, $\frac{1}{1-\tau}$ is a finite positive constant. Picking $C_P = \max\left(2^\tau, \frac{1}{1-\tau}\right)$ completes the induction for all $t \geq 2$.

The last obstacle is the cross-term $\mathbb{E}[\langle \mathbf{m}_{t-1}, \nabla f(\boldsymbol{\theta}_{t-1}) - \mathbf{m}_{t-1} \rangle]$, which can be further decomposed into

$$\mathbb{E}[\langle \mathbf{m}_{t-1}, \nabla f(\boldsymbol{\theta}_{t-1}) - \mathbf{m}_{t-1} \rangle] = \mathbb{E}[\langle \mathbf{m}_{t-1}, \nabla f(\boldsymbol{\theta}_{t-1}) - \mathbb{E}[\mathbf{m}_{t-1}] \rangle] + \mathbb{E}[\langle \mathbf{m}_{t-1}, \mathbb{E}[\mathbf{m}_{t-1}] - \mathbf{m}_{t-1} \rangle].$$

Since

$$\mathbb{E}[\langle \mathbf{m}_{t-1}, \nabla f(\boldsymbol{\theta}_{t-1}) - \mathbb{E}[\mathbf{m}_{t-1}] \rangle] \leq \sqrt{\mathbb{E}[\mathbf{m}_{t-1}^2]} \sqrt{\mathbb{E}[\|\nabla f(\boldsymbol{\theta}_{t-1}) - \mathbb{E}[\mathbf{m}_{t-1}\|^2]} \leq \mathcal{O}(\frac{LG^2}{\epsilon} t^{\tau - \delta}),$$

and

$$\begin{aligned}
\mathbb{E}[\langle \mathbf{m}_{t-1}, \mathbb{E}[\mathbf{m}_{t-1}] - \mathbf{m}_{t-1} \rangle] &= \mathbb{E}\left[\langle \mathbb{E}[\mathbf{m}_{t-1}] + (\mathbf{m}_{t-1} - \mathbb{E}[\mathbf{m}_{t-1}]),\ \mathbb{E}[\mathbf{m}_{t-1}] - \mathbf{m}_{t-1} \rangle \right] \\
&= \langle \mathbb{E}[\mathbf{m}_{t-1}],\ \mathbb{E}[\mathbb{E}[\mathbf{m}_{t-1}] - \mathbf{m}_{t-1}] \rangle - \mathbb{E}\left[\|\mathbf{m}_{t-1} - \mathbb{E}[\mathbf{m}_{t-1}]\|^2 \right] \\
&= -\mathbb{E}\left[\|\mathbf{m}_{t-1} - \mathbb{E}[\mathbf{m}_{t-1}]\|^2 \right] \leq 0,
\end{aligned}$$

it follows that

$$\mathbb{E}[\langle \mathbf{m}_{t-1}, \nabla f(\boldsymbol{\theta}_{t-1}) - \mathbf{m}_{t-1} \rangle] \leq \mathcal{O}(\frac{LG^2}{\epsilon} t^{\tau-\delta}).$$

Gathering together, we have

$$\begin{aligned}\mathbb{E}[\langle \mathbf{m}_t, \mathbf{g}_t - \mathbf{m}_t \rangle] &\leq t^{-\tau}\mathbb{E}[\|\nabla f(\boldsymbol{\theta}_t) - \mathbb{E}[\mathbf{m}_{t-1}]\|^2 + \|\mathbb{E}[\mathbf{m}_{t-1}] - \mathbf{m}_{t-1}\|^2] + \mathbb{E}[\langle \mathbf{m}_{t-1}, \nabla f(\boldsymbol{\theta}_{t-1}) - \mathbf{m}_{t-1} \rangle] + t^{-\tau}\sigma^2 \\ &\leq \mathcal{O}(\frac{L^2G^2}{\epsilon^2} t^{\tau-2\delta} + t^{-2\tau}) + \mathcal{O}(\frac{LG^2}{\epsilon} t^{\tau-\delta}) + \mathcal{O}(t^{-\tau}\sigma^2) = \mathcal{O}(\frac{LG^2}{\epsilon} t^{-\kappa}),\end{aligned}$$

where $\kappa = \min(\delta - \tau, \tau)$ and achieves the supremum of 1/2 by letting $\delta = 1$ and $\tau = 1/2$.

On the other hand, it follows from Equation (2) that

$$\langle \mathbf{m}_t, \mathbf{m}_t - \mathbf{g}_t \rangle = -\beta_{1,t}(1 - \beta_{1,t})\|\mathbf{g}_t - \mathbf{m}_{t-1}\|^2 + \beta_{1,t}\langle \mathbf{m}_{t-1}, \mathbf{m}_{t-1} - \mathbf{g}_t \rangle.$$

Consequently,

$$\mathbb{E}[\langle \mathbf{m}_t, \mathbf{m}_t - \mathbf{g}_t \rangle] \leq \mathbb{E}[\langle \mathbf{m}_{t-1}, \mathbf{m}_{t-1} - \nabla f(\boldsymbol{\theta}_{t-1}) \rangle].$$

Using previous results, we have

$$\begin{aligned}\mathbb{E}[\langle \mathbf{m}_t, \mathbf{m}_t - \mathbf{g}_t \rangle] &\leq \mathbb{E}[\|\mathbb{E}[\mathbf{m}_{t-1}] - \mathbf{m}_{t-1}\|^2] + \mathbb{E}[\langle \mathbf{m}_{t-1}, \mathbf{m}_{t-1} - \nabla f(\boldsymbol{\theta}_{t-1}) \rangle] \\ &\leq \mathcal{O}(t^{-\tau}) + \mathcal{O}(\frac{LG^2}{\epsilon} t^{\tau-\delta}) = \mathcal{O}(\frac{LG^2}{\epsilon} t^{-\kappa}),\end{aligned}$$

where again $\kappa = \min(\delta - \tau, \tau)$ and achieves the supremum of 1/2 by letting $\delta = 1$ and $\tau = 1/2$. Therefore,

$$|\mathbb{E}[\langle \mathbf{m}_t, \mathbf{g}_t - \mathbf{m}_t \rangle]| \leq \mathcal{O}(\frac{LG^2}{\epsilon} t^{-\kappa}),$$

concluding the proof. □

Proof of Theorem 3.6. The proof follows the same steps as Theorem 3.4 with the exception of Equation (3). By L-smoothness of f and update rule of Lova,

$$\|\nabla f(\boldsymbol{\theta}_t) - \nabla f(\boldsymbol{\theta}_k)\| \leq L\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_k\| \leq L \sum_{j=k+1}^{t} \|\boldsymbol{\theta}_j - \boldsymbol{\theta}_{j-1}\| = L \sum_{j=k+1}^{t} \eta_j \|\frac{\mathbf{m}_j}{\mathbf{v}_j + \epsilon}\| \leq \frac{L\sqrt{d}}{\sqrt{\beta_2}} \sum_{j=k+1}^{t} \eta_j.$$

Using the same arguments in the proof of Theorem 3.4, it is easy to conclude that

$$|\mathbb{E}[\langle \mathbf{m}_t, \mathbf{g}_t - \mathbf{m}_t \rangle]| \leq \mathcal{O}(\frac{LG\sqrt{d}}{\sqrt{\beta_2}} t^{-\kappa}),$$

where $\kappa = \min(\delta - \tau, \tau)$ and achieves the supremum of 1/2 by letting $\delta = 1$ and $\tau = 1/2$. □

Proof of Theorem 4.1. Because $\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1)\mathbf{g}_t$ is an exponential moving average of bounded gradients, it is also bounded. Namely, $\|\mathbf{m}_t\|_\infty \leq \|\mathbf{m}_t\|_2 \leq G$ for all $t \in [T]$. Consequently, the memoryless denominator $\mathbf{v}_t$ has strict bounds as well. For a single coordinate i,

$$\mathbf{v}_{t,i}^2 = \beta_2 \mathbf{m}_{t,i}^2 + (1 - \beta_2)(\mathbf{g}_{t,i} - \mathbf{m}_{t,i})^2 \leq \beta_2 G^2 + (1 - \beta_2)(2G)^2 = (4 - 3\beta_2)G^2.$$

Since we need to add an $\epsilon > 0$ in the denominator for numerical stability, in the following context we use $\mathbf{v}_t = \mathbf{v}_t + \epsilon$ for convenience. Therefore, for any $t \in [T]$,

$$\|\mathbf{v}_t\|_\infty \leq \sqrt{4 - 3\beta_2} G + \epsilon \triangleq V_\infty.$$

Since the objective function $f_t(\boldsymbol{\theta}_t)$ is convex such that $f_t(\boldsymbol{\theta}_t) - f_t(\boldsymbol{\theta}^*) \leq \langle \mathbf{g}_t, \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle$, our goal is to bound the regret

$$R_T = \sum_{t=1}^{T} (f_t(\boldsymbol{\theta}_t) - f_t(\boldsymbol{\theta}^*)) \leq \sum_{t=1}^{T} \langle \mathbf{g}_t, \boldsymbol{\theta}_t - \boldsymbol{\theta}^* \rangle = \sum_{t=1}^{T} \sum_{i=1}^{d} \mathbf{g}_{t,i}(\boldsymbol{\theta}_{t,i} - \boldsymbol{\theta}_i^*).$$

First, we express the gradient $\mathbf{g}_t$ strictly in terms of the momentum $\mathbf{m}_t$ by rearranging the momentum update rule $\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1)\mathbf{g}_t$,

$$\mathbf{g}_t = \frac{1}{1 - \beta_1} \mathbf{m}_t - \frac{\beta_1}{1 - \beta_1} \mathbf{m}_{t-1}.$$

Substituting this into the regret equation yields

$$\sum_{i=1}^{d}\sum_{t=1}^{T}\mathbf{g}_{t,i}(\boldsymbol{\theta}_{t,i}-\boldsymbol{\theta}_i^*)=\sum_{i=1}^{d}\frac{1}{1-\beta_1}\sum_{t=1}^{T}\mathbf{m}_{t,i}(\boldsymbol{\theta}_{t,i}-\boldsymbol{\theta}_i^*)-\frac{\beta_1}{1-\beta_1}\sum_{t=1}^{T}\mathbf{m}_{t-1,i}(\boldsymbol{\theta}_{t,i}-\boldsymbol{\theta}_i^*).$$

By shifting the index on the second sum and applying summation by parts (using $\mathbf{m}_0=0$), we get

$$\sum_{i=1}^{d}\sum_{t=1}^{T}\mathbf{g}_{t,i}(\boldsymbol{\theta}_{t,i}-\boldsymbol{\theta}_i^*)=\sum_{i=1}^{d}\sum_{t=1}^{T}\mathbf{m}_{t,i}(\boldsymbol{\theta}_{t,i}-\boldsymbol{\theta}_i^*)+\frac{\beta_1}{1-\beta_1}\left[\mathbf{m}_{T,i}(\boldsymbol{\theta}_{T,i}-\boldsymbol{\theta}_i^*)+\sum_{t=1}^{T-1}\mathbf{m}_{t,i}(\boldsymbol{\theta}_{t,i}-\boldsymbol{\theta}_{t+1,i})\right].$$

Let us evaluate the bracketed term on the right-hand side. Recall that the update rule is

$$\boldsymbol{\theta}_{t+1}=\boldsymbol{\theta}_t-\eta_t\boldsymbol{\mu}_t,$$

where $\boldsymbol{\mu}_t=\frac{\mathbf{m}_t}{\mathbf{v}_t}$ and $\|\boldsymbol{\mu}_t\|_\infty\le 1/\sqrt{\beta_2}$ for all $t\in[T]$. Substituting this in,

$$\sum_{i=1}^{d}\sum_{t=1}^{T-1}\mathbf{m}_{t,i}(\boldsymbol{\theta}_{t,i}-\boldsymbol{\theta}_{t+1,i})=\sum_{i=1}^{d}\sum_{t=1}^{T-1}\eta_t\mathbf{m}_{t,i}\boldsymbol{\mu}_{t,i}.$$

Since $\mathbf{m}_{t,i}=\mathbf{v}_{t,i}\boldsymbol{\mu}_{t,i}$, we have $\mathbf{m}_{t,i}\boldsymbol{\mu}_{t,i}=\mathbf{v}_{t,i}\boldsymbol{\mu}_{t,i}^2\ge 0$. Because $\mathbf{v}_{t,i}\le V_\infty$ and $\boldsymbol{\mu}_{t,i}^2\le 1/\beta_2$, we can bound this sequence,

$$\sum_{i=1}^{d}\sum_{t=1}^{T-1}\eta_t\mathbf{v}_{t,i}\boldsymbol{\mu}_{t,i}^2\le\frac{dV_\infty}{\beta_2}\sum_{t=1}^{T-1}\eta_t\le\frac{dV_\infty}{\beta_2}2\eta\sqrt{T}.$$

Moreover, the terminal term is trivially bounded,

$$\sum_{i=1}^{d}m_{T,i}(\boldsymbol{\theta}_{T,i}-\boldsymbol{\theta}_i^*)\le dG_\infty D_\infty.$$

Now we bound the main term $\sum_{i=1}^{d}\sum_{t=1}^{T}\mathbf{m}_{t,i}(\boldsymbol{\theta}_{t,i}-\boldsymbol{\theta}_i^*)$. Using $\mathbf{m}_t=\mathbf{v}_t\boldsymbol{\mu}_t$, we write

$$\sum_{i=1}^{d}\sum_{t=1}^{T}\mathbf{m}_{t,i}(\boldsymbol{\theta}_{t,i}-\boldsymbol{\theta}_i^*)=\sum_{i=1}^{d}\sum_{t=1}^{T}\mathbf{v}_{t,i}\boldsymbol{\mu}_{t,i}(\boldsymbol{\theta}_{t,i}-\boldsymbol{\theta}_i^*).$$

From the distance-squared expansion of the update rule, we isolate $\boldsymbol{\mu}_{t,i}(\boldsymbol{\theta}_{t,i}-\boldsymbol{\theta}_i^*)$,

$$\boldsymbol{\mu}_{t,i}(\boldsymbol{\theta}_{t,i}-\boldsymbol{\theta}_i^*)=\frac{(\boldsymbol{\theta}_{t,i}-\boldsymbol{\theta}_i^*)^2-(\boldsymbol{\theta}_{t+1,i}-\boldsymbol{\theta}_i^*)^2}{2\eta_t}+\frac{\eta_t}{2}\boldsymbol{\mu}_{t,i}^2.$$

Multiplying by $\mathbf{v}_{t,i}$ and summing over all T iterations,

$$\sum_{i=1}^{d}\sum_{t=1}^{T}\mathbf{v}_{t,i}\boldsymbol{\mu}_{t,i}(\boldsymbol{\theta}_{t,i}-\boldsymbol{\theta}_i^*)=\sum_{i=1}^{d}\sum_{t=1}^{T}\frac{\mathbf{v}_{t,i}}{2\eta_t}\left[(\boldsymbol{\theta}_{t,i}-\boldsymbol{\theta}_i^*)^2-(\boldsymbol{\theta}_{t+1,i}-\boldsymbol{\theta}_i^*)^2\right]+\sum_{i=1}^{d}\sum_{t=1}^{T}\frac{\eta_t}{2}\mathbf{v}_{t,i}\boldsymbol{\mu}_{t,i}^2.$$

The second term on the right is bounded exactly as before,

$$\sum_{i=1}^{d}\sum_{t=1}^{T}\frac{\eta_t}{2}\mathbf{v}_{t,i}\boldsymbol{\mu}_{t,i}^2\le\frac{dV_\infty}{2\beta_2}\sum_{t=1}^{T}\eta_t\le\frac{dV_\infty}{\beta_2}\eta\sqrt{T}.$$

For the first term, we expand the telescoping sum. This requires rearranging the terms to group by $(\boldsymbol{\theta}_{t,i}-\boldsymbol{\theta}_i^*)^2$,

$$\sum_{t=1}^{T}\frac{\mathbf{v}_{t,i}}{2\eta_t}\left[(\boldsymbol{\theta}_{t,i}-\boldsymbol{\theta}_i^*)^2-(\boldsymbol{\theta}_{t+1,i}-\boldsymbol{\theta}_i^*)^2\right]=\frac{\mathbf{v}_{1,i}(\boldsymbol{\theta}_{1,i}-\boldsymbol{\theta}_i^*)^2}{2\eta}+\sum_{t=2}^{T}(\boldsymbol{\theta}_{t,i}-\boldsymbol{\theta}_i^*)^2\left(\frac{\mathbf{v}_{t,i}}{2\eta_t}-\frac{\mathbf{v}_{t-1,i}}{2\eta_{t-1}}\right).$$

Applying the domain bound $(\boldsymbol{\theta}_{t,i}-\boldsymbol{\theta}_i^*)^2\le D_\infty^2$ and the monotonicity of $\mathbf{v}_{t,i}$, we arrive at the absolute upper bound,

$$\sum_{t=1}^{T}\frac{\mathbf{v}_{t,i}}{2\eta_t}\left[(\boldsymbol{\theta}_{t,i}-\boldsymbol{\theta}_i^*)^2-(\boldsymbol{\theta}_{t+1,i}-\boldsymbol{\theta}_i^*)^2\right]\le\frac{V_\infty D_\infty^2}{2\eta}+\frac{D_\infty^2}{2}\sum_{t=2}^{T}\left(\frac{\mathbf{v}_{t,i}}{\eta_t}-\frac{\mathbf{v}_{t-1,i}}{\eta_{t-1}}\right)\le\frac{V_\infty D_\infty^2\sqrt{T}}{2\eta}.$$

Combining different parts together, it follows that

$$R_T\le\left(\frac{D_\infty^2}{2\eta}+\frac{\beta_1\eta}{(1-\beta_1)\beta_2}\right)d\left[\sqrt{4-3\beta_2}G+\epsilon\right]\sqrt{T}+\frac{\beta_1}{1-\beta_1}dGD_\infty,$$

concluding the proof. □

Proof of Theorem 4.3. We start by applying the L-smoothness assumption to the update step $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \eta_t \frac{\mathbf{m}_t}{\mathbf{v}_t}$,

$$f(\boldsymbol{\theta}_t) \leq f(\boldsymbol{\theta}_{t-1}) - \eta_t \left\langle \nabla f(\boldsymbol{\theta}_{t-1}), \frac{\mathbf{m}_t}{\mathbf{v}_t + \epsilon} \right\rangle + \frac{L\eta_t^2}{2} \left\| \frac{\mathbf{m}_t}{\mathbf{v}_t + \epsilon} \right\|_2^2.$$

Since $\mathbf{v}_t^2 = \beta_2 \mathbf{m}_t^2 + (1-\beta_2)(\mathbf{g}_t - \mathbf{m}_t)^2$, we can immediately bound the second-order term as follows

$$f(\boldsymbol{\theta}_t) \leq f(\boldsymbol{\theta}_{t-1}) - \eta_t \left\langle \nabla f(\boldsymbol{\theta}_{t-1}), \frac{\mathbf{m}_t}{\mathbf{v}_t + \epsilon} \right\rangle + \frac{L\eta_t^2 d}{2\beta_2}.$$

For a single coordinate i,

$$\mathbf{v}_{t,i}^2 = \beta_2 \mathbf{m}_{t,i}^2 + (1-\beta_2)(\mathbf{g}_{t,i} - \mathbf{m}_{t,i})^2 \leq \beta_2 G^2 + (1-\beta_2)(2G)^2 = (4 - 3\beta_2)G^2.$$

Since we need to add an $\epsilon > 0$ in the denominator for numerical stability, in the following context we use $\mathbf{v}_t = \mathbf{v}_t + \epsilon$ for convenience. Therefore, for any $t \in [T]$,

$$\|\mathbf{v}_t\|_\infty \leq \sqrt{4 - 3\beta_2} G + \epsilon \triangleq V_\infty.$$

Now we continue to bound the second term,

$$-\left\langle \nabla f(\boldsymbol{\theta}_{t-1}), \frac{\mathbf{m}_t}{\mathbf{v}_t} \right\rangle = \sum_{i=1}^d \left[-\nabla f(\boldsymbol{\theta}_{t-1})_i \frac{\mathbf{m}_{t,i}}{\mathbf{v}_{t,i}} \right] = \sum_{i=1}^d \left[-\frac{\mathbf{m}_{t,i}^2}{\mathbf{v}_{t,i}} + (\mathbf{m}_{t,i} - \nabla f(\boldsymbol{\theta}_{t-1})_i) \frac{\mathbf{m}_{t,i}}{\mathbf{v}_{t,i}} \right]$$

$$\leq \sum_{i=1}^d \left[-\frac{\mathbf{m}_{t,i}^2}{V_\infty} + (\mathbf{m}_{t,i} - \nabla f(\boldsymbol{\theta}_{t-1})_i) \frac{\mathbf{m}_{t,i}}{\mathbf{v}_{t,i}} \right] \leq -\frac{\|\mathbf{m}_t\|_2^2}{V_\infty} + \sum_{i=1}^d (\mathbf{m}_{t,i} - \nabla f(\boldsymbol{\theta}_{t-1})_i) \frac{\mathbf{m}_{t,i}}{\mathbf{v}_{t,i}}.$$

Note that

$$\sum_{i=1}^d (\mathbf{m}_{t,i} - \nabla f(\boldsymbol{\theta}_{t-1})_i) \frac{\mathbf{m}_{t,i}}{\mathbf{v}_{t,i}} \leq \sum_{i=1}^d |\mathbf{m}_{t,i} - \nabla f(\boldsymbol{\theta}_{t-1})_i| \cdot |\frac{\mathbf{m}_{t,i}}{\mathbf{v}_{t,i}}| \leq \|\mathbf{m}_t - \nabla f(\boldsymbol{\theta}_{t-1})\|_1 \cdot \frac{1}{\sqrt{\beta_2}}$$

$$\leq \|\mathbf{m}_t - \nabla f(\boldsymbol{\theta}_{t-1})\|_2 \cdot \frac{\sqrt{d}}{\sqrt{\beta_2}} \leq \frac{t^\kappa}{2} \|\mathbf{m}_t - \nabla f(\boldsymbol{\theta}_{t-1})\|_2^2 + \frac{d}{2t^\kappa \beta_2},$$

where the last inequality is due to the fact that $ab \leq \frac{t^\kappa}{2} a^2 + \frac{b^2}{2t^\kappa}$ for any $\kappa > 0$. Moreover, since $\|a\|_2^2 \geq \|a+b\|_2^2/2 - \|b\|_2^2$, by letting $a = \mathbf{m}_t$ and $b = \nabla f(\boldsymbol{\theta}_{t-1}) - \mathbf{m}_t$, it follows that

$$\|\mathbf{m}_t\|_2^2 \geq \frac{1}{2} \|\nabla f(\boldsymbol{\theta}_{t-1})\|_2^2 - \|\nabla f(\boldsymbol{\theta}_{t-1}) - \mathbf{m}_t\|_2^2.$$

Consequently, we have

$$-\left\langle \nabla f(\boldsymbol{\theta}_{t-1}), \frac{\mathbf{m}_t}{\mathbf{v}_t} \right\rangle \leq -\frac{1}{2V_\infty} \|\nabla f(\boldsymbol{\theta}_{t-1})\|_2^2 + \left(\frac{1}{V_\infty} + \frac{t^\kappa}{2} \right) \|\mathbf{m}_t - \nabla f(\boldsymbol{\theta}_{t-1})\|_2^2 + \frac{d}{2\beta_2 t^\kappa}.$$

Modify the momentum update rule to the form below

$$\mathbf{m}_t = \mathbf{g}_t + \beta_{1,t}(\mathbf{m}_{t-1} - \mathbf{g}_t) = (1 - \beta_{1,t})\mathbf{g}_t + \beta_{1,t}\mathbf{m}_{t-1}.$$

Notice that algebraically, it can be further written as

$$\mathbf{m}_t - \nabla f(\boldsymbol{\theta}_{t-1}) = (1-\beta_{1,t})(\mathbf{g}_t - \nabla f(\boldsymbol{\theta}_{t-1})) + \beta_{1,t}(\mathbf{m}_{t-1} - \nabla f(\boldsymbol{\theta}_{t-1})).$$

When taking the conditioned expectation, the unbiased gradient property $\mathbb{E}[\mathbf{g}_t] = \nabla f(\boldsymbol{\theta}_{t-1})$ causes the cross-term between $(\mathbf{g}_t - \nabla f(\boldsymbol{\theta}_{t-1}))$ and $(\mathbf{m}_{t-1} - \nabla f(\boldsymbol{\theta}_{t-1}))$ to vanish directly, yielding

$$\mathbb{E}[\|\mathbf{m}_t - \nabla f(\boldsymbol{\theta}_{t-1})\|^2] = (1-\beta_{1,t})^2 \mathbb{E}[\|\mathbf{g}_t - \nabla f(\boldsymbol{\theta}_{t-1})\|^2] + \beta_{1,t}^2 \mathbb{E}[\|\mathbf{m}_{t-1} - \nabla f(\boldsymbol{\theta}_{t-1})\|^2].$$

Using the bounded variance property, it follows that

$$\mathbb{E}[\|\mathbf{m}_t - \nabla f(\boldsymbol{\theta}_{t-1})\|^2] \leq (1-\beta_{1,t})^2 \sigma^2 + \beta_{1,t}^2 \mathbb{E}[\|\mathbf{m}_{t-1} - \nabla f(\boldsymbol{\theta}_{t-1})\|^2].$$

According to Young's inequality, $\|a+b\|^2 \leq (1+c)\|a\|^2 + (1+\frac{1}{c})\|b\|^2$, where $c = \frac{1-\beta_{1,t}}{\beta_{1,t}}$, we have

$$\mathbb{E}[\|\mathbf{m}_{t-1} - \nabla f(\boldsymbol{\theta}_{t-1})\|^2] \leq \frac{1}{\beta_{1,t}} \mathbb{E}[\|\mathbf{m}_{t-1} - \nabla f(\boldsymbol{\theta}_{t-2})\|^2] + \frac{1}{1-\beta_{1,t}} \mathbb{E}[\|\nabla f(\boldsymbol{\theta}_{t-1}) - \nabla f(\boldsymbol{\theta}_{t-2})\|^2].$$

Furthermore, by L-smoothness,

$$\|\nabla f(\boldsymbol{\theta}_{t-1}) - \nabla f(\boldsymbol{\theta}_{t-2})\|^2 \leq L^2 \|\boldsymbol{\theta}_{t-1} - \boldsymbol{\theta}_{t-2}\|^2 = \eta_{t-1}^2 L^2 \left\| \frac{\mathbf{m}_{t-1}}{\mathbf{v}_{t-1}} \right\|^2 \leq \frac{\eta_{t-1}^2 L^2 d}{\beta_2}.$$

Substituting this back results in

$$\mathbb{E}[\|\mathbf{m}_t - \nabla f(\boldsymbol{\theta}_{t-1})\|^2] \leq \beta_{1,t} \mathbb{E}[\|\mathbf{m}_{t-1} - \nabla f(\boldsymbol{\theta}_{t-2})\|^2] + \frac{\beta_{1,t}^2 \eta_{t-1}^2 L^2 d}{(1-\beta_{1,t})\beta_2} + (1-\beta_{1,t})^2 \sigma^2.$$

Denote $e_t \triangleq \mathbb{E}[\|\mathbf{m}_t - \nabla f(\boldsymbol{\theta}_{t-1})\|^2]$, and unroll the recurrence using product operators, we have

$$e_t \leq \prod_{k=1}^{t} \beta_{1,k} \mathbb{E}[\|\mathbf{m}_0 - \nabla f(\boldsymbol{\theta}_0)\|^2] + \sum_{k=1}^{t} \prod_{j=k+1}^{t} \beta_{1,j} \left[\frac{\beta_{1,k}^2 \eta_{k-1}^2 L^2 d}{(1-\beta_{1,k})\beta_2} + (1-\beta_{1,k})^2 \sigma^2 \right].$$

Without loss of generality, we assume $1 - \beta_{1,k} = 1/k^\tau$, $\eta_k = \eta/k^\delta$, and $0 < \tau, \delta < 1$. Since $\mathbf{m}_0 = 0$, $\eta_{k-1}/\eta_k \leq 2$, and $\nabla f(\boldsymbol{\theta}_0)$ is bounded, it yields

$$e_t \leq \prod_{k=1}^{t} \beta_{1,k} d G_\infty + \sum_{k=1}^{t} \prod_{j=k+1}^{t} \beta_{1,j} \left(\frac{4L^2 \eta^2 d}{\beta_2 k^{2\delta-\tau}} + \frac{\sigma^2}{k^{2\tau}} \right).$$

We wish to evaluate the sequence $I_p(t)$ defined by

$$I_p(t) = \sum_{k=1}^{t} \left(\prod_{j=k+1}^{t} \beta_{1,j} \right) k^{-p},$$

where $\beta_{1,j} = 1 - j^{-\tau}$ with $\tau \in (0,1)$ and $p = 2\tau, 2\delta - \tau > 0$. Actually, for $\tau \in (0,1)$ and all $1 \leq k \leq t$, by using $1 - x \leq e^{-x}$, we immediately have

$$\prod_{j=k+1}^{t} (1 - j^{-\tau}) \leq \exp\left(- \sum_{j=k+1}^{t} j^{-\tau} \right) \leq \exp\left(- \int_{k+1}^{t+1} x^{-\tau} dx \right) \leq \exp\left(1 - \frac{t^{1-\tau} - k^{1-\tau}}{1-\tau} \right).$$

As a result, this implies that $\prod_{k=1}^{t} \beta_{1,k} \leq \exp\left(1 - \frac{t^{1-\tau}-1}{1-\tau}\right)$ vanishes exponentially fast and can be safely ignored without sacrificing the convergence rate. And $I_p(t)$ is upper-bounded by

$$I_p(t) \leq \exp\left(1 - \frac{t^{1-\tau}}{1-\tau}\right) \sum_{k=1}^{t} k^{-p} \exp\left(\frac{k^{1-\tau}}{1-\tau}\right).$$

Let $h(x) = x^{-p} \exp\left(\frac{x^{1-\tau}}{1-\tau}\right)$, and take logarithmic derivative on both sides, we find

$$h'(x) = h(x) \left(x^{-\tau} - \frac{p}{x} \right).$$

This suggests that $h(x)$ is strictly increasing for $x > p^{1/(1-\tau)}$. Consequently,

$$\sum_{k=1}^{t} h(k) \leq C_0 + \int_{x_0}^{t} x^{-p} \exp\left(\frac{x^{1-\tau}}{1-\tau}\right) \mathrm{d}x$$

where C_0 is a constant handling initial terms up to $x_0 = \lceil p^{1/(1-\tau)} \rceil$.

Define $J_p(t) = \int_{x_0}^{t} x^{-p} \exp\left(\frac{x^{1-\tau}}{1-\tau}\right) \mathrm{d}x$. Denote $u = \frac{x^{1-\tau}}{1-\tau}$. Then, $x = ((1-\tau)u)^{\frac{1}{1-\tau}}$ and $\mathrm{d}x = x^\tau \mathrm{d}u = ((1-\tau)u)^{\frac{\tau}{1-\tau}} \mathrm{d}u$. The integral now transforms to

$$J_p(t) = \int_{u_0}^{u(t)} e^u \cdot ((1-\tau)u)^{\frac{\tau-p}{1-\tau}} \mathrm{d}u = (1-\tau)^{\frac{\tau-p}{1-\tau}} \int_{u_0}^{u(t)} e^u u^{-\gamma} \mathrm{d}u,$$

where $\gamma = \frac{p-\tau}{1-\tau}$ and $u(t) = \frac{t^{1-\tau}}{1-\tau}$. Applying integration by parts with $v = u^{-\gamma}$ and $\mathrm{d}w = e^u \mathrm{d}u$,

$$\int_{u_0}^{u(t)} e^u u^{-\gamma} \mathrm{d}u = \left[e^u u^{-\gamma} \right]_{u_0}^{u(t)} + \gamma \int_{u_0}^{u(t)} e^u u^{-(\gamma+1)} \mathrm{d}u.$$

Choose $x_0 = \max\left(\lceil(2\gamma(1-\tau))^{1/(1-\tau)}\rceil, \lceil p^{1/(1-\tau)}\rceil\right)$, it comes that $u \geq u_0 \geq 2\gamma$, and

$$\gamma \int_{u_0}^{u(t)} e^u u^{-(\gamma+1)} \mathrm{d}u \leq \frac{\gamma}{u_0} \int_{u_0}^{u(t)} e^u u^{-\gamma} \mathrm{d}u \leq \frac{1}{2} \int_{u_0}^{u(t)} e^u u^{-\gamma} \mathrm{d}u.$$

Subtracting this from the left-hand side yields

$$\frac{1}{2} \int_{u_0}^{u(t)} e^u u^{-\gamma} \mathrm{d}u \leq e^{u(t)}(u(t))^{-\gamma} - e^{u_0} u_0^{-\gamma} \implies \int_{u_0}^{u(t)} e^u u^{-\gamma} \mathrm{d}u \leq 2e^{u(t)}(u(t))^{-\gamma}.$$

Plugging $u(t) = \frac{t^{1-\tau}}{1-\tau}$ back,

$$J_p(t) \leq 2(1-\tau)^{\frac{\tau-p}{1-\tau}} \left(\frac{t^{1-\tau}}{1-\tau}\right)^{-\frac{p-\tau}{1-\tau}} \exp\left(\frac{t^{1-\tau}}{1-\tau}\right) = 2\, t^{\tau-p} \exp\left(\frac{t^{1-\tau}}{1-\tau}\right).$$

Finally, multiply $J_p(t)$ back by the outer exponential contraction factor $\exp\left(-\frac{t^{1-\tau}}{1-\tau}\right)$,

$$I_p(t) \leq \exp\left(1 - \frac{t^{1-\tau}}{1-\tau}\right)\left(C_0 + 2\, t^{\tau-p} \exp\left(\frac{t^{1-\tau}}{1-\tau}\right)\right) \leq C_0 \exp\left(1 - \frac{t^{1-\tau}}{1-\tau}\right) + 2\, t^{\tau-p}.$$

Since $\exp\left(-\frac{t^{1-\tau}}{1-\tau}\right) = o(t^{\tau-p})$ as $t \to \infty$, it holds that

$$I_p(t) \leq \mathcal{O}(2\, t^{\tau-p}).$$

Therefore, we have

$$e_t \leq \mathcal{O}\left(\frac{8L^2\eta^2 d}{\beta_2} t^{2\tau-2\delta} + \sigma^2 t^{-\tau}\right),$$

and

$$-\left\langle \nabla f(\boldsymbol{\theta}_{t-1}), \frac{\mathbf{m}_t}{\mathbf{v}_t} \right\rangle \leq -\frac{1}{2V_\infty} \|\nabla f(\boldsymbol{\theta}_{t-1})\|_2^2 + \mathcal{O}\left(\frac{4L^2\eta^2 d}{\beta_2} t^{2\tau-2\delta+\kappa} + \frac{\sigma^2}{2} t^{-\tau+\kappa} + \frac{d}{2\beta_2} t^{-\kappa}\right).$$

Consequently,

$$\begin{aligned} &\frac{\sum_{t=1}^T \eta_t \mathbb{E}\left[\|\nabla f(\boldsymbol{\theta}_t)\|_2^2\right]}{2V_\infty} \\ &\leq [f(\boldsymbol{\theta}_0) - f(\boldsymbol{\theta}_T)] \\ &+ \mathcal{O}\left(\sum_{t=1}^T \left[\frac{4L^2\eta^2 d}{\beta_2} t^{2\tau-2\delta+\kappa} + \frac{\sigma^2}{2} t^{-\tau+\kappa} + \frac{d}{2\beta_2} t^{-\kappa} + \frac{Ld}{2\beta_2} t^{-\delta}\right] t^{-\delta}\eta\right) \end{aligned} \tag{4}$$

and

$$\min_{t=1,\cdots,T} \left\{\mathbb{E}\left[\|\nabla f(\boldsymbol{\theta}_t)\|^2\right]\right\} \leq \mathcal{O}\left(\frac{2V_\infty [f(\boldsymbol{\theta}_0) - f(\boldsymbol{\theta}_T)]}{T^{1-\delta}} + \frac{8L^2\eta^2 d V_\infty}{\beta_2 T^{2\delta-2\tau-\kappa}} + \frac{\sigma^2 V_\infty}{T^{\tau-\kappa}} + \frac{dV_\infty}{\beta_2 T^\kappa} + \frac{LdV_\infty}{\beta_2 T^\delta}\right).$$

Let $m = \min(1-\delta, 2\delta-2\tau-\kappa, \tau-\kappa, \kappa, \delta)$, we want to maximize m. If m is feasible, then necessarily

$$\begin{aligned} 1-\delta &\geq m, \\ 2\delta-2\tau-\kappa &\geq m, \\ \tau-\kappa &\geq m, \\ \kappa &\geq m, \\ \delta &\geq m. \end{aligned}$$

Note that

$$2*(1-\delta) + 1*(2\delta-2\tau-\kappa) + 2*(\tau-\kappa) + 3*\kappa = 2 \geq 8m.$$

Therefore, $m \leq 1/4$ and this maximum value is achievable by letting $\delta = 3/4$, $\tau = 1/2$, and $\kappa = 1/4$. Finally, substitute these values into Equation (4) and we note that $\kappa + \delta = 1$, which incurs a $\ln T$ term in the numerator. As a result, after absorbing the constant V_∞ that depends only G, β_2, ϵ, we have

$$\min_{t=1,\cdots,T} \left\{\mathbb{E}\left[\|\nabla f(\boldsymbol{\theta}_t)\|^2\right]\right\} \leq \mathcal{O}\left(\frac{2[f(\boldsymbol{\theta}_0) - f(\boldsymbol{\theta}_T)]}{T^{1/4}} + \frac{8L^2\eta^2 d}{\beta_2 T^{1/4}} + \frac{\sigma^2}{T^{1/4}} + \frac{d\ln T}{\beta_2 T^{1/4}} + \frac{Ld}{\beta_2 T^{3/4}}\right).$$

□