Beyond the Ideal: Analyzing the Inexact Muon Update

Egor Shulgin **Sultan Alrashed** **Francesco Orabona** **Peter Richtárik**
King Abdullah University of Science and Technology (KAUST)
KAUST Center of Excellence for Generative AI
Thuwal, Saudi Arabia
shulgin.yegor@gmail.com, sultan.m.rashed@gmail.com
francesco@orabona.com, peter.richtarik@kaust.edu.sa

Abstract

The Muon optimizer has rapidly emerged as a powerful, geometry-aware alternative to AdamW, demonstrating strong performance in large-scale training of neural networks. However, a critical theory-practice disconnect exists: Muon's efficiency relies on fast, approximate orthogonalization, while most theoretical analyses study idealized exact-SVD updates. This work moves beyond the ideal by providing a general analysis of the *inexact* orthogonalized update at Muon's core. We develop our analysis within the general framework of Linear Minimization Oracle (LMO)-based optimization, introducing a realistic additive error model to capture the inexactness of practical approximation schemes. Our analysis yields explicit bounds that quantify performance degradation as a function of the LMO inexactness/error, δ. We reveal a fundamental coupling between this inexactness and the optimal step size and momentum: lower oracle precision requires a smaller step size but larger momentum parameter. These findings elevate the approximation procedure (e.g., the number of `Newton-Schulz` steps) from an implementation detail to a critical parameter that must be *co-tuned* with the learning schedule. NanoGPT experiments directly confirm the predicted coupling, with optimal learning rates clearly shifting as approximation precision changes.

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

1 INTRODUCTION

For over a decade, the landscape of deep learning optimization has been dominated by adaptive first-order methods, with AdamW (Kingma and Ba, 2015; Loshchilov and Hutter, 2019) serving as the de facto standard for training large and complex neural networks. Its robustness and general effectiveness have powered progress across numerous domains. Yet a new class of geometry-aware optimizers has recently emerged, challenging this paradigm. Among them, Muon (Jordan et al., 2024b) has quickly gained prominence as a successor to AdamW. By leveraging matrix structure in neural network parameters, Muon has demonstrated superior performance and scalability, setting new training speed records for models like nanoGPT (Jordan et al., 2024a) and enabling the efficient training of state-of-the-art Large Language Models (LLMs), such as Kimi from Moonshot AI (Bai et al., 2025). Benchmarking studies consistently show that Muon can be significantly more computationally efficient than AdamW (Liu et al., 2025; Shah et al., 2025; Wen et al., 2026).

However, a critical disconnect lies at the heart of Muon's success. The practical efficiency of the optimizer is entirely predicated on its use of fast, approximate orthogonalization methods, most notably the `Newton-Schulz` iteration (Jordan et al., 2024b). This iterative matrix-multiplication-based procedure provides a computationally cheap way to approximate the orthogonal factor of a matrix's polar decomposition, avoiding a full Singular Value Decomposition (SVD) which would be prohibitively expensive (Grishina et al., 2025). Yet, a significant gap exists between this practical implementation and its theoretical understanding. Existing theoretical analyses of Muon (e.g., Li and Hong, 2025; Shen et al., 2025; Chen et al., 2025) have largely studied an *idealized*, computationally intractable version of the algorithm, assuming access to an exact orthogonalization oracle that

computes the perfect SVD-based update. Hence, the existing theory still falls short of fully explaining the practical optimizer.

This work moves beyond the ideal to provide a general theoretical analysis of the *inexact* Muon update. We situate our analysis within the general framework of Linear Minimization Oracle (LMO)-based optimization (Pethick et al., 2025b), also known as the Frank-Wolfe framework (Frank and Wolfe, 1956; Hazan, 2008; Clarkson, 2010; Jaggi, 2013). This perspective frames the core Muon operation as an LMO call over the unit ball with respect to the spectral norm. To account for the realities of practical computation, we introduce a realistic additive error model. This model is directly motivated by the behavior of practical approximation schemes like `Newton-Schulz`, allowing us to capture the inexactness inherent in any efficient implementation. By analyzing the algorithm under this inexact LMO, we bridge the critical gap between theory and practice.

1.1 Contributions

Our main contributions to the theory and practice of LMO-based optimizers are:

• **An analysis of the implemented inexact Muon update.** We analyze the *implemented* Muon update through an additive direction-space inexactness model (Assumption 1) tailored to practical polar-factor approximation schemes such as `Newton-Schulz` and `PolarExpress`.

• **Uncovering a fundamental hyperparameter coupling.** Our analysis reveals that the LMO inexactness, δ, is a critical parameter that alters the optimization dynamics. We derive explicit formulas for the step size ($\gamma^* \propto (1+\delta)^{-1/4}$) and momentum ($\alpha^* \propto \sqrt{1+\delta}$) in the stochastic setting (Corollary 3), uncovering a crucial coupling: a less precise LMO (larger δ) requires a *smaller* learning rate but a *larger* momentum parameter. This elevates the approximation quality from an implementation detail to a core hyperparameter.

• **A general convergence framework.** We establish convergence rates for both deterministic and stochastic settings. In the Appendix, we further extend the *deterministic* analysis to the (L^0, L^1)-smoothness model (Zhang et al., 2020) and to a layer-wise setting (Riabinin et al., 2025). Setting the inexactness to zero recovers the corresponding exact-LMO rates in the LMO framework.

• **Empirical validation.** We validate our key theoretical predictions with experiments on CIFAR-10 and NanoGPT. We confirm that performance degrades as LMO precision decreases and, crucially, our experiments with NanoGPT (Figure 1b) provide clear empirical evidence for the predicted hyperparameter coupling, showing that the optimal learning rate shifts to a lower value when a less precise LMO is used.

1.2 Related work

We survey the key developments toward a theoretical understanding of Muon.

The conceptual basis for Muon's orthogonalized update was introduced by Jordan et al. (2024b) and further motivated by Bernstein and Newhouse (2024), who connected orthogonalized matrix updates to steepest-descent geometry under the spectral norm. The first formal convergence analyses for Muon sought to connect it to existing theoretical frameworks. In particular, three different groups independently provided convergence guarantees for Muon, with varying degrees of generality: Li and Hong (2025), Pethick et al. (2025b), and Kovalev (2025).

Li and Hong (2025) provided an initial convergence guarantee by viewing the Muon update as a matrix-based generalization of normalized SGD with momentum, building upon the analysis of Cutkosky and Mehta (2020).

Instead, Pethick et al. (2025b) and Kovalev (2025) proposed two powerful unifying frameworks, providing a more general perspective. Pethick et al. (2025b) introduced the Scion framework, which situates Muon within the broader class of methods based on the Linear Minimization Oracle (LMO), a core component of the Frank-Wolfe algorithm (Frank and Wolfe, 1956). Kovalev (2025) developed a non-Euclidean trust-region interpretation, also for arbitrary norms. Both frameworks successfully recover the idealized Muon update as a special case when the spectral norm is chosen, and provided the first general convergence guarantees for this class of optimizers.

Beyond the Muon literature, inexact LMOs / inexact Frank–Wolfe or generalized conditional gradient methods have been studied extensively in convex and constrained optimization, under several oracle models including multiplicative and additive inaccuracies; see, e.g., Lacoste-Julien et al. (2013); Locatello et al. (2017a,b); Yurtsever et al. (2018); Silveti-Falls et al. (2021). Our setting differs in three important ways: we study unconstrained, non-convex, momentum-based Muon/Scion-style updates; we focus on the matrix-orthogonalization regime relevant to Muon; and we use an additive model motivated by practical polar-factor solvers such as `Newton-Schulz` and `PolarExpress`, without assuming that the approximate direction remains feasible.

An idealized trust-region method, called the ball-proximal ("broximal") point method (BPM) was con-

currently developed by Gruntkowska et al. (2025a). BPM has remarkable theoretical guarantees for convex optimization—it converges in finitely many steps, and has linear rate independent of the condition number, with the geometric factor improving in each iteration—without requiring differentiability, finite-valuedness, or strong convexity. Moreover, the method provably converges to the global minimizer for a certain class of non-convex problems. A non-Euclidean variant was recently proposed and analyzed by Gruntkowska and Richtárik (2025), with the non-Euclidean norm playing the role of a hyper-parameter performing a form of geometric preconditioning. Muon can be seen as an approximate version of non-Euclidean BPM for a specific norm choice, where the broximal operator is applied to a stochastic linear approximation of the loss instead of the original loss, with momentum added to the mix as a means of handling stochastic noise.

Subsequent work has focused on making these general frameworks more reflective of practical deep learning scenarios. The Gluon framework of Riabinin et al. (2025) extended the analysis to a more realistic layer-wise setting and introduced a generalized non-Euclidean smoothness model to better capture the heterogeneous structure of neural networks. Concurrently, Pethick et al. (2025c) introduced a clipped Scion variant that provides guarantees under a similar generalized smoothness condition. Further, Gruntkowska et al. (2025b) have shown that it may be theoretically suboptimal to update *all* the layers of a neural network in each iteration, and as a remedy, proposed and analyzed (also under generalized smoothness) the Drop-Muon method, showing wall-clock improvements on toy networks.

An extension of Muon to the distributed setting, with support for communication compression, error-feedback and generalized smoothness, was developed by Gruntkowska et al. (2026), who proposed the EF21-Muon method. The MuonBP method of Khaled et al. (2025) proposes to apply orthogonalization independently to matrix shards on each device, while periodically performing full orthogonalization to maintain training stability at scale.

Despite all this theoretical understanding, a substantial part of the literature still analyzes an idealized algorithm that assumes access to an exact, error-free LMO. This is a significant gap, as the practical efficiency and success of Muon are entirely predicated on the use of fast but approximate solvers for the orthogonalized update. The concept of spectral descent and orthogonalized updates has historical roots in deep learning (Carlson et al., 2015a, 2016, 2015b; Tuddenham et al., 2022). Our work addresses the remaining broader disconnect between theory and practice through a solver-agnostic inexact-LMO analysis.[1]

2 FROM IDEALIZED THEORY TO PRACTICAL IMPLEMENTATION

We begin by formalizing the class of algorithms our analysis covers, starting from the Muon optimizer.

The idealized update Muon (Jordan et al., 2024b) is matrix-aware, leveraging the geometric structure of two-dimensional weight parameters ($W \in \mathbb{R}^{n\times m}$). Instead of element-wise scaling as in AdamW, Muon applies momentum and then performs an orthogonalization step on the resulting update matrix. The idealized update has the form

$$M^k = (1-\alpha)M^{k-1} + \alpha G^k, \quad D^k = T(M^k),$$
$$X^{k+1} = X^k - \gamma^k D^k,$$

where G^k is a stochastic gradient, $\gamma^k > 0$ is the step size, M^k is the momentum matrix, $\alpha > 0$ is the momentum parameter, and $T(\cdot)$ represents the projection onto the set of orthogonal matrices.

As shown by Bernstein and Newhouse (2024), this update is equivalent to performing steepest descent with respect to the spectral norm geometry, $\|\cdot\| := \|\cdot\|_{\mathrm{sp}}$. Given a gradient matrix G, the steepest descent direction under this norm is equal to

$$\operatorname{argmin}\ \{\langle G, D\rangle\ :\ \|D\|_{\mathrm{sp}} \le 1\}. \tag{1}$$

The solution to (1) is the orthogonal polar factor of the negative gradient, $D = \operatorname{polar}(-G)$. If the SVD of the gradient is $G = USV^\top$, then the solution is $D = -UV^\top$. This idealized update direction forms the theoretical basis of the Muon optimizer. In its practical implementation, the computationally expensive polar decomposition is approximated by the efficient, SVD-free `Newton-Schulz` iteration. This practical algorithm demonstrated remarkable empirical success, but was introduced without a formal convergence analysis.

Subsequently, Pethick et al. (2025b) (using the *Linear Minimization Oracle* (LMO) framework) and Kovalev (2025) (via a non-Euclidean trust-region framework) provided the first meaningful convergence guarantees for this type of update. These frameworks consider a general update of the form[2] $x^{k+1} = x^k + \gamma^k d^k$, where the direction d^k is the solution to the LMO

$$d^k := \operatorname{argmin}\ \left\{\left\langle m^k, d\right\rangle\ :\ \|d\| \le 1\right\}, \tag{2}$$

[1] Concurrent with our work, Kim and Oh (2026) study the original Muon algorithm with finitely many Newton–Schulz steps. We defer a more detailed comparison to Appendix A.1.

[2] From now on, we drop the upper-case matrix notation in favor of simpler, lower-case vector-space notation.

where m^k is the momentum term, $\langle \cdot, \cdot \rangle$ refers to an inner product (trace inner product in matrix spaces), and $\|\cdot\|$ refers to an arbitrary, possibly non-Euclidean norm. These analyses recover the idealized Muon update when $\|\cdot\|$ is chosen as the spectral norm.

The practical imperative of approximation While these frameworks provide invaluable insight, they analyze an idealized algorithm that is never run in practice. For many norms, such as the ℓ_∞ norm in SignSGD, the LMO in (2) can be computed exactly and with minimal overhead. For the spectral norm, however, the exact LMO solution requires a full, prohibitively expensive, SVD.

This computational bottleneck makes the practical implementations of Muon entirely dependent on *approximate updates*. The original optimizer employs 5 steps of the `Newton-Schulz` iteration (Higham, 2008), a matrix polynomial-based method that efficiently approximates the polar factor using only fast matrix-matrix multiplications (Jordan et al., 2024b). The importance of this approximation is underscored by an active line of research into developing superior iterative schemes (Cesista et al., 2025), such as `PolarExpress` (Amsel et al., 2026) and `CANS` (Grishina et al., 2025), which offer better error guarantees or faster convergence. This highlights a crucial fact: the algorithm achieving state-of-the-art results is not the idealized one, but one of its many possible inexact instantiations. Much of the existing theory still focuses on the idealized case, assuming access to an error-free LMO.

Modeling inexactness To bridge this theory-practice divide, we must first establish a realistic model for the error produced by the LMO approximation. We replace the exact direction d^k in the update rule with an inexact direction $\hat{d}^k$, which is assumed to satisfy the following assumption.

Assumption 1 (**Inexact LMO**). *Let d^k be the exact solution to (2). The inexact solution $\hat{d}^k$ is assumed to satisfy an additive error bound for some $\delta_k \geq 0$:*

$$\|\hat{d}^k - d^k\| \leq \delta_k. \tag{3}$$

This assumption is not arbitrary; it is directly motivated by the convergence guarantees of the iterative methods used in practice. Algorithms like `Newton-Schulz` and `PolarExpress` produce an approximation whose error decreases with the number of iterations performed. Specifically, Amsel et al. (2026) prove that `PolarExpress` satisfies an error bound of the form $\|\hat{d}^k - d^k\| \leq C|1 - l|^p$, where $l < 1$ and p is determined by the number of iterations (see Appendix A for details). Since these methods are always run for a small, fixed number of steps in practice (e.g., five steps in the standard Muon implementation (Jordan et al., 2024b)), their final error is bounded by a constant. Our parameter δ_k models this error, allowing our analysis to capture the behavior of the implemented algorithm. A key feature of our analysis is that we do not assume the output $\hat{d}^k$ to be feasible, i.e., we do not assume $\|\hat{d}^k\| \leq 1$, which is a realistic property of these approximation schemes.

3 MAIN THEORETICAL RESULTS

Having established the practical importance of the inexact LMO, we now develop a theoretical framework for analyzing its impact on convergence. Our analysis is general and holds for any norm $\|\cdot\|$. We consider the unconstrained optimization problem

$$\min \ \{f(x) \ : \ x \in \mathcal{X}\}, \tag{4}$$

where $(\mathcal{X}, \|\cdot\|)$ is a normed space. We will use the following standard assumption on the objective function.

Assumption 2. *The objective function $f : \mathcal{X} \to \mathbb{R}$ is continuously differentiable and its gradient ∇f is Lipschitz continuous with constant $L \geq 0$ with respect to the norm $\|\cdot\|$. This means for any $x, y \in \mathcal{X}$ we have*

$$\|\nabla f(x) - \nabla f(y)\|_\star \leq L\|x - y\|,$$

where $\|\cdot\|_\star$ is the dual norm of $\|\cdot\|$. Furthermore, we assume that f is bounded below by $f^* > -\infty$.

We begin with the deterministic setting to build intuition, before moving to the full stochastic analysis with momentum.

3.1 Deterministic case

In the deterministic setting, we analyze the update

$$x^{k+1} = x^k + \gamma^k \hat{d}^k, \tag{5}$$

where $\gamma^k > 0$ is the step size and $\hat{d}^k$ is the output of an inexact LMO for the gradient $g^k := \nabla f(x^k)$, i.e.,

$$\hat{d}^k \approx \operatorname{argmin} \ \left\{\langle g^k, d\rangle \ : \ \|d\| \leq 1\right\}.$$

The inexact direction $\hat{d}^k$ is assumed to satisfy Assumption 1 with some inexactness level $\delta_k \geq 0$. We now state our first convergence result for this method.

Theorem 1 (General Result). *Let Assumption 2 hold. Let the sequence $\{x^k\}_{k=0}^{K-1}$ be generated by the update rule (5) with step sizes $\gamma^k > 0$. Assume the inexact LMO satisfies Assumption 1 with $\delta_k < 1$ for all k and let $\Delta^0 := f(x^0) - f^*$. Then, after K iterations,*

$$\min_{0 \leq k < K} \ \left\|\nabla f(x^k)\right\|_\star \leq \frac{\Delta^0 + \frac{L}{2}\sum_{k=0}^{K-1}(\gamma^k)^2(1+\delta_k)^2}{\sum_{k=0}^{K-1}\gamma^k(1-\delta_k)}.$$

Theorem 1 provides a general convergence guarantee that explicitly characterizes how the interplay of step sizes $\{\gamma^k\}$ and inexactness levels $\{\delta_k\}$ affects convergence. The bound reveals that inexactness degrades performance in two ways: the numerator is amplified by a $(1+\delta_k)^2$ factor due to the potential infeasibility of the update, while the denominator, representing total progress, is diminished by a $(1-\delta_k)$ factor. Condition $\delta_k < 1$ is mathematically necessary to ensure that the denominator is positive, which guarantees that the algorithm makes progress on average. As shown in the proof (see Appendix B), this condition is required for the per-iteration descent property and is satisfied by practical approximation schemes like `PolarExpress` (Amsel et al., 2026). To gain clearer, quantitative insights into these trade-offs, we next analyze the important special case of constant parameters.

Corollary 1 (Constant Parameters)**.** *Under the conditions of Theorem 1, if the step size is constant, $\gamma^k = \gamma > 0$, and the LMO error is constant, $\delta_k = \delta < 1$, for all k, then after K iterations we have*

$$\frac{1}{K}\sum_{k=0}^{K-1}\left\|\nabla f(x^k)\right\|_\star \le \frac{\Delta^0}{K\gamma(1-\delta)} + \frac{L\gamma(1+\delta)^2}{2(1-\delta)}.$$

Corollary 1 simplifies the general bound, making the trade-offs more apparent. The bound consists of two terms that exhibit the classical trade-off in the choice of step size γ: a larger γ reduces the first term but increases the second. This structure implies that an optimal convergence rate is achieved when the step size is of the order $\mathcal{O}(1/\sqrt{K})$, which balances these two terms.

The inexactness level δ degrades both terms in the bound. They are each amplified by a factor of $1/(1-\delta)$, which arises from the reduced quality of the descent direction. The second term is further penalized by a $(1+\delta)^2$ factor, which quantifies the cost of potential infeasibility of the update step.

Our analysis is general, holding for any norm in a framework akin to that of Pethick et al. (2025b) and Kovalev (2025). For the specific choice of the spectral norm, $\|\cdot\| = \|\cdot\|_\star$, this result provides a convergence guarantee for the practical, inexact Muon update. Notably, if we set the inexactness to zero ($\delta = 0$), our bound recovers the standard $\mathcal{O}(1/\sqrt{K})$ rate for the exact LMO method. Our framework also accommodates more complex step-size rules; in Appendix B.2, we analyze an adaptive step-size policy and show that it achieves the same optimal rate. To make the trade-off in the constant step size explicit, we now derive the optimal choice of γ that minimizes this bound.

By minimizing the right-hand side of the bound in Corollary 1 with respect to γ, we obtain the optimal constant step size and the corresponding best rate.

Corollary 2 (Optimal Parameters)**.** *Under the conditions of Corollary 1, by choosing the optimal constant step size $\gamma^* = \frac{1}{1+\delta}\sqrt{\frac{2\Delta^0}{KL}}$, the average gradient norm is bounded as*

$$\frac{1}{K}\sum_{k=0}^{K-1}\left\|\nabla f(x^k)\right\|_\star \le \frac{1+\delta}{1-\delta}\sqrt{\frac{2\Delta^0 L}{K}}.$$

Corollary 2 makes the theoretical trade-offs concrete, revealing a direct coupling between the LMO precision and the optimal strategy. *The optimal step size, $\gamma^* \propto 1/(1+\delta)$, must decrease as the oracle becomes less precise.* This has a direct practical implication for optimizers like Muon: using a less accurate approximation of the orthogonalized update (e.g., by reducing the number of `Newton-Schulz` iterations) might require a corresponding decrease in the step size to maintain optimal performance. This adaptation, however, does not fully mitigate the error, as the best achievable rate is degraded by a factor of $\frac{1+\delta}{1-\delta}$. This factor quantifies the dual cost of the potential update infeasibility (from the $1+\delta$ term) and reduced descent quality (from the $1-\delta$ term). The explicit dependence of the rate on this degradation factor also provides a theoretical justification for the empirical benefits of more advanced approximation schemes. In fact, methods such as `PolarExpress` (Amsel et al., 2026) and `CANS` (Grishina et al., 2025) are designed to achieve a smaller approximation error δ more efficiently, which our analysis shows directly translates to an improved convergence rate for the overall optimization.

Our analysis is a tight generalization of prior work; setting $\delta = 0$ recovers the exact rate for the idealized LMO method (Kovalev, 2025; Pethick et al., 2025b). Importantly, while the constant is degraded by the inexactness, the $\mathcal{O}(1/\sqrt{K})$ rate implies an iteration complexity of $\mathcal{O}(1/\varepsilon^2)$ to find an ε-stationary point. This matches the optimal complexity for first-order methods on smooth non-convex problems (Carmon et al., 2020). Our result thus establishes that the inexact LMO method remains optimal, while characterizing the degradation as a function of the oracle's error.

3.2 Stochastic case

We now extend our analysis to the more practical setting where the optimizer has access only to a stochastic oracle for the gradient and incorporates momentum.

For this setting, we require two additional standard assumptions. First, we assume access to an unbiased stochastic first-order oracle with bounded variance.

Algorithm 1 Inexact Generalized Muon

Input: Initial point x^0, momentum m^0
for $k = 0, 1, \ldots, K-1$ **do**
  Compute stochastic gradient g^k
  Update momentum: $m^{k+1} = (1-\alpha_k)m^k + \alpha_k g^k$
  Compute inexact LMO with m^{k+1}:

$$\hat{d}^k \approx \operatorname*{argmin}_{\|d\| \le 1} \left\langle m^{k+1}, d \right\rangle$$

  Update parameters: $x^{k+1} = x^k + \gamma^k \hat{d}^k$
end for

Assumption 3. *The oracle returns a stochastic gradient $g^k = \nabla f(x^k) + \xi^k$ for a random variable ξ^k, such that it is an unbiased estimator of the true gradient, $\mathbb{E}\left[g^k \mid x^k\right] = \nabla f(x^k)$, and has a uniformly bounded variance, $\mathbb{E}\left[\|g^k - \nabla f(x^k)\|_2^2 \mid x^k\right] \le \sigma^2$, for $\sigma^2 \ge 0$.*

Second, since our analysis is for a general norm $\|\cdot\|$ but the variance is typically assumed to be bounded in the Euclidean norm $\|\cdot\|_2$, we assume a norm compatibility condition that is always true in finite-dimensional spaces. So, we denote by $\rho > 0$ the constant such that for all $v \in \mathcal{X}$, we have $\|v\|_\star \le \rho \|v\|_2$.

The method we analyze is a general momentum-based algorithm with an inexact LMO, see Algorithm 1.

To analyze this algorithm, our proof proceeds by establishing a per-iteration descent guarantee and then bounding the error of the momentum term. While the full proofs are deferred to Appendix B.3, we present the key lemma for the momentum error here, as it reveals the direct impact of the inexact LMO.

Lemma 1. *Let Assumptions 1, 2, 3 hold. For Algorithm 1, the expected momentum error is bounded by*

$$\mathbb{E}\|m^{k+1} - \nabla f(x^k)\|_\star \le (1-\alpha_k)\mathbb{E}[\|m^k - \nabla f(x^{k-1})\|_\star] + \frac{\rho\sigma\sqrt{\alpha_k}}{\sqrt{2-\alpha_k}} + L\gamma^k(1+\delta_k).$$

Commentary on the Proof. This lemma bounds the expected deviation of the momentum from the true gradient. The critical difference in our analysis, compared to prior work on exact LMOs, arises when bounding the "gradient drift" term, which depends on the step length $\|\nabla f(x^{k-1}) - \nabla f(x^k)\|_\star \le L\|x^{k-1} - x^k\|$. An exact LMO guarantees a feasible direction $\|d^{k-1}\| \le 1$, yielding a step length of exactly γ^{k-1}. In our analysis, the potential infeasibility of the inexact direction, $\|\hat{d}^{k-1}\| \le 1 + \delta_{k-1}$, leads to a looser step length bound of $\gamma^{k-1}(1+\delta_{k-1})$. This modification introduces the crucial $(1+\delta_k)$ factor into the final term of the lemma, quantifying the cost of potential infeasibility inherent to any practical LMO approximation. The full proof is provided in Appendix B.3.

Building on this lemma, we now present the main convergence guarantee for Algorithm 1.

Theorem 2. *Let Assumptions 1,2,3 hold, along with the norm compatibility condition. For Algorithm 1 with parameters $\gamma > 0$, $\alpha \in (0,1)$, and $\delta < 1$, after K iterations, the average expected gradient norm is upper bounded by*

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\|\nabla f(x^k)\|_\star \le \frac{1}{1-\delta}\left[\frac{\Delta^0}{K\gamma} + 2\rho\sigma\left(\frac{1}{\alpha K} + \sqrt{\alpha}\right) + L\gamma\left(\frac{7+3\delta}{2} + \frac{2(1+\delta)}{\alpha}\right)\right],$$

where $\Delta^0 := f(x^0) - f^*$.

Theorem 2 provides a convergence guarantee for the practical, inexact LMO-based method with momentum. Compared to the deterministic analysis in Corollary 1, the bound contains additional terms dependent on the noise variance σ^2. With a constant step size, these terms establish convergence that the algorithm will minimize the expected gradient up to a floor governed by the choice of parameters.

Crucially, every term in the bound is amplified by factors involving the inexactness level δ, which confirms the intuition that a more precise LMO (smaller δ) leads to faster convergence. As a sanity check, setting the inexactness to zero ($\delta = 0$) recovers the corresponding convergence guarantee for the idealized, exact LMO method (Kovalev, 2025; Pethick et al., 2025b), demonstrating that our analysis is a strict generalization.

For clarity, we present this result for the practical case of constant parameters, the case of time-varying parameters is deferred to Appendix B.4. To better understand the complex interplay of the constant parameters, we now derive the optimal choices of γ and α that minimize this bound (details in Appendix B.3).

Corollary 3. *Let the conditions of Theorem 2 hold. By choosing γ and α of orders*

$$\gamma \asymp \left(\frac{\Delta^0}{K}\right)^{3/4} \frac{1}{(\sigma^2 L(1+\delta))^{1/4}},$$
$$\alpha \asymp \sqrt{\frac{\Delta^0 L(1+\delta)}{K\sigma^2}},$$

the average expected gradient norm is bounded as

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left\|\nabla f(x^k)\right\|_\star = \mathcal{O}\left(\frac{(L\Delta^0)^{1/4}\sigma^{1/2}(1+\delta)^{1/4}}{K^{1/4}(1-\delta)}\right).$$

Corollary 3 provides the main insights of our stochastic analysis, revealing a fundamental coupling between the LMO precision and the hyperparameter strategy. The best achievable rate is degraded by a factor of $\mathcal{O}\left(\frac{(1+\delta)^{1/4}}{1-\delta}\right)$. This explicit dependence on δ provides a theoretical justification for the empirical benefits of more advanced approximation schemes like `PolarExpress` (Amsel et al., 2026) and `CANS` (Grishina et al., 2025), as any reduction in approximation error directly improves the convergence rate. Interestingly, the degradation from inexactness is less severe than in the deterministic setting (Corollary 2), where the rate is penalized by $\mathcal{O}\left(\frac{1+\delta}{1-\delta}\right)$. This suggests that the presence of stochastic noise, which already necessitates a degree of caution, makes the optimization dynamics less sensitive to the additional instability from the inexact LMO.

This rate is achieved when the hyperparameters are adapted to the LMO's precision. The step size, $\gamma^* \propto 1/(1+\delta)^{1/4}$, must **decrease** as the oracle becomes less precise. In contrast, the momentum parameter, $\alpha^* \propto \sqrt{1+\delta}$, must **increase**. This prescribes a clear strategy: a less accurate LMO (larger δ) requires more caution (smaller step size) but also greater agility (shorter momentum memory) to adapt to less reliable update directions.

Finally, the $\mathcal{O}(1/K^{1/4})$ rate implies an iteration complexity of $\mathcal{O}(1/\varepsilon^4)$ to find an ε-stationary point. This matches the established lower bounds for first-order stochastic methods on non-convex problems under our assumptions (Ghadimi and Lan, 2013), and this complexity is known to be unimprovable in general (Arjevani et al., 2023). Our analysis thus establishes that the inexact LMO method remains efficient in a formal sense, while precisely characterizing the degradation as a function of the oracle's error.

3.3 Extensions and generalizations

Beyond standard smoothness. Our analysis can be extended beyond the standard L-smoothness assumption, which can be restrictive for neural networks. In Appendix C.1, we provide a full analysis for the more general non-Euclidean (L^0, L^1)-smoothness model (Zhang et al., 2020; Yu et al., 2025; Riabinin et al., 2025). For this class of functions, we establish an iteration complexity of

$$\mathcal{O}\left(\frac{\Delta^0(1+\delta)^2}{(1-\delta)^2}\left(\frac{L^0}{\varepsilon^2}+\frac{L^1}{\varepsilon}\right)\right).$$

This result recovers the standard $\mathcal{O}(1/\varepsilon^2)$ rate when $L^1 = 0$. More importantly, it allows for an improved rate over gradient descent in regimes where L^0 is small, a phenomenon that has been empirically observed for the training trajectories of nanoGPT on FineWeb (Riabinin et al., 2025). Our analysis shows that the dependence on the inexactness level δ remains consistent, and by setting $\delta = 0$, our results tightly recover the prior guarantees for the exact LMO method on (L^0, L^1)-smooth functions (Vankov et al., 2025).

Layer-wise analysis. In practice, optimizers like Muon and Scion are applied in a layer-wise manner, often with different norms and update rules for different parameter blocks (e.g., spectral norm for weight matrices, ℓ_∞ norm for biases) (Pethick et al., 2025b). This heterogeneity, along with empirical observations of varying smoothness across layers (Riabinin et al., 2025), motivates a more fine-grained analysis. In Appendix C.2, we extend our framework by considering a block-wise structure of the parameter space $x = (x_1, \ldots, x_p)$, where each block x_i is associated with its own norm $\|\cdot\|_{(i)}$, smoothness constant L_i, and inexactness δ_i.

Our analysis in this setting reveals that the impact of LMO inexactness is not uniform across the network. This suggests that a uniform precision level for all layers may be suboptimal and opens the door to a principled strategy for computational savings, where one could allocate fewer resources (e.g., fewer `Newton-Schulz` iterations) to approximate the LMO for layers that are more robust to inexactness. This type of heterogeneity is visible empirically in Figure 5, where attention and MLP matrices exhibit markedly different `PolarExpress` difficulty profiles even at the same training checkpoint.

4 EXPERIMENTS

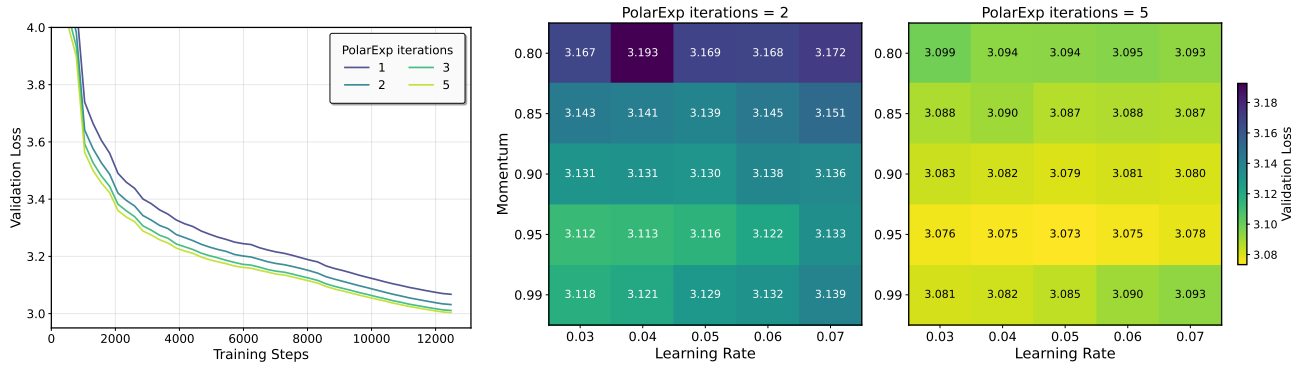
We now present empirical results to validate the key predictions of our theoretical analysis. Our experiments are designed to answer two main questions:

1. How does optimizer performance degrade as the LMO becomes less precise?
2. Do the hyperparameters for the best performance (step size and momentum) shift in response to the LMO inexactness, as predicted by our analysis?

We use the number of iterations in the approximation algorithm (either `Newton-Schulz` or `PolarExpress`) as a practical proxy for the inexactness level δ, where fewer iterations correspond to a higher δ. We conduct experiments on two standard benchmarks: training a nanoGPT model on the FineWeb dataset (Penedo et al., 2024) and a CNN on CIFAR-10 (Krizhevsky, 2009). Full details are provided in Appendix D.1.

4.1 nanoGPT on FineWeb

We train a 124M parameter nanoGPT model with Muon with a batch size of 512,000 using the code-



(a) Validation loss behavior; more iterations (smaller δ) drive faster convergence.

(b) Hyperparameter sweeps for 2 (left) and 5 (right) `PolarExpress` iterations; higher precision widens the stable region.

Figure 1: NanoGPT on FineWeb: convergence (left) and hyperparameter sensitivity (right) as `PolarExpress` (PolarExp) iterations vary.

base by Jordan et al. (2024a). For the LMO approximation, we use the `PolarExpress` algorithm (Amsel et al., 2026).

Validating performance degradation. Our theory predicts that a less precise LMO (higher δ) should lead to degraded convergence performance. Figure 1a confirms this trend. We trained the model for 12,000 steps and observe that increasing the precision of the LMO by using more `PolarExpress` iterations consistently leads to a lower final validation loss. While the most significant gains are seen when moving from one to three iterations, further increases in precision continue to improve performance, though with diminishing returns (see Appendix D.2 for results with up to 8 iterations). This observation directly supports the degradation factor in our convergence bounds.

Validating hyperparameter coupling. A key prediction of our stochastic analysis (Corollary 3) is the coupling between the LMO inexactness δ, the step size γ^*, and the momentum α^*. Figure 1b provides empirical evidence for this theoretical insight. The figure shows the final validation loss after 6,000 training steps across a grid of step sizes and momentum values for two different levels of LMO precision: a highly inexact oracle (2 `PolarExp` iterations, left) and a more precise one (5 `PolarExp` iterations, right).

For the **highly inexact** case (left panel), the region of best performance (lowest loss) is concentrated at a **low step size** (around 0.03). When the LMO is made **more precise** (right panel), the optimal region not only shifts to a **higher step size** (around 0.05) but also becomes broader, indicating greater stability across different hyperparameter choices. This empirical result aligns with our theory: Corollary 3 predicts that as inexactness δ increases (fewer iterations), the optimal step size should decrease.

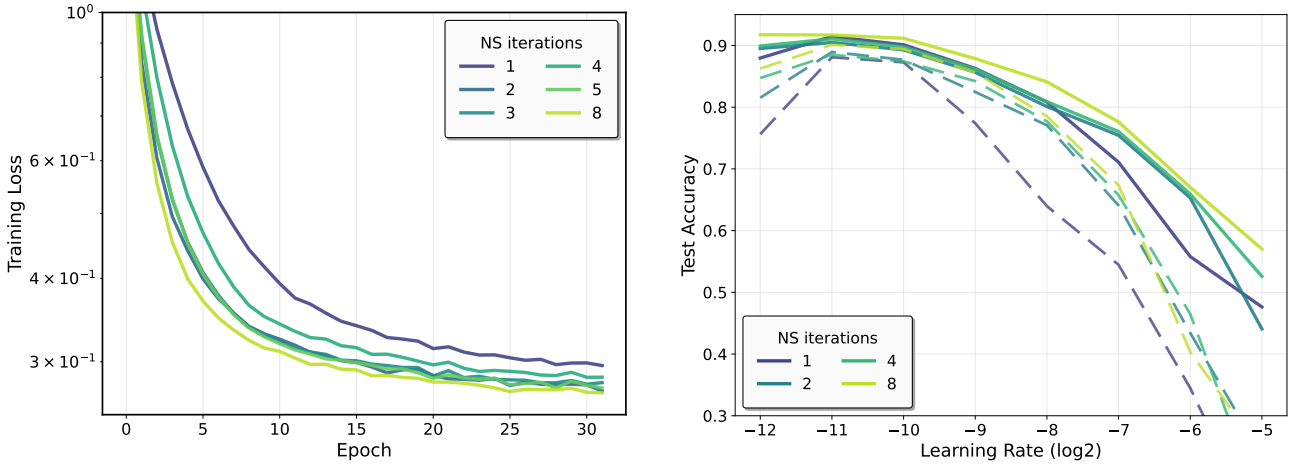
The learning-rate trend is much more sharply resolved than the momentum trend at the resolution of our grid. Near the optimum, the validation loss is substantially flatter in momentum than in learning rate, so discrete sweeps can easily mask the weaker momentum dependence predicted by Corollary 3. For this reason, we interpret the NanoGPT heatmaps primarily as evidence for the learning-rate component of the coupling.

4.2 CNN on CIFAR-10

To test our findings in a different domain, we train a simple CNN on the CIFAR-10 dataset with a batch size of 100. Our implementation is based on the unconstrained `Scion` method (Pethick et al., 2025a), which we run with a constant step size to isolate the effect of inexactness. For these experiments, we use the standard `Newton-Schulz` iteration (Jordan et al., 2024b) to approximate the LMO.

Convergence dynamics. Our theory predicts that a less precise LMO should lead to degraded convergence. Figure 2a confirms this behavior. We plot the training loss over 32 epochs for different numbers of `Newton-Schulz` iterations, using a fixed step size and tuned momentum for each run. The results clearly show that a more precise approximation (e.g., 8 iterations) converges faster and to a lower final training loss compared to less precise approximations (e.g., 1 or 4 iterations). This aligns with the degradation factor present in our theoretical bounds.

Performance and stability versus inexactness. Beyond the training loss, we investigate how LMO inexactness affects the final test accuracy and the optimizer's stability with respect to its hyperparameters. Figure 2b shows the test accuracy achieved after a sweep over a range of step sizes for different levels of LMO precision. The solid lines represent the best ac-



(a) Training loss drops faster as `Newton-Schulz` precision increases.

(b) Best (solid) vs. worst (dashed) accuracy across step sizes; higher precision narrows the gap.

Figure 2: CIFAR-10 training loss and test accuracy across different `Newton-Schulz` (NS) precision levels.

curacy achieved for a given step size (by tuning momentum), while the dashed lines show the worst.

Two key observations can be made. First, the peak of the solid lines (the best possible performance) consistently increases with the number of `Newton-Schulz` iterations, again confirming that higher precision leads to better final performance. Second, the gap between the best (solid) and worst (dashed) performance is large for a highly inexact LMO (1 iteration) and becomes progressively smaller as precision increases. This suggests that a more precise LMO makes the optimizer more robust to the choice of other hyperparameters like momentum, a valuable practical property not explicitly captured by the convergence rate alone.

5 CONCLUSION

This work provides a formal convergence analysis of LMO-based optimizers with an inexact oracle in the unconstrained, non-convex setting, addressing the critical gap between the theory and practice of prominent algorithms like Muon. By introducing a realistic additive error model, our analysis moves beyond the idealized assumption of a perfect oracle to study the algorithm as it is truly implemented. Our analysis, for both deterministic and stochastic settings, reveals that there is a coupling between the LMO inexactness, δ, and the hyperparameter strategy. Our theory prescribes that a less precise LMO requires a more cautious (smaller) step size, a prediction that our experiments confirm.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback and suggestions, which helped improve the paper.

The research reported in this publication was supported by funding from King Abdullah University of Science and Technology (KAUST): i) KAUST Baseline Research Scheme, ii) Center of Excellence for Generative AI, under award number 5940, iii) SDAIA-KAUST Center of Excellence in Artificial Intelligence and Data Science.

References

Noah Amsel, David Persson, Christopher Musco, and Robert M Gower. The polar express: Optimal matrix sign methods and their application to the Muon algorithm. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=yRtgZ1K8hO. (Cited on page 4, 5, 7, 8, and 15)

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. doi: 10.1007/s10107-022-01822-7. (Cited on page 7)

Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, Yichen Feng, Kelin Fu, Bofei Gao, Hongcheng Gao, Peizhong Gao, Tong Gao, Xinran Gu, Longyu Guan, Haiqing Guo, Jianhang Guo, Hao Hu, Xiaoru Hao, Tianhong He, Weiran He, Wenyang He, Chao Hong, Yangyang Hu, Zhenxing Hu, Weixiao Huang, Zhiqi Huang, Zihao Huang, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yongsheng Kang, Guokun Lai, Cheng Li, Fang Li,

Haoyang Li, Ming Li, Wentao Li, Yanhao Li, Yiwei Li, Zhaowei Li, Zheming Li, Hongzhan Lin, Xiaohan Lin, Zongyu Lin, Chengyin Liu, Chenyu Liu, Hongzhang Liu, Jingyuan Liu, Junqi Liu, Liang Liu, Shaowei Liu, T. Y. Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yibo Liu, Yiping Liu, Yue Liu, Zhengying Liu, Enzhe Lu, Lijun Lu, Shengling Ma, Xinyu Ma, Yingwei Ma, Shaoguang Mao, Jie Mei, Xin Men, Yibo Miao, Siyuan Pan, Yebo Peng, Ruoyu Qin, Bowen Qu, Zeyu Shang, Lidong Shi, Shengyuan Shi, Feifan Song, Jianlin Su, Zhengyuan Su, Xinjie Sun, Flood Sung, Heyi Tang, Jiawen Tao, Qifeng Teng, Chensi Wang, Dinglu Wang, Feng Wang, and Haiming Wang. Kimi K2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025. URL https://arxiv.org/abs/2507.20534. (Cited on page 1)

Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology. In *OPT 2024: Optimization for Machine Learning*, 2024. URL https://arxiv.org/abs/2409.20325. (Cited on page 2 and 3)

David Carlson, Volkan Cevher, and Lawrence Carin. Stochastic spectral descent for restricted boltzmann machines. In *Artificial Intelligence and Statistics*, pages 111–119. PMLR, 2015a. (Cited on page 3)

David Carlson, Ya-Ping Hsieh, Edo Collins, Lawrence Carin, and Volkan Cevher. Stochastic spectral descent for discrete graphical models. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):296–311, 2016. doi: 10.1109/JSTSP.2015.2505684. (Cited on page 3)

David E Carlson, Edo Collins, Ya-Ping Hsieh, Lawrence Carin, and Volkan Cevher. Preconditioned spectral descent for deep learning. *Advances in Neural Information Processing Systems*, 28:2971–2979, 2015b. (Cited on page 3)

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *Mathematical Programming*, 184(1):71–120, 2020. doi: 10.1007/s10107-019-01406-y. (Cited on page 5)

Franz Louis Cesista, Jiacheng You, and Keller Jordan. Squeezing 1-2% efficiency gains out of Muon by optimizing the Newton-Schulz coefficients, February 2025. URL https://leloykun.github.io/ponder/muon-opt-coeffs/. (Cited on page 4)

Lizhang Chen, Jonathan Li, and Qiang Liu. Muon optimizes under spectral norm constraints. *arXiv preprint arXiv:2506.15054*, 2025. URL https://arxiv.org/abs/2506.15054. (Cited on page 1)

Kenneth L Clarkson. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Transactions on Algorithms*, 6(4):1–30, 2010. doi: 10.1145/1824777.1824783. (Cited on page 2)

Ashok Cutkosky and Harsh Mehta. Momentum improves normalized SGD. In *International Conference on Machine Learning*, pages 2260–2268. PMLR, 2020. (Cited on page 2)

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956. doi: 10.1002/nav.3800030109. (Cited on page 2)

Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013. doi: 10.1137/120880811. (Cited on page 7)

Ekaterina Grishina, Matvey Smirnov, and Maxim Rakhuba. Accelerating Newton-Schulz iteration for orthogonalization via Chebyshev-type polynomials. *arXiv preprint arXiv:2506.10935*, 2025. URL https://arxiv.org/abs/2506.10935. (Cited on page 1, 4, 5, and 7)

Kaja Gruntkowska and Peter Richtárik. Non-Euclidean broximal point method: a blueprint for geometry-aware optimization. *arXiv preprint arXiv:2510.00823*, 2025. URL https://arxiv.org/abs/2510.00823. (Cited on page 3)

Kaja Gruntkowska, Hanmin Li, Aadi Rane, and Peter Richtárik. The ball-proximal (="broximal") point method: a new algorithm, convergence theory, and applications. *arXiv preprint arXiv:2502.02002*, 2025a. URL https://arxiv.org/abs/2502.02002. (Cited on page 3)

Kaja Gruntkowska, Yassine Maziane, Zheng Qu, and Peter Richtárik. Drop-Muon: Update less, converge faster. *arXiv preprint arXiv:2510.02239*, 2025b. URL https://arxiv.org/abs/2510.02239. (Cited on page 3)

Kaja Gruntkowska, Alexander Gaponov, Zhirayr Tovmasyan, and Peter Richtárik. Error feedback for Muon and friends. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=rex7s82Iav. (Cited on page 3)

Elad Hazan. Sparse approximate solutions to semidefinite programs. In *Latin American Symposium on Theoretical Informatics*, pages 306–316. Springer, 2008. doi: 10.1007/978-3-540-78773-0_27. (Cited on page 2)

Chuan He, Zhanwang Deng, and Zhaosong Lu. Low-rank orthogonalization for large-scale matrix optimization with applications to foundation model training. *arXiv preprint arXiv:2509.11983*, 2025.

URL https://arxiv.org/abs/2509.11983. (Cited on page 15)

Nicholas J Higham. *Functions of matrices: Theory and computation*. SIAM, 2008. (Cited on page 4 and 15)

Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*, pages 427–435. PMLR, 2013. URL https://proceedings.mlr.press/v28/jaggi13.html. (Cited on page 2)

Keller Jordan, Jeremy Bernstein, Ben Rappazzo, Vlado Boža, Jiacheng You, Franz Cesista, and Braden Koszarsky. Modded-nanoGPT: Speedrunning the nanoGPT baseline, 2024a. URL https://github.com/KellerJordan/modded-nanogpt. (Cited on page 1 and 8)

Keller Jordan, Yuchen Jin, Vlado Boža, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024b. URL https://kellerjordan.github.io/posts/muon/. (Cited on page 1, 2, 3, 4, and 8)

Ahmed Khaled, Kaan Ozkara, Tao Yu, Mingyi Hon, and Youngsuk Par. MuonBP: Faster Muon via block-periodic orthogonalization. *arXiv preprint arXiv:2510.16981*, 2025. URL https://arxiv.org/abs/2510.16981. (Cited on page 3)

Gyu Yeol Kim and Min-hwan Oh. Convergence of Muon with Newton–Schulz. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=lJSfxtLpLm. (Cited on page 3 and 15)

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980. (Cited on page 1)

Dmitry Kovalev. Understanding gradient orthogonalization for deep learning via non-Euclidean trust-region optimization, 2025. URL https://arxiv.org/abs/2503.12645. (Cited on page 2, 3, 5, and 6)

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical Report Technical Report TR-2009, University of Toronto, 2009. (Cited on page 7)

Simon Lacoste-Julien, Martin Jaggi, Mark Schmidt, and Patrick Pletscher. Block-coordinate frank-wolfe optimization for structural svms. In *International Conference on Machine Learning*, pages 53–61. PMLR, 2013. (Cited on page 2)

Jiaxiang Li and Mingyi Hong. A note on the convergence of Muon. *arXiv preprint arXiv:2502.02900*, 2025. URL https://arxiv.org/abs/2502.02900. (Cited on page 1 and 2)

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. Muon is scalable for LLM training. *arXiv preprint arXiv:2502.16982*, 2025. URL https://arxiv.org/abs/2502.16982. (Cited on page 1)

Francesco Locatello, Rajiv Khanna, Michael Tschannen, and Martin Jaggi. A unified optimization view on generalized matching pursuit and frank-wolfe. In *Artificial intelligence and statistics*, pages 860–868. PMLR, 2017a. (Cited on page 2)

Francesco Locatello, Michael Tschannen, Gunnar Rätsch, and Martin Jaggi. Greedy algorithms for cone constrained optimization with convergence guarantees. *Advances in Neural Information Processing Systems*, 30, 2017b. (Cited on page 2)

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7. (Cited on page 1)

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin A Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37: 30811–30849, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/370df50ccfdf8bde18f8f9c2d9151bda-Abstract-Datasets_and_Benchmarks_Track.html. (Cited on page 7)

Thomas Pethick, Wanyun Xie, Kimon Antonakopoulos, Zhenyu Zhu, Antonio Silveti-Falls, and Volkan Cevher. Scion, 2025a. URL https://github.com/LIONS-EPFL/scion.git. (Cited on page 8)

Thomas Pethick, Wanyun Xie, Kimon Antonakopoulos, Zhenyu Zhu, Antonio Silveti-Falls, and Volkan Cevher. Training deep learning models with norm-constrained LMOs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 49069–49104. PMLR, 2025b. URL https://proceedings.mlr.press/v267/pethick25a.html. (Cited on page 2, 3, 5, 6, and 7)

Thomas Pethick, Wanyun Xie, Mete Erdogan, Kimon Antonakopoulos, Antonio Silveti-Falls, and Volkan Cevher. Generalized gradient norm clip-

ping & non-euclidean (L_0, L_1)-smoothness. In *Advances in Neural Information Processing Systems*, 2025c. URL https://openreview.net/forum?id=rMdf8jhLR7. (Cited on page 3)

Yehonathan Refael, Guy Smorodinsky, Tom Tirer, and Ofir Lindenbaum. SUMO: Subspace-aware moment-orthogonalization for accelerating memory-efficient LLM training. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=DIjRvEKOeG. (Cited on page 15)

Artem Riabinin, Egor Shulgin, Kaja Gruntkowska, and Peter Richtárik. Gluon: Making Muon & Scion great again! (Bridging theory and practice of LMO-based optimizers for LLMs). *arXiv preprint arXiv:2505.13416*, 2025. URL https://arxiv.org/abs/2505.13416. (Cited on page 2, 3, 7, 30, and 32)

Ishaan Shah, Anthony M. Polloreno, Karl Stratos, Philip Monk, Adarsh Chaluvaraju, Andrew Hojel, Andrew Ma, Anil Thomas, Ashish Tanwer, Darsh J. Shah, Khoi Nguyen, Kurt Smith, Michael Callahan, Michael Pust, Mohit Parmar, Peter Rushton, Platon Mazarakis, Ritvik Kapila, Saurabh Srivastava, Somanshu Singla, Tim Romanski, Yash Vanjani, and Ashish Vaswani. Practical efficiency of Muon for pretraining. *arXiv preprint arXiv:2505.02222*, 2025. URL https://arxiv.org/abs/2505.02222. (Cited on page 1)

Wei Shen, Ruichuan Huang, Minhui Huang, Cong Shen, and Jiawei Zhang. On the convergence analysis of Muon. *arXiv preprint arXiv:2505.23737*, 2025. URL https://arxiv.org/abs/2505.23737. (Cited on page 1)

Antonio Silveti-Falls, Cesare Molinari, and Jalal Fadili. Inexact and stochastic generalized conditional gradient with augmented lagrangian and proximal step. *Journal of Nonsmooth Analysis and Optimization*, 2(Original research articles), 2021. (Cited on page 2)

Yuki Takezawa, Anastasia Koloskova, Xiaowen Jiang, and Sebastian U. Stich. FedMuon: Federated learning with bias-corrected LMO-based optimization. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=9k7bvBVenZ. (Cited on page 15)

Mark Tuddenham, Adam Prügel-Bennett, and Jonathan Hare. Orthogonalising gradients to speed up neural network optimisation. *arXiv preprint arXiv:2202.07052*, 2022. URL https://arxiv.org/abs/2202.07052. (Cited on page 3)

Daniil Vankov, Anton Rodomanov, Angelia Nedich, Lalitha Sankar, and Sebastian U Stich. Optimizing (L_0, L_1)-smooth functions by gradient methods. In *Proceedings of the International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=GQ1Tc3vHbt. (Cited on page 7)

Kaiyue Wen, David Leo Wright Hall, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=2J51qUZ0iG. (Cited on page 1)

Junchi Yang, Xiang Li, Ilyas Fatkhullin, and Niao He. Two sides of one coin: The limits of untuned SGD and the power of adaptive methods. *Advances in Neural Information Processing Systems*, 36:74257–74288, 2023. URL https://openreview.net/forum?id=fUZUoSLXw3. (Cited on page 29 and 30)

Dingzhi Yu, Wei Jiang, Yuanyu Wan, and Lijun Zhang. Mirror descent under generalized smoothness. *arXiv preprint arXiv:2502.00753*, 2025. URL https://arxiv.org/abs/2502.00753. (Cited on page 7)

Alp Yurtsever, Olivier Fercoq, Francesco Locatello, and Volkan Cevher. A conditional gradient framework for composite convex minimization with applications to semidefinite programming. In *International conference on machine learning*, pages 5727–5736. PMLR, 2018. (Cited on page 2)

Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJgnXpVYwS. (Cited on page 2 and 7)

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. Yes
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. Yes
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. No
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. Yes
   (b) Complete proofs of all theoretical results. Yes
   (c) Clear explanations of any assumptions. Yes
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). Yes
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). Yes
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). Not Applicable
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). Yes
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. Yes
   (b) The license information of the assets, if applicable. No
   (c) New assets either in the supplemental material or as a URL, if applicable. Not Applicable
   (d) Information about consent from data providers/curators. Not Applicable
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. Not Applicable
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. Not Applicable
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. Not Applicable
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. Not Applicable

Beyond the Ideal: Analyzing the Inexact Muon Update Supplementary Materials

Contents

1 INTRODUCTION **1**
- 1.1 Contributions . . . 2
- 1.2 Related work . . . 2

2 FROM IDEALIZED THEORY TO PRACTICAL IMPLEMENTATION **3**

3 MAIN THEORETICAL RESULTS **4**
- 3.1 Deterministic case . . . 4
- 3.2 Stochastic case . . . 5
- 3.3 Extensions and generalizations . . . 7

4 EXPERIMENTS **7**
- 4.1 nanoGPT on FineWeb . . . 7
- 4.2 CNN on CIFAR-10 . . . 8

5 CONCLUSION **9**

A JUSTIFICATION OF THE INEXACT LMO ERROR MODEL **14**
- A.1 Additional discussion of recent closely related work . . . 15

B PROOFS **16**
- B.1 Deterministic case . . . 16
  - B.1.1 Proof of Theorem 1 (general result) . . . 16
  - B.1.2 Proof of Corollary 1 (constant parameters) . . . 17
  - B.1.3 Proof of Corollary 2 (optimal parameters) . . . 17
- B.2 Analysis with an adaptive step size . . . 18
- B.3 Stochastic case . . . 20
  - B.3.1 Proof of Lemma 1 . . . 21
  - B.3.2 Proof of Theorem 2 . . . 23
  - B.3.3 Proof of Corollary 3 . . . 24
- B.4 Convergence with time-varying parameters . . . 26

C EXTENSIONS **30**
- C.1 (L^0, L^1)-smoothness . . . 30
- C.2 Layer-wise analysis . . . 32

D ADDITIONAL EXPERIMENTS AND DETAILS **35**
- D.1 Experimental setup: details . . . 35
- D.2 Additional results for nanoGPT . . . 35
  - D.2.1 LMO Error Analysis. . . . 35
- D.3 Additional results for CIFAR-10 . . . 38

A JUSTIFICATION OF THE INEXACT LMO ERROR MODEL

In this section, we provide additional details on how practical approximation schemes for the orthogonalized update satisfy Assumption 1.

The PolarExpress algorithm (Amsel et al., 2026) is designed to approximate the polar factor $U = \mathrm{polar}(M)$ of a matrix M through iterative refinement. Theorem 4.3 in Amsel et al. (2026) establishes that for a matrix M with singular values normalized to lie in $[\ell, 1]$, the PolarExpress approximation after $r = 2q+1$ (odd) iterations satisfies:

$$\|\,\mathrm{polar}(M) - X_p\|_2 \le |1-\ell^2|^{(q+1)^p},$$

where $\|\cdot\|_2$ denotes the spectral norm (largest singular value). For $r = 3$ this yields quadratic convergence, and for $r = 5$ cubic convergence. We refer to Amsel et al. (2026) for a comprehensive analysis including extensions to rectangular matrices and detailed convergence properties.

The classical Newton-Schulz iteration (Higham, 2008) exhibits similar convergence properties, with the approximation error decreasing rapidly with the number of iterations.

Since these methods are always run for a small, fixed number of steps in practice (e.g., $p = 5$ in Muon), their error is bounded by a constant δ. Different iteration counts correspond to different values of δ, with more iterations producing smaller error. This directly justifies the additive error model in Assumption 1.

A.1 Additional discussion of recent closely related work

Several recent papers reinforce the same broad message as ours—that the quality of the orthogonalization step matters—but from more specialized angles. In contrast, our paper studies a general additive inexact-LMO model, allows infeasible approximate directions, covers both deterministic and stochastic settings, and derives explicit δ-dependent prescriptions for the step size and momentum.

SUMO. Refael et al. (2025) analyze the approximation error of the commonly used Newton–Schulz5 orthogonalization and show that it worsens with the condition number of the moment matrix. They then compare Muon using Newton–Schulz5 to exact SVD-based orthogonalization and advocate replacing the former by exact orthogonalization in an adaptive low-rank subspace. Thus, their treatment of inexactness is algorithm-specific and condition-number-driven, and the remedy is to modify the optimizer itself. We view this as complementary to our work: they provide a concrete mechanism explaining *why* approximation quality can deteriorate, whereas our analysis isolates how a generic inexactness level δ propagates into convergence rates and tuning rules.

Low-rank orthogonalization. He et al. (2025) propose a different inner solver based on low-rank orthogonalization via sketching, and then build low-rank matrix-signed gradient descent and a low-rank variant of Muon on top of it. Their theory is tailored to this modified direction computation, giving deterministic stationarity guarantees for the low-rank signed method and stochastic guarantees for low-rank Muon under heavy-tailed noise. In particular, they do not analyze the original practical Muon update under a generic approximation model; instead, they replace the orthogonalization step by a low-rank surrogate. The complementary insight is that orthogonalization error may be reduced not only by running more inner iterations, but also by restricting attention to a more informative low-rank subspace.

FedMuon. Takezawa et al. (2026) study inexact Newton–Schulz orthogonalization in FedMuon. The most relevant point for our paper is that, in the spectral-norm/Newton–Schulz setting, they prove a solver-specific result that is in some sense sharper than a generic perturbation bound: the method can converge for *any* number of Newton–Schulz iterations, with the approximate-LMO guarantee stated in a q-dependent Schatten-p norm that improves toward the trace-norm guarantee as q increases. This differs substantially from our abstraction, which is centralized, solver-agnostic, and based on an additive direction error $\|\hat{d} - d\| \le \delta$. We therefore view Takezawa et al. (2026) as complementary evidence that specialized analyses can exploit the structure of a particular solver more sharply than a general inexact-LMO framework.

Convergence of Muon with Newton–Schulz. The closest concurrent work is Kim and Oh (2026), which analyzes the original Muon algorithm with finitely many Newton–Schulz steps. Their results are specialized to operator-nuclear geometry and show that Muon with Newton–Schulz matches the exact SVD-polar rate up to a q-dependent constant factor, which approaches 1 doubly exponentially fast in the number of Newton–Schulz steps. Compared to our paper, this is narrower but sharper: their analysis is explicitly tied to Newton–Schulz and yields a direct dependence on the inner-iteration count q, while our framework covers multiple approximation schemes (e.g., Newton–Schulz and PolarExpress), allows infeasible approximate directions, and yields generic δ-dependent guidance for learning rate and momentum. In this sense, Kim and Oh (2026) refine the Newton–Schulz-specific part of the story that our broader inexact-LMO model abstracts.

B PROOFS

B.1 Deterministic case

B.1.1 Proof of Theorem 1 (general result)

The proof establishes a general convergence guarantee for the deterministic inexact LMO method. It proceeds by deriving a per-iteration descent inequality and then summing it over all iterations to obtain a global bound.

Proof. **Per-Iteration Descent Inequality.** We begin with the standard L-smoothness inequality for f, as stated in Assumption 2:

$$f(x^{k+1}) \leq f(x^k) + \left\langle \nabla f(x^k), x^{k+1} - x^k \right\rangle + \frac{L}{2}\|x^{k+1} - x^k\|^2.$$

Let $g^k := \nabla f(x^k)$. We substitute the update rule from Equation (5), $x^{k+1} - x^k = \gamma^k \hat{d}^k$, into the inequality:

$$f(x^{k+1}) \leq f(x^k) + \gamma^k \left\langle g^k, \hat{d}^k \right\rangle + \frac{L}{2}\|\gamma^k \hat{d}^k\|^2. \tag{6}$$

We now bound the two rightmost terms involving the inexact direction $\hat{d}^k$.

Bounding the Squared Norm Term. Let d^k be the exact LMO solution for the gradient g^k. We use the triangle inequality to bound the norm of the inexact direction $\|\hat{d}^k\|$:

$$\|\hat{d}^k\| = \|\hat{d}^k - d^k + d^k\| \leq \|\hat{d}^k - d^k\| + \|d^k\|.$$

Using the inexactness from Assumption 1 ($\|\hat{d}^k - d^k\| \leq \delta_k$) and the property of the exact LMO solution ($\|d^k\| \leq 1$), we obtain $\|\hat{d}^k\| \leq \delta_k + 1$. Therefore, the squared norm term is bounded as:

$$\|\gamma^k \hat{d}^k\|^2 = (\gamma^k)^2 \|\hat{d}^k\|^2 \leq (\gamma^k)^2 (1 + \delta_k)^2.$$

Bounding the Inner Product Term. We decompose the inner product using the exact direction d^k:

$$\left\langle g^k, \hat{d}^k \right\rangle = \left\langle g^k, d^k + (\hat{d}^k - d^k) \right\rangle = \left\langle g^k, d^k \right\rangle + \left\langle g^k, \hat{d}^k - d^k \right\rangle.$$

By the definition of the LMO and the dual norm, the first term is exactly $-\|g^k\|_\star$. The second term is bounded using the definition of dual norm:

$$\left\langle g^k, \hat{d}^k - d^k \right\rangle \leq \left| \left\langle g^k, \hat{d}^k - d^k \right\rangle \right| \leq \|g^k\|_\star \|\hat{d}^k - d^k\| \leq \|g^k\|_\star \delta_k.$$

Combining these results gives an upper bound on the inner product:

$$\left\langle g^k, \hat{d}^k \right\rangle \leq -\|g^k\|_\star + \|g^k\|_\star \delta_k = -\|g^k\|_\star (1 - \delta_k).$$

Combining the Bounds. Substituting these two bounds back into the smoothness inequality (6), we obtain the per-iteration descent guarantee:

$$f(x^{k+1}) \leq f(x^k) - \gamma^k \|g^k\|_\star (1 - \delta_k) + \frac{L}{2}(\gamma^k)^2 (1 + \delta_k)^2.$$

Global Convergence Bound. For the descent property to hold, we require $1 - \delta_k > 0$. Rearranging the per-iteration inequality to isolate the stationarity measure yields:

$$\gamma^k \|\nabla f(x^k)\|_\star (1 - \delta_k) \leq f(x^k) - f(x^{k+1}) + \frac{L}{2}(\gamma^k)^2 (1 + \delta_k)^2.$$

Summing this inequality from $k = 0$ to $K - 1$:

$$\sum_{k=0}^{K-1} \gamma^k \|\nabla f(x^k)\|_\star (1 - \delta_k) \leq \sum_{k=0}^{K-1} \left(f(x^k) - f(x^{k+1})\right) + \frac{L}{2} \sum_{k=0}^{K-1} (\gamma^k)^2 (1 + \delta_k)^2$$
$$= f(x^0) - f(x^K) + \frac{L}{2} \sum_{k=0}^{K-1} (\gamma^k)^2 (1 + \delta_k)^2.$$

Using the fact that $f(x^K) \geq f^*$ and letting $\Delta^0 = f(x^0) - f^*$, we have $f(x^0) - f(x^K) \leq \Delta^0$. This gives the general bound

$$\sum_{k=0}^{K-1} \gamma^k \|\nabla f(x^k)\|_\star (1 - \delta_k) \leq \Delta^0 + \frac{L}{2} \sum_{k=0}^{K-1} (\gamma^k)^2 (1 + \delta_k)^2.$$

To bound the minimum gradient norm, we note that for any non-negative sequence $\{a_k\}$ and positive sequence $\{w_k\}$, we have

$$\min_{0 \leq j < K} a_j \cdot \sum_{k=0}^{K-1} w_k \leq \sum_{k=0}^{K-1} w_k a_k.$$

Applying this with $a_k = \|\nabla f(x^k)\|_\star$ and $w_k = \gamma^k (1 - \delta_k)$, we get:

$$\left(\min_{0 \leq j < K} \|\nabla f(x^j)\|_\star\right) \sum_{k=0}^{K-1} \gamma^k (1 - \delta_k) \leq \sum_{k=0}^{K-1} \gamma^k \|\nabla f(x^k)\|_\star (1 - \delta_k).$$

Combining the inequalities and dividing by the sum $\sum \gamma^k (1 - \delta_k)$ yields the final result stated in Theorem 1. □

B.1.2 Proof of Corollary 1 (constant parameters)

Proof. The proof follows by simplifying the general bound from Theorem 1 under the specific assumptions of the corollary. We are given that the step size is constant, $\gamma^k = \gamma > 0$, and the LMO error is constant, $\delta_k = \delta < 1$, for all $k = 0, \ldots, K - 1$.

We start with the summed inequality from the proof of Theorem 1:

$$\sum_{k=0}^{K-1} \gamma (1 - \delta) \|\nabla f(x^k)\|_\star \leq \Delta^0 + \frac{L}{2} \sum_{k=0}^{K-1} \gamma^2 (1 + \delta)^2.$$

The constants can be factored out of the summations:

$$\gamma (1 - \delta) \sum_{k=0}^{K-1} \|\nabla f(x^k)\|_\star \leq \Delta^0 + \frac{L K \gamma^2 (1 + \delta)^2}{2}.$$

To obtain a bound on the average gradient norm, we divide the entire inequality by $K\gamma(1 - \delta)$:

$$\frac{1}{K} \sum_{k=0}^{K-1} \|\nabla f(x^k)\|_\star \leq \frac{\Delta^0}{K\gamma(1 - \delta)} + \frac{L\gamma(1 + \delta)^2}{2(1 - \delta)},$$

which gives the final result stated in Corollary 1. □

B.1.3 Proof of Corollary 2 (optimal parameters)

Proof. The goal is to find the constant step size γ that minimizes the upper bound on the average gradient norm derived in Corollary 1. Let the upper bound be denoted by the function $\mathcal{E}(\gamma)$:

$$\mathcal{E}(\gamma) = \frac{\Delta^0}{K\gamma(1 - \delta)} + \frac{L\gamma(1 + \delta)^2}{2(1 - \delta)}.$$

This expression is of the form $A/\gamma + B\gamma$, where $A = \frac{\Delta^0}{K(1-\delta)}$ and $B = \frac{L(1+\delta)^2}{2(1-\delta)}$ are positive constants. This function is convex for $\gamma > 0$. To find the minimizer γ^*, we take the derivative with respect to γ and set it to zero:

$$\frac{\partial \mathcal{E}}{\partial \gamma} = -\frac{A}{\gamma^2} + B = 0 \implies (\gamma^*)^2 = \frac{A}{B}.$$

Substituting the expressions for A and B, we obtain

$$(\gamma^*)^2 = \frac{\frac{\Delta^0}{K(1-\delta)}}{\frac{L(1+\delta)^2}{2(1-\delta)}} = \frac{\Delta^0}{K(1-\delta)} \cdot \frac{2(1-\delta)}{L(1+\delta)^2} = \frac{2\Delta^0}{KL(1+\delta)^2}.$$

Taking the square root gives the optimal constant step size:

$$\gamma^* = \sqrt{\frac{2\Delta^0}{KL(1+\delta)^2}} = \frac{1}{1+\delta}\sqrt{\frac{2\Delta^0}{KL}}.$$

To find the best achievable convergence rate, we substitute this optimal step size γ^* back into the bound $\mathcal{E}(\gamma)$. At the minimum, the two terms A/γ^* and $B\gamma^*$ are equal, so the total error is $2\sqrt{AB}$:

$$\begin{aligned}
\mathcal{E}(\gamma^*) = 2\sqrt{AB} &= 2\sqrt{\frac{\Delta^0}{K(1-\delta)} \cdot \frac{L(1+\delta)^2}{2(1-\delta)}} \\
&= 2\frac{\sqrt{\Delta^0 L}(1+\delta)}{\sqrt{2K}(1-\delta)} \\
&= \frac{\sqrt{2\Delta^0 L}}{\sqrt{K}} \frac{1+\delta}{1-\delta}.
\end{aligned}$$

This confirms the final optimized convergence rate stated in Corollary 2. □

Corollary 4 (Iteration Complexity with Constant Step Size)**.** *Under the conditions of Corollary 1, to guarantee that the average stationarity measure satisfies* $\frac{1}{K}\sum_{k=0}^{K-1} \|\nabla f(x^k)\|_\star \leq \varepsilon$ *for a target precision* $\varepsilon > 0$*, it is sufficient to run the algorithm for a number of iterations* K *of the order of*

$$\mathcal{O}\left(\frac{L\Delta^0}{\varepsilon^2} \cdot \frac{(1+\delta)^2}{(1-\delta)^2}\right).$$

Proof. The result is obtained by setting the optimized bound from Corollary 2, $\frac{1+\delta}{1-\delta}\sqrt{\frac{2\Delta^0 L}{K}}$, to be less than or equal to ε and solving for K. □

B.2 Analysis with an adaptive step size

In the main text, we focus on the analysis with constant or pre-defined diminishing step sizes. Here, we present a complementary result for the deterministic case that considers an adaptive step size, chosen at each iteration to maximize the guaranteed descent. This analysis further highlights the impact of the inexactness level δ_k.

Theorem 3. *Let Assumption 2 hold and let the LMO errors be a sequence* $\{\delta_k\}$ *with* $\delta_k \in [0, 1)$*. If the algorithm is run with the time-varying adaptive step size*

$$\gamma^k = \frac{\left\|\nabla f(x^k)\right\|_\star (1-\delta_k)}{L(1+\delta_k)^2},$$

then the minimum gradient norm after K *iterations is bounded by:*

$$\min_{0\leq j<K} \left\|\nabla f(x^j)\right\|_\star^2 \leq \frac{2L\Delta^0}{\sum_{k=0}^{K-1} \frac{(1-\delta_k)^2}{(1+\delta_k)^2}}. \tag{7}$$

Proof. The proof starts from the per-iteration descent inequality derived in the proof of Theorem 1:

$$f(x^{k+1}) \leq f(x^k) - \gamma^k \left\|\nabla f(x^k)\right\|_\star (1-\delta_k) + \frac{L(\gamma^k)^2(1+\delta_k)^2}{2}.$$

The adaptive step size γ^k is chosen to minimize the right-hand side of this inequality, which is a quadratic in γ^k. Its minimizer is precisely the step size given in the theorem statement. Substituting this optimal choice of γ^k back into the inequality yields

$$\begin{aligned} f(x^{k+1}) &\leq f(x^k) - \frac{\left\|\nabla f(x^k)\right\|_\star^2 (1-\delta_k)^2}{L(1+\delta_k)^2} + \frac{L}{2}\left(\frac{\left\|\nabla f(x^k)\right\|_\star (1-\delta_k)}{L(1+\delta_k)^2}\right)^2 (1+\delta_k)^2 \\ &= f(x^k) - \frac{\left\|\nabla f(x^k)\right\|_\star^2 (1-\delta_k)^2}{2L(1+\delta_k)^2}. \end{aligned}$$

Rearranging gives a lower bound on the progress at step k:

$$\frac{(1-\delta_k)^2}{(1+\delta_k)^2}\left\|\nabla f(x^k)\right\|_\star^2 \leq 2L\left(f(x^k) - f(x^{k+1})\right).$$

Summing from $k=0$ to $K-1$, we get

$$\sum_{k=0}^{K-1} \frac{(1-\delta_k)^2}{(1+\delta_k)^2}\left\|\nabla f(x^k)\right\|_\star^2 \leq 2L\sum_{k=0}^{K-1}\left(f(x^k) - f(x^{k+1})\right) \leq 2L\Delta^0.$$

Let $w_k = \frac{(1-\delta_k)^2}{(1+\delta_k)^2}$. We have $\sum_{k=0}^{K-1} w_k \left\|\nabla f(x^k)\right\|_\star^2 \leq 2L\Delta^0$. Since $\left\|\nabla f(x^j)\right\|_\star^2 \geq \min_{0\leq i<K}\left\|\nabla f(x^i)\right\|_\star^2$ for any j, we have

$$\left(\min_{0\leq i<K}\left\|\nabla f(x^i)\right\|_\star^2\right)\sum_{k=0}^{K-1} w_k \leq \sum_{k=0}^{K-1} w_k \left\|\nabla f(x^k)\right\|_\star^2 \leq 2L\Delta^0.$$

Solving for the minimum squared gradient norm gives the desired inequality:

$$\min_{0\leq j<K}\left\|\nabla f(x^j)\right\|_\star^2 \leq \frac{2L\Delta^0}{\sum_{k=0}^{K-1}\frac{(1-\delta_k)^2}{(1+\delta_k)^2}}.$$

□

Theorem 3 provides a general expression for the convergence rate, which is governed by the growth of the sum $\sum_{k=0}^{K-1}\frac{(1-\delta_k)^2}{(1+\delta_k)^2}$ in the denominator of the bound. This allows us to analyze the impact of specific schedules for the inexactness level $\{\delta_k\}$. The following corollary establishes a simple and practical sufficient condition for achieving the optimal rate.

Corollary 5. *Under the assumptions of Theorem 3, the optimal convergence rate of $\mathcal{O}(1/\sqrt{K})$ for the minimum gradient norm is achieved if the error sequence $\{\delta_k\}$ is uniformly bounded away from 1. Specifically, if there exists a $\delta_{\max} = \delta \in [0,1)$ such that $\delta_k \leq \delta_{\max}$ for all k, then*

$$\min_{0\leq j<K}\left\|\nabla f(x^j)\right\|_\star \leq \frac{1+\delta}{1-\delta}\sqrt{\frac{2L\Delta^0}{K}}. \tag{8}$$

Proof. If $\delta_k \leq \delta$, then the term $w_k := \frac{(1-\delta_k)^2}{(1+\delta_k)^2}$ is bounded below by a positive constant:

$$w_k \geq \frac{(1-\delta)^2}{(1+\delta)^2} > 0.$$

Therefore, the sum in the denominator of (7) grows at least linearly with K:

$$\sum_{k=0}^{K-1} w_k \geq \sum_{k=0}^{K-1}\frac{(1-\delta)^2}{(1+\delta)^2} = K\frac{(1-\delta)^2}{(1+\delta)^2}.$$

Substituting this lower bound into the inequality (7) and taking the square root of both sides yields the desired result. □

Corollary 6 (Iteration Complexity with Adaptive Step Size)**.** *Under the conditions of Corollary 5, to guarantee that the minimum gradient norm satisfies* $\min_{0\le j<K} \|\nabla f(x^j)\|_\star \le \varepsilon$ *for a target precision* $\varepsilon > 0$*, it is sufficient to run the algorithm for a number of iterations* K *of the order of*

$$\mathcal{O}\left(\frac{L\Delta^0}{\varepsilon^2} \cdot \frac{(1+\delta)^2}{(1-\delta)^2}\right).$$

Proof. The result is obtained by setting the bound from Corollary 5 to be less than or equal to ε and solving for K. □

B.3 Stochastic case

This section provides the full proofs for the main theoretical results in the stochastic setting. The analysis proceeds in three main stages. First, in Lemma 2, we establish a crucial per-iteration descent guarantee that holds deterministically for any single step of the algorithm. Second, in Lemma 1, we derive a bound on the expected error of the momentum term. Finally, in the proof of Theorem 2, we combine these results to derive the global convergence guarantee.

Lemma 2 (Inexact Descent Lemma)**.** *Let Assumption 2 hold. For any iteration* k *of Algorithm 1, the following inequality holds:*

$$f(x^{k+1}) \le f(x^k) - \gamma^k \|m^{k+1}\|_\star (1-\delta_k) + \gamma^k (1+\delta_k)\|\nabla f(x^{k+1}) - m^{k+1}\|_\star + \frac{3L}{2}(\gamma^k)^2(1+\delta_k)^2.$$

Proof. The proof establishes a bound on the progress made in a single step. We start with the L-smoothness inequality of f from Assumption 2:

$$f(x^{k+1}) \le f(x^k) + \left\langle \nabla f(x^k), x^{k+1} - x^k \right\rangle + \frac{L}{2}\|x^{k+1} - x^k\|^2.$$

Substituting the update rule $x^{k+1} - x^k = \gamma^k \hat{d}^k$ gives

$$f(x^{k+1}) \le f(x^k) + \gamma^k \left\langle \nabla f(x^k), \hat{d}^k \right\rangle + \frac{L}{2}\|\gamma^k \hat{d}^k\|^2. \tag{9}$$

We decompose the inner product term by introducing the momentum vector m^{k+1}:

$$\gamma^k \left\langle \nabla f(x^k), \hat{d}^k \right\rangle = \gamma^k \left\langle m^{k+1}, \hat{d}^k \right\rangle + \gamma^k \left\langle \nabla f(x^k) - m^{k+1}, \hat{d}^k \right\rangle.$$

The proof proceeds by bounding the key terms involving the inexact direction $\hat{d}^k$.

1. Bounding the norm of the inexact direction. Let $d^k := \operatorname{argmin}_{\|d\|\le 1} \left\langle m^{k+1}, d \right\rangle$ be the exact LMO solution. We use the triangle inequality to bound $\|\hat{d}^k\|$:

$$\|\hat{d}^k\| = \|\hat{d}^k - d^k + d^k\| \le \|\hat{d}^k - d^k\| + \|d^k\|.$$

Using Assumption 1 ($\|\hat{d}^k - d^k\| \le \delta_k$) and $\|d^k\| \le 1$, we have $\|\hat{d}^k\| \le \delta_k + 1$. This provides a bound for the quadratic term from smoothness:

$$\|\gamma^k \hat{d}^k\|^2 = (\gamma^k)^2 \|\hat{d}^k\|^2 \le (\gamma^k)^2 (1+\delta_k)^2.$$

2. Bounding the LMO inner product term. We explicitly account for the LMO error δ_k:

$$\begin{aligned}
\left\langle m^{k+1}, \hat{d}^k \right\rangle &= \left\langle m^{k+1}, d^k \right\rangle + \left\langle m^{k+1}, \hat{d}^k - d^k \right\rangle \\
&\le -\|m^{k+1}\|_\star + |\left\langle m^{k+1}, \hat{d}^k - d^k \right\rangle| \\
&\le -\|m^{k+1}\|_\star + \|m^{k+1}\|_\star \|\hat{d}^k - d^k\| \quad \text{(by definition of dual norm)} \\
&\le -\|m^{k+1}\|_\star (1-\delta_k).
\end{aligned}$$

3. Bounding the momentum error term. We introduce $\nabla f(x^{k+1})$ to align the error term with the structure needed for subsequent analysis:

$$\begin{aligned}\left\langle \nabla f(x^k) - m^{k+1}, \hat{d}^k \right\rangle &= \left\langle \nabla f(x^k) - \nabla f(x^{k+1}), \hat{d}^k \right\rangle + \left\langle \nabla f(x^{k+1}) - m^{k+1}, \hat{d}^k \right\rangle \\ &\leq \|\nabla f(x^k) - \nabla f(x^{k+1})\|_\star \|\hat{d}^k\| + \|\nabla f(x^{k+1}) - m^{k+1}\|_\star \|\hat{d}^k\|.\end{aligned}$$

By Assumption 2, we have $\|\nabla f(x^k) - \nabla f(x^{k+1})\|_\star \leq L\|x^k - x^{k+1}\| = L\gamma^k \|\hat{d}^k\|$. Substituting this in yields

$$\begin{aligned}\left\langle \nabla f(x^k) - m^{k+1}, \hat{d}^k \right\rangle &\leq (L\gamma^k \|\hat{d}^k\|)\|\hat{d}^k\| + \|\hat{d}^k\| \cdot \|\nabla f(x^{k+1}) - m^{k+1}\|_\star \\ &= L\gamma^k \|\hat{d}^k\|^2 + \|\hat{d}^k\| \cdot \|\nabla f(x^{k+1}) - m^{k+1}\|_\star.\end{aligned}$$

Using our bound $\|\hat{d}^k\| \leq 1 + \delta_k$, this becomes

$$\left\langle \nabla f(x^k) - m^{k+1}, \hat{d}^k \right\rangle \leq L\gamma^k (1+\delta_k)^2 + (1+\delta_k)\|\nabla f(x^{k+1}) - m^{k+1}\|_\star.$$

4. Assembling the Final Inequality. We now substitute all the bounded components back into inequality (9):

$$\begin{aligned}f(x^{k+1}) \leq\ & f(x^k) + \gamma^k \left(-\|m^{k+1}\|_\star (1-\delta_k)\right) \\ & + \gamma^k \left(L\gamma^k (1+\delta_k)^2 + (1+\delta_k)\|\nabla f(x^{k+1}) - m^{k+1}\|_\star\right) \\ & + \frac{L}{2}(\gamma^k)^2 (1+\delta_k)^2.\end{aligned}$$

Finally, collecting the terms proportional to $L(\gamma^k)^2(1+\delta_k)^2$ gives the result stated in the lemma. □

B.3.1 Proof of Lemma 1

Proof. The proof bounds the expected deviation of the momentum vector m^{k+1} from the true gradient $\nabla f(x^k)$. We remind the reader that the update rule for the momentum is $m^{k+1} = (1-\alpha_k)m^k + \alpha_k g^k$. We start by expanding the error term recursively, obtaining

$$\begin{aligned}m^{k+1} - \nabla f(x^k) &= (1-\alpha_k)m^k + \alpha_k g^k - \nabla f(x^k) \\ &= (1-\alpha_k)(m^k - \nabla f(x^{k-1})) + \alpha_k (g^k - \nabla f(x^k)) + (1-\alpha_k)(\nabla f(x^{k-1}) - \nabla f(x^k)).\end{aligned}$$

This decomposition separates the error into three distinct components: (1) the decayed error from the previous step, (2) the new stochastic noise, and (3) the gradient drift. By unrolling this recursion from k down to 0, we can express the total error as a sum of its historical components:

$$\begin{aligned}m^{k+1} - \nabla f(x^k) = \left(\prod_{i=0}^{k}(1-\alpha_i)\right)(m^0 - \nabla f(x^0)) &+ \sum_{i=0}^{k}\left(\prod_{j=i+1}^{k}(1-\alpha_j)\right)\alpha_i (g^i - \nabla f(x^i)) \\ &+ \sum_{i=1}^{k}\left(\prod_{j=i}^{k}(1-\alpha_j)\right)(\nabla f(x^{i-1}) - \nabla f(x^i)).\end{aligned}$$

We now take the expectation of the norm and apply the triangle inequality, to get

$$\begin{aligned}\mathbb{E}[\|m^{k+1} - \nabla f(x^k)\|_\star] \leq\ & \left(\prod_{i=0}^{k}(1-\alpha_i)\right)\mathbb{E}[\|m^0 - \nabla f(x^0)\|_\star] \\ & + \mathbb{E}\left[\left\|\sum_{i=0}^{k}\left(\prod_{j=i+1}^{k}(1-\alpha_j)\right)\alpha_i (g^i - \nabla f(x^i))\right\|_\star\right] \\ & + \sum_{i=1}^{k}\left(\prod_{j=i}^{k}(1-\alpha_j)\right)\mathbb{E}[\|\nabla f(x^{i-1}) - \nabla f(x^i)\|_\star].\end{aligned}$$

To obtain the result in the statement of the lemma, we analyze the bound for constant parameters α, γ, δ.

1. Bounding the Initial Error. With the standard initialization $m^0 = g^0$, and using the tower property of expectation along with Assumption 3 and the norm compatibility condition, we have

$$\mathbb{E}[\|m^0 - \nabla f(x^0)\|_\star] = \mathbb{E}[\|g^0 - \nabla f(x^0)\|_\star] \leq \rho\, \mathbb{E}[\|g^0 - \nabla f(x^0)\|_2].$$

By Jensen's inequality, $\mathbb{E}[X] \leq \sqrt{\mathbb{E}[X^2]}$, this is further bounded by

$$\rho\sqrt{\mathbb{E}[\|g^0 - \nabla f(x^0)\|_2^2]} \leq \rho\sigma.$$

2. Bounding the Accumulated Drift. This is the term modified by our inexact LMO. Using Assumption 2 (L-smoothness) and the fact that $\|x^{i-1} - x^i\| = \gamma\|\hat{d}^{i-1}\| \leq \gamma(1+\delta)$, we can bound the sum as

$$\begin{aligned}\sum_{i=1}^{k}(1-\alpha)^{k-i+1}\mathbb{E}[\|\nabla f(x^{i-1}) - \nabla f(x^i)\|_\star] &\leq \sum_{i=1}^{k}(1-\alpha)^{k-i+1}L\gamma(1+\delta) \\ &= L\gamma(1+\delta)\sum_{j=0}^{k-1}(1-\alpha)^{j+1},\end{aligned}$$

where we changed the summation index to $j = i - 1$. This is a geometric series which we can bound by its infinite sum:

$$L\gamma(1+\delta)\sum_{j=0}^{\infty}(1-\alpha)^{j+1} = L\gamma(1+\delta)\frac{1-\alpha}{1-(1-\alpha)} \leq \frac{L\gamma(1+\delta)}{\alpha}.$$

3. Bounding the Accumulated Noise. Let $V_k = \sum_{i=0}^{k}\alpha(1-\alpha)^{k-i}(g^i - \nabla f(x^i))$. The random vectors $(g^i - \nabla f(x^i))$ are zero-mean and independent conditioned on the past. Using the norm compatibility followed by Jensen's inequality, we have

$$\mathbb{E}[\|V_k\|_\star] \leq \rho\,\mathbb{E}[\|V_k\|_2] \leq \rho\sqrt{\mathbb{E}[\|V_k\|_2^2]}.$$

Since the random vectors are independent and zero-mean, the cross-terms vanish in expectation. So, the expected squared norm of the sum is the sum of the expected squared norms:

$$\begin{aligned}\mathbb{E}[\|V_k\|_2^2] &= \sum_{i=0}^{k}\alpha^2(1-\alpha)^{2(k-i)}\mathbb{E}[\|g^i - \nabla f(x^i)\|_2^2] \\ &\leq \sum_{i=0}^{k}\alpha^2(1-\alpha)^{2(k-i)}\sigma^2 = \alpha^2\sigma^2\sum_{j=0}^{k}((1-\alpha)^2)^j.\end{aligned}$$

We bound this geometric series by its infinite sum:

$$\sum_{j=0}^{k}((1-\alpha)^2)^j \leq \sum_{j=0}^{\infty}((1-\alpha)^2)^j = \frac{1}{1-(1-\alpha)^2} = \frac{1}{2\alpha - \alpha^2} = \frac{1}{\alpha(2-\alpha)}.$$

Substituting this back, we get the bound for the accumulated noise:

$$\mathbb{E}[\|V_k\|_\star] \leq \rho\sqrt{\alpha^2\sigma^2\frac{1}{\alpha(2-\alpha)}} = \rho\sqrt{\frac{\alpha\sigma^2}{2-\alpha}} = \frac{\rho\sigma\sqrt{\alpha}}{\sqrt{2-\alpha}}.$$

4. Combining the Bounds. Combining the bounds for the three components gives the final expression. For the simplified steady-state bound (which holds for any k), we combine the infinite-sum bounds for the drift and noise terms with the initial error term, which decays exponentially over time due to the $(1-\alpha)^{k+1}$ factor:

$$\mathbb{E}[\|m^{k+1} - \nabla f(x^k)\|_\star] \leq (1-\alpha)^{k+1}\rho\sigma + \frac{\rho\sigma\sqrt{\alpha}}{\sqrt{2-\alpha}} + \frac{L\gamma(1+\delta)}{\alpha}.$$

This is the result stated for constant parameters in Lemma 1. □

B.3.2 Proof of Theorem 2

Proof. The proof derives the final convergence guarantee by combining the per-iteration progress from the "Inexact Descent Lemma" (Lemma 2) with the bound on the momentum error from Lemma 1.

We begin with the per-iteration guarantee from Lemma 2, specialized for constant parameters γ, α, δ:

$$f(x^{k+1}) \leq f(x^k) - \gamma\|m^{k+1}\|_\star(1-\delta) + \gamma(1+\delta)\|\nabla f(x^{k+1}) - m^{k+1}\|_\star + \frac{3L}{2}\gamma^2(1+\delta)^2.$$

To obtain a guarantee on the true gradient norm, $\|\nabla f(x^{k+1})\|_\star$, we apply the reverse triangle inequality to the main descent term:

$$\|m^{k+1}\|_\star = \|\nabla f(x^{k+1}) - (\nabla f(x^{k+1}) - m^{k+1})\|_\star \geq \|\nabla f(x^{k+1})\|_\star - \|\nabla f(x^{k+1}) - m^{k+1}\|_\star.$$

Substituting this back into the descent inequality (noting that the leading minus sign flips the inequality for the error term) gives

$$-\gamma\|m^{k+1}\|_\star(1-\delta) \leq -\gamma\left(\|\nabla f(x^{k+1})\|_\star - \|\nabla f(x^{k+1}) - m^{k+1}\|_\star\right)(1-\delta).$$

This yields a new per-iteration guarantee directly in terms of the gradient norm:

$$\begin{aligned} f(x^{k+1}) \leq f(x^k) &- \gamma\|\nabla f(x^{k+1})\|_\star(1-\delta) + \gamma(1-\delta)\|\nabla f(x^{k+1}) - m^{k+1}\|_\star \\ &+ \gamma(1+\delta)\|\nabla f(x^{k+1}) - m^{k+1}\|_\star + \frac{3L}{2}\gamma^2(1+\delta)^2. \end{aligned}$$

The coefficients for the two momentum error terms sum to $(1-\delta)+(1+\delta) = 2$. Combining them yields a cleaner expression:

$$f(x^{k+1}) \leq f(x^k) - \gamma\|\nabla f(x^{k+1})\|_\star(1-\delta) + 2\gamma\|\nabla f(x^{k+1}) - m^{k+1}\|_\star + \frac{3L}{2}\gamma^2(1+\delta)^2.$$

Rearranging to isolate the gradient norm, we get

$$\gamma(1-\delta)\|\nabla f(x^{k+1})\|_\star \leq f(x^k) - f(x^{k+1}) + 2\gamma\|\nabla f(x^{k+1}) - m^{k+1}\|_\star + \frac{3L}{2}\gamma^2(1+\delta)^2.$$

We now sum this inequality from $k = 0$ to $K-1$, take the total expectation, and perform an index shift on the summation of the gradient norm ($j = k+1$), obtaining

$$\begin{aligned} \gamma(1-\delta)\sum_{j=1}^{K}\mathbb{E}[\|\nabla f(x^j)\|_\star] \leq \mathbb{E}\left[\sum_{k=0}^{K-1}(f(x^k) - f(x^{k+1}))\right] &+ 2\gamma\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla f(x^{k+1}) - m^{k+1}\|_\star] \\ &+ K\frac{3L}{2}\gamma^2(1+\delta)^2. \end{aligned}$$

The first term on the right-hand side is a telescoping sum bounded by $\Delta^0 = f(x^0) - f^*$.

Bounding the Sum of Momentum Errors. The core of the proof is to bound the sum of momentum errors. We use the triangle inequality to bridge the index gap:

$$\mathbb{E}[\|\nabla f(x^{k+1}) - m^{k+1}\|_\star] \leq \mathbb{E}[\|\nabla f(x^{k+1}) - \nabla f(x^k)\|_\star] + \mathbb{E}[\|\nabla f(x^k) - m^{k+1}\|_\star].$$

The first term (gradient drift) is bounded by $L\gamma(1+\delta)$. The second term is bounded by Lemma 1. We use the slightly looser but simpler bound $\frac{\rho\sigma\sqrt{\alpha}}{\sqrt{2-\alpha}} \leq \rho\sigma\sqrt{\alpha}$ for $\alpha \in (0,1)$. The full error is thus bounded by

$$\mathbb{E}[\|\nabla f(x^{k+1}) - m^{k+1}\|_\star] \leq L\gamma(1+\delta) + (1-\alpha)^{k+1}\rho\sigma + \rho\sigma\sqrt{\alpha} + \frac{L\gamma(1+\delta)}{\alpha}.$$

We now substitute this bound back into the main summation:

$$\begin{aligned} \gamma(1-\delta)\sum_{j=1}^{K}\mathbb{E}[\|\nabla f(x^j)\|_\star] \leq \Delta^0 &+ K\frac{3L\gamma^2(1+\delta)^2}{2} \\ &+ 2\gamma\sum_{k=0}^{K-1}\left(L\gamma(1+\delta) + (1-\alpha)^{k+1}\rho\sigma + \rho\sigma\sqrt{\alpha} + \frac{L\gamma(1+\delta)}{\alpha}\right). \end{aligned}$$

Evaluating the summation over k for each component gives

$$\gamma(1-\delta)\sum_{j=1}^{K}\mathbb{E}[\|\nabla f(x^j)\|_\star] \le \Delta^0 + K\frac{3L\gamma^2(1+\delta)^2}{2} + 2KL\gamma^2(1+\delta) + 2\gamma\left(\sum_{k=0}^{K-1}(1-\alpha)^{k+1}\rho\sigma\right) + 2K\gamma\rho\sigma\sqrt{\alpha} + \frac{2KL\gamma^2(1+\delta)}{\alpha}.$$

For the geometric series, we have $\sum_{k=0}^{K-1}(1-\alpha)^{k+1} \le \sum_{j=1}^{\infty}(1-\alpha)^j = \frac{1-\alpha}{\alpha} \le \frac{1}{\alpha}$. Plugging this in gives

$$\gamma(1-\delta)\sum_{j=1}^{K}\mathbb{E}[\|\nabla f(x^j)\|_\star] \le \Delta^0 + K\frac{3L\gamma^2(1+\delta)^2}{2} + 2KL\gamma^2(1+\delta) + \frac{2\gamma\rho\sigma}{\alpha} + 2K\gamma\rho\sigma\sqrt{\alpha} + \frac{2KL\gamma^2(1+\delta)}{\alpha}.$$

Finally, dividing by $K\gamma(1-\delta)$ yields the average bound for the gradient norm. Grouping the terms by their dependency gives the final stated result in Theorem 2. □

B.3.3 Proof of Corollary 3

The proof consists of two parts. First, we derive the asymptotically optimal choices for the step size γ and momentum parameter α by minimizing the convergence upper bound from Theorem 2. Second, we substitute these optimal parameters back into the bound to obtain the final convergence rate.

1. Optimal Parameter Derivation. Our goal is to find $\gamma > 0$ and $\alpha \in (0,1)$ that minimize the expression $\mathcal{E}(\gamma,\alpha)$:

$$\mathcal{E}(\gamma,\alpha) = \frac{\Delta^0}{K\gamma(1-\delta)} + \frac{2\rho\sigma}{\alpha K(1-\delta)} + \frac{2\rho\sigma\sqrt{\alpha}}{1-\delta} + \frac{L\gamma(1+\delta)}{1-\delta}\left(\frac{3(1+\delta)}{2}+2\right) + \frac{2L\gamma(1+\delta)}{\alpha(1-\delta)}.$$

To simplify the algebraic manipulation, we define the following constants:

- $C_1 = \frac{\Delta^0}{K(1-\delta)}$,
- $C_2 = \frac{2\rho\sigma}{K(1-\delta)}$,
- $C_3 = \frac{2\rho\sigma}{1-\delta}$,
- $C_4 = \frac{L(1+\delta)}{1-\delta}\left(\frac{3(1+\delta)}{2}+2\right)$,
- $C_5 = \frac{2L(1+\delta)}{1-\delta}$.

Hence, the expression to minimize is $\mathcal{E}(\gamma,\alpha) = \frac{C_1}{\gamma} + \frac{C_2}{\alpha} + C_3\sqrt{\alpha} + C_4\gamma + \frac{C_5\gamma}{\alpha}$. We find the optimal parameters by setting the partial derivatives with respect to γ and α to zero, yielding the system of equations

$$\frac{\partial\mathcal{E}}{\partial\gamma} = -\frac{C_1}{\gamma^2} + C_4 + \frac{C_5}{\alpha} = 0, \tag{10}$$

$$\frac{\partial\mathcal{E}}{\partial\alpha} = -\frac{C_2}{\alpha^2} + \frac{C_3}{2\sqrt{\alpha}} - \frac{C_5\gamma}{\alpha^2} = 0. \tag{11}$$

An exact closed-form solution is intractable. However, we are interested in the asymptotic behavior as $K \to \infty$, where we expect $\gamma^*, \alpha^* \to 0$. The constants scale as: $C_1 \propto 1/K$, $C_2 \propto 1/K$, while C_3, C_4, C_5 are constants with respect to K.

We now solve the system in this asymptotic regime. From (10), as $\alpha \to 0$, the term C_5/α dominates the constant C_4. Thus, the equation asymptotically behaves as

$$\gamma^2 \approx \frac{C_1\alpha}{C_5}. \tag{12}$$

From (11), since $C_2 \propto 1/K$ and we expect γ to decay slower than $1/K$, the term C_2 is asymptotically negligible compared to $C_5\gamma$. This gives:

$$\frac{C_3}{2}\alpha^{3/2} \approx C_5\gamma. \tag{13}$$

From (13), we express γ in terms of α: $\gamma \approx \frac{C_3}{2C_5}\alpha^{3/2}$. Substituting this into (12):

$$\left(\frac{C_3}{2C_5}\alpha^{3/2}\right)^2 \approx \frac{C_1\alpha}{C_5} \implies \frac{C_3^2}{4C_5^2}\alpha^3 \approx \frac{C_1\alpha}{C_5}.$$

Solving for α^2 (assuming $\alpha \neq 0$) gives $\alpha^2 \approx \frac{4C_1C_5}{C_3^2}$. Now, we substitute back the full definitions of the constants:

$$(\alpha^*)^2 = \frac{4 \cdot \frac{\Delta^0}{K(1-\delta)} \cdot \frac{2L(1+\delta)}{1-\delta}}{\left(\frac{2\rho\sigma}{1-\delta}\right)^2} = \frac{\frac{8\Delta^0 L(1+\delta)}{K(1-\delta)^2}}{\frac{4(\rho\sigma)^2}{(1-\delta)^2}} = \frac{2\Delta^0 L(1+\delta)}{K(\rho\sigma)^2}.$$

This yields the optimal momentum parameter:

$$\alpha^* = \frac{\sqrt{2\Delta^0 L(1+\delta)}}{\sqrt{K}\rho\sigma}.$$

To find the optimal step size γ^*, we use the relation $\gamma^2 \approx (C_1\alpha)/C_5$:

$$\begin{aligned}(\gamma^*)^2 &= \frac{\Delta^0}{K(1-\delta)} \cdot \left(\frac{\sqrt{2\Delta^0 L(1+\delta)}}{\sqrt{K}\rho\sigma}\right) \cdot \frac{1-\delta}{2L(1+\delta)} = \frac{\Delta^0\sqrt{2\Delta^0 L(1+\delta)}}{K^{3/2}\rho\sigma \cdot 2L(1+\delta)} \\ &= \frac{(\Delta^0)^{3/2}\sqrt{2L}\sqrt{1+\delta}}{K^{3/2}\rho\sigma \cdot 2L(1+\delta)} = \frac{(\Delta^0)^{3/2}}{K^{3/2}\sqrt{2L}\rho\sigma\sqrt{1+\delta}}.\end{aligned}$$

Taking the square root gives the optimal step size

$$\gamma^* = \frac{(\Delta^0)^{3/4}}{2^{1/4}K^{3/4}L^{1/4}(\rho\sigma)^{1/2}(1+\delta)^{1/4}}.$$

2. Derivation of the Final Convergence Rate. We now substitute the asymptotically optimal parameters, γ^* and α^*, back into the full five-term convergence bound $\mathcal{E}(\gamma, \alpha)$ to determine the best achievable convergence rate. The bound is the sum of five distinct terms:

$$\mathcal{E}(\gamma^*, \alpha^*) = \underbrace{\frac{C_1}{\gamma^*}}_{\text{Term 1}} + \underbrace{\frac{C_2}{\alpha^*}}_{\text{Term 2}} + \underbrace{C_3\sqrt{\alpha^*}}_{\text{Term 3}} + \underbrace{C_4\gamma^*}_{\text{Term 4}} + \underbrace{\frac{C_5\gamma^*}{\alpha^*}}_{\text{Term 5}}.$$

We compute the explicit value of each term to verify the dominant rate and analyze the contribution of the non-dominant, higher-order terms.

Dominant Terms ($\mathcal{O}(1/K^{1/4})$). These are the terms that dictate the asymptotic convergence rate.

Term 1: The Initial Gap Component.

$$\begin{aligned}\text{Term 1} = \frac{C_1}{\gamma^*} &= \frac{\Delta^0}{K(1-\delta)} \cdot (\gamma^*)^{-1} = \frac{\Delta^0}{K(1-\delta)} \cdot \left(\frac{2^{1/4}K^{3/4}L^{1/4}(\rho\sigma)^{1/2}(1+\delta)^{1/4}}{(\Delta^0)^{3/4}}\right) \\ &= \frac{2^{1/4}(\Delta^0)^{1/4}L^{1/4}(\rho\sigma)^{1/2}(1+\delta)^{1/4}}{K^{1/4}(1-\delta)}.\end{aligned}$$

Term 3: The Steady-State Variance Component.

$$\begin{aligned}\text{Term 3} = C_3\sqrt{\alpha^*} &= \frac{2\rho\sigma}{1-\delta} \cdot \left(\frac{\sqrt{2\Delta^0 L(1+\delta)}}{\sqrt{K}\rho\sigma}\right)^{1/2} = \frac{2\rho\sigma}{1-\delta} \cdot \frac{(2\Delta^0 L(1+\delta))^{1/4}}{K^{1/4}(\rho\sigma)^{1/2}} \\ &= \frac{2 \cdot 2^{1/4}(\rho\sigma)^{1/2}(L\Delta^0(1+\delta))^{1/4}}{K^{1/4}(1-\delta)} = 2 \cdot (\text{Term 1}).\end{aligned}$$

Term 5: The Drift/Interaction Component.

$$\text{Term 5} = \frac{C_5\gamma^*}{\alpha^*} = \frac{2L(1+\delta)}{1-\delta}\cdot\frac{\gamma^*}{\alpha^*} = \frac{2L(1+\delta)}{1-\delta}\cdot\frac{C_3}{2C_5}\sqrt{\alpha^*} = \frac{C_3\sqrt{\alpha^*}}{1-\delta}\cdot\frac{L(1+\delta)}{C_5}$$
$$= \frac{1}{2}C_3\sqrt{\alpha^*} = \text{Term 1}.$$

The asymptotic balancing ensures Term 1, Term 3, and Term 5 are all of the same order, $\mathcal{O}(1/K^{1/4})$.

Higher-Order Terms. The remaining terms decay faster and are asymptotically negligible.

Term 2: The Decaying Variance Component ($\mathcal{O}(1/K^{1/2})$).

$$\text{Term 2} = \frac{C_2}{\alpha^*} = \frac{2\rho\sigma}{K(1-\delta)}\cdot\left(\frac{\sqrt{K}\rho\sigma}{\sqrt{2\Delta^0 L(1+\delta)}}\right) = \frac{\sqrt{2}(\rho\sigma)^2}{\sqrt{K}(1-\delta)\sqrt{\Delta^0 L(1+\delta)}}.$$

Term 4: The Step Size Bias Component ($\mathcal{O}(1/K^{3/4})$).

$$\text{Term 4} = C_4\gamma^* = \frac{L(1+\delta)}{1-\delta}\left(\frac{3(1+\delta)}{2}+2\right)\cdot\frac{(\Delta^0)^{3/4}}{2^{1/4}K^{3/4}L^{1/4}(\rho\sigma)^{1/2}(1+\delta)^{1/4}} = \frac{(7+3\delta)L^{3/4}(\Delta^0)^{3/4}(1+\delta)^{3/4}}{2^{5/4}K^{3/4}(1-\delta)(\rho\sigma)^{1/2}}.$$

The Full Bound. Combining all terms, the full optimized convergence bound is

$$\mathcal{E}(\gamma^*,\alpha^*) = \underbrace{\frac{2^{9/4}(\Delta^0)^{1/4}(\rho\sigma)^{1/2}(L(1+\delta))^{1/4}}{K^{1/4}(1-\delta)}}_{\text{Dominant Part: } \mathcal{O}(K^{-1/4})} + \underbrace{\frac{\sqrt{2}(\rho\sigma)^2}{K^{1/2}(1-\delta)\sqrt{\Delta^0 L(1+\delta)}}}_{\text{Higher-Order: } \mathcal{O}(K^{-1/2})} + \underbrace{\frac{(7+3\delta)L^{3/4}(\Delta^0)^{3/4}(1+\delta)^{3/4}}{2^{5/4}K^{3/4}(1-\delta)(\rho\sigma)^{1/2}}}_{\text{Higher-Order: } \mathcal{O}(K^{-3/4})}.$$

This confirms that the convergence rate is $\mathcal{O}(1/K^{1/4})$ with the precise dependence of the dominant part on all problem parameters, matching the result stated in Corollary 3.

B.4 Convergence with time-varying parameters

The analysis with constant parameters provides crucial insights into the fundamental trade-offs of the algorithm. However, the optimal choices for these constant parameters, γ^* and α^*, depend on problem-specific constants such as the smoothness L and the initial sub-optimality Δ^0, as well as the total iteration budget K. In practice, these quantities are typically unknown, making this "optimal" tuning infeasible.

To develop a more practical and robust guarantee, we now analyze the algorithm with a pre-defined, *time-varying* schedule for the step size and momentum parameters. This approach is "parameter-agnostic," meaning the schedules are chosen based only on the iteration counter and do not require prior knowledge of the problem's characteristics. The primary goal of this analysis is to prove that the algorithm minimizes the expected gradient (i.e., $\mathbb{E}[\|\nabla f(x^k)\|_\star] \to 0$) for any problem satisfying our general assumptions, without any manual tuning of the learning rates with respect to L or K.

The cornerstone of this analysis is a general bound on the expected momentum error that holds for any valid time-varying schedule.

Lemma 3 (Momentum Error Bound for Time-Varying Parameters)**.** *Let Assumptions 2,3, and the norm compatibility condition hold. Let the sequence $\{x^k\}$ be generated by Algorithm 1 with LMO errors satisfying Assumption 1. The expected momentum error at iteration T is bounded by*

$$E_{T-1} = \mathbb{E}[\|m^T - \nabla f(x^{T-1})\|_\star] \le \left(\prod_{t=0}^{T-1}(1-\alpha_t)\right)\rho\sigma + \left(\sum_{t=0}^{T-1}\alpha_t^2\prod_{\tau=t+1}^{T-1}(1-\alpha_\tau)\right)^{1/2}\rho\sigma$$
$$+ L\sum_{t=0}^{T-1}\gamma^t(1+\delta_t)\prod_{\tau=t+1}^{T-1}(1-\alpha_\tau).$$

Proof. The proof proceeds by unrolling the one-step recursion for the momentum error. Let $e_k := m^{k+1} - \nabla f(x^k)$, $v_k := g^k - \nabla f(x^k)$ be the stochastic noise, and $s_{k-1} := \nabla f(x^{k-1}) - \nabla f(x^k)$ be the gradient drift. The error recursion is

$$e_k = (1-\alpha_k)e_{k-1} + \alpha_k v_k + (1-\alpha_k)s_{k-1}.$$

Unrolling this relationship from $k = T-1$ down to 0 gives the expression for the total error:

$$e_{T-1} = \left(\prod_{t=0}^{T-1}(1-\alpha_t)\right) e_{-1} + \sum_{t=0}^{T-1}\left(\prod_{\tau=t+1}^{T-1}(1-\alpha_\tau)\right)(\alpha_t v_t + (1-\alpha_t)s_{t-1}),$$

where we define $s_{-1} = 0$ and $e_{-1} = m^0 - \nabla f(x^0)$. For the initialization $m^0 = g^0$, we can bound $\mathbb{E}[\|e_{-1}\|_\star] \le \rho\sigma$. Taking the expectation of the norm and applying the triangle inequality yields

$$\begin{aligned}\mathbb{E}[\|e_{T-1}\|_\star] \le \left(\prod_{t=0}^{T-1}(1-\alpha_t)\right)\rho\sigma + \mathbb{E}\left[\left\|\sum_{t=0}^{T-1}\alpha_t v_t \prod_{\tau=t+1}^{T-1}(1-\alpha_\tau)\right\|_\star\right] \\ + \mathbb{E}\left[\left\|\sum_{t=0}^{T-1}(1-\alpha_t)s_{t-1}\prod_{\tau=t+1}^{T-1}(1-\alpha_\tau)\right\|_\star\right].\end{aligned}$$

We now bound the two summation terms separately.

1. Bounding the Accumulated Noise Term. We apply the norm compatibility condition and then Jensen's inequality, to have

$$\mathbb{E}\left[\left\|\sum_{t=0}^{T-1}\alpha_t v_t \dots\right\|_\star\right] \le \rho\left(\mathbb{E}\left[\left\|\sum_{t=0}^{T-1}\alpha_t v_t \prod_{\tau=t+1}^{T-1}(1-\alpha_\tau)\right\|_2^2\right]\right)^{1/2}.$$

The stochastic noise vectors v_t are zero-mean and independent conditioned on the past. Therefore, the cross-terms in the squared sum vanish in expectation, yielding

$$\mathbb{E}\left[\left\|\sum\dots\right\|_2^2\right] = \sum_{t=0}^{T-1}\alpha_t^2\left(\prod_{\tau=t+1}^{T-1}(1-\alpha_\tau)\right)^2\mathbb{E}[\|v_t\|_2^2] \le \sigma^2\sum_{t=0}^{T-1}\alpha_t^2\left(\prod_{\tau=t+1}^{T-1}(1-\alpha_\tau)\right)^2.$$

Using the upper bound $(1-x)^2 \le (1-x)$ for any $x \in [0,1]$, we have

$$\mathbb{E}\left[\left\|\sum\dots\right\|_2^2\right] \le \sigma^2\sum_{t=0}^{T-1}\alpha_t^2\prod_{\tau=t+1}^{T-1}(1-\alpha_\tau).$$

Substituting this back yields the final bound for the noise term:

$$\text{Noise Term} \le \rho\sigma\left(\sum_{t=0}^{T-1}\alpha_t^2\prod_{\tau=t+1}^{T-1}(1-\alpha_\tau)\right)^{1/2}.$$

2. Bounding the Accumulated Drift Term. We apply the triangle inequality for sums, Assumption 2, and the step-size bound $\|x^t - x^{t-1}\| \le \gamma^{t-1}(1+\delta_{t-1})$, to obtain

$$\begin{aligned}\mathbb{E}\left[\left\|\sum(1-\alpha_t)s_{t-1}\dots\right\|_\star\right] &\le \sum_{t=0}^{T-1}(1-\alpha_t)\mathbb{E}[\|s_{t-1}\|_\star]\prod_{\tau=t+1}^{T-1}(1-\alpha_\tau) \\ &\le L\sum_{t=0}^{T-1}(1-\alpha_t)\gamma^{t-1}(1+\delta_{t-1})\prod_{\tau=t+1}^{T-1}(1-\alpha_\tau).\end{aligned}$$

Using the bound $(1-\alpha_t) \le 1$ and re-indexing the summation gives the simplified form

$$\text{Drift Term} \le L \sum_{t=0}^{T-1} \gamma^t (1+\delta_t) \prod_{\tau=t+1}^{T-1} (1-\alpha_\tau).$$

3. Combining the Bounds. Assembling the bounds for the three components gives the inequality stated in the lemma. □

Before analyzing the convergence for a specific time-varying schedule, we first establish a general convergence bound that holds for any choice of sequences $\{\gamma^k\}$ and $\{\alpha_k\}$. This theorem is the stochastic analogue of the deterministic result in Theorem 1 and serves as the starting point for all subsequent rate analysis.

Lemma 4 (General Convergence Bound)**.** *Let Assumptions 1, 2, 3, and the norm compatibility condition hold. The sequence of iterates $\{x^k\}$ generated by Algorithm 1 with time-varying parameters satisfies the following inequality for any $K \ge 1$:*

$$\sum_{k=0}^{K-1} \gamma^k (1-\delta_k) \mathbb{E}[\|\nabla f(x^{k+1})\|_\star] \le \Delta^0 + \frac{3L}{2} \sum_{k=0}^{K-1} (\gamma^k)^2 (1+\delta_k)^2 + 2L \sum_{k=0}^{K-1} (\gamma^k)^2 (1+\delta_k) + 2 \sum_{k=0}^{K-1} \gamma^k \mathbb{E}[\|\nabla f(x^k) - m^{k+1}\|_\star],$$

where $\Delta^0 = f(x^0) - f^*$.

Proof. The proof begins with the per-iteration guarantee from the Inexact Descent Lemma (Lemma 2). To obtain a guarantee on the true gradient norm, we apply the reverse triangle inequality to the main descent term, $\|m^{k+1}\|_\star \ge \|\nabla f(x^{k+1})\|_\star - \|\nabla f(x^{k+1}) - m^{k+1}\|_\star$. Substituting this into the descent lemma's bound gives

$$f(x^{k+1}) \le f(x^k) - \gamma^k \|\nabla f(x^{k+1})\|_\star (1-\delta_k) + \gamma^k (1-\delta_k) \|\nabla f(x^{k+1}) - m^{k+1}\|_\star + \gamma^k (1+\delta_k) \|\nabla f(x^{k+1}) - m^{k+1}\|_\star + \frac{3L}{2} (\gamma^k)^2 (1+\delta_k)^2.$$

Combining the two momentum error terms (whose coefficients sum to 2) and rearranging yields

$$\gamma^k (1-\delta_k) \|\nabla f(x^{k+1})\|_\star \le f(x^k) - f(x^{k+1}) + 2\gamma^k \|\nabla f(x^{k+1}) - m^{k+1}\|_\star + \frac{3L}{2} (\gamma^k)^2 (1+\delta_k)^2.$$

We now sum this inequality from $k=0$ to $K-1$ and take the total expectation:

$$\sum_{k=0}^{K-1} \gamma^k (1-\delta_k) \mathbb{E}[\|\nabla f(x^{k+1})\|_\star] \le \sum_{k=0}^{K-1} \mathbb{E}[f(x^k) - f(x^{k+1})] + 2 \sum_{k=0}^{K-1} \gamma^k \mathbb{E}[\|\nabla f(x^{k+1}) - m^{k+1}\|_\star] + \sum_{k=0}^{K-1} \frac{3L}{2} (\gamma^k)^2 (1+\delta_k)^2.$$

The first term on the right-hand side is a telescoping sum bounded by Δ^0. The core of the proof is to decompose the sum of the momentum errors using the triangle inequality to bridge the index gap

$$2 \sum_{k=0}^{K-1} \gamma^k \mathbb{E}[\|\nabla f(x^{k+1}) - m^{k+1}\|_\star] \le 2 \sum_{k=0}^{K-1} \gamma^k \left(\mathbb{E}[\|\nabla f(x^{k+1}) - \nabla f(x^k)\|_\star] + \mathbb{E}[\|\nabla f(x^k) - m^{k+1}\|_\star] \right).$$

This decomposition separates the error into two distinct components.

1. Bounding the Gradient Drift Component. The first part of this sum is bounded using Assumption 2 (L-smoothness) and the property of the update step, $\|x^{k+1} - x^k\| = \gamma^k \|\hat{d}^k\| \le \gamma^k (1+\delta_k)$, as

$$2 \sum_{k=0}^{K-1} \gamma^k \mathbb{E}[\|\nabla f(x^{k+1}) - \nabla f(x^k)\|_\star] \le 2 \sum_{k=0}^{K-1} \gamma^k \left(L\gamma^k (1+\delta_k) \right) = 2L \sum_{k=0}^{K-1} (\gamma^k)^2 (1+\delta_k).$$

This term represents the accumulated error caused by the gradient changing between steps.

2. The Momentum Lag Component. The second part of the sum is the total accumulated momentum error, which is the final term in the theorem statement:

$$2\sum_{k=0}^{K-1}\gamma^k\mathbb{E}[\|\nabla f(x^k)-m^{k+1}\|_\star].$$

Assembling the Final Bound. Combining all parts, we arrive at the final inequality stated in the theorem. This result cleanly separates the total progress (LHS) from the initial sub-optimality gap (Δ^0), the cost from the step size's magnitude (the two $\sum(\gamma^k)^2$ terms), and the accumulated momentum lag error arising from momentum and stochastic noise. □

We now use the "General Convergence Bound" to derive an explicit convergence rate for a specific, parameter-agnostic schedule. The choice of the time-varying step size and momentum parameter is critical for ensuring that the accumulated error terms are properly controlled, leading to a guarantee of convergence. We adopt a schedule inspired by the analysis in Yang et al. (2023), which has been proven effective for this type of recursive error structure.

Theorem 4 (Convergence with Time-Varying Parameters)**.** *Let Assumptions 1, 2, 3, and the norm compatibility condition hold, with a uniform inexactness bound $\delta_k \le \delta < 1$. If we choose the time-varying step size and momentum parameters for $k \ge 0$ as*

$$\gamma^k = \frac{\gamma}{(k+1)^{3/4}} \quad \textit{and} \quad \alpha_k = \frac{\alpha}{(k+1)^{1/2}},$$

for some user-chosen constants $\gamma > 0$ and $\alpha \in (0,1]$, then after K iterations, the minimum expected gradient norm is bounded as

$$\min_{0\le j<K}\mathbb{E}[\|\nabla f(x^{j+1})\|_\star] = \mathcal{O}\left(\frac{\Delta^0 + L\gamma^2((1+\delta)^2+(1+\delta))}{(1-\delta)\gamma K^{1/4}}\right) + \mathcal{O}\left(\frac{\left(\rho\sigma\sqrt{\alpha}+L\gamma(1+\delta)\alpha^{-1}\right)\log K}{(1-\delta)K^{1/4}}\right).$$

Proof. The proof starts from the General Convergence Bound (Theorem 4). After rearranging, it provides a bound on the minimum expected gradient norm:

$$\left(\min_{0\le j<K}\mathbb{E}[\|\nabla f(x^{j+1})\|_\star]\right)\sum_{k=0}^{K-1}\gamma^k(1-\delta_k) \le \Delta^0 + \frac{3L}{2}\sum_{k=0}^{K-1}(\gamma^k)^2(1+\delta_k)^2 + 2L\sum_{k=0}^{K-1}(\gamma^k)^2(1+\delta_k) + 2\sum_{k=0}^{K-1}\gamma^k\mathbb{E}[\|\nabla f(x^k)-m^{k+1}\|_\star].$$

We analyze each term in this expression under the chosen parameter schedule and the uniform bound $\delta_k \le \delta$.

1. Bounding the Sums of Step Sizes. With the schedule $\gamma^k = \gamma/(k+1)^{3/4}$, the sums have the following behavior. The progress-weighting sum on the LHS is bounded below by an integral for $K \ge 1$:

$$\sum_{k=0}^{K-1}\gamma^k(1-\delta_k) \ge \gamma(1-\delta)\sum_{k=0}^{K-1}\frac{1}{(k+1)^{3/4}} \ge 4\gamma(1-\delta)((K+1)^{1/4}-1).$$

The squared step size sums on the RHS correspond to a convergent p-series (since the exponent $p = 3/2 > 1$) and are bounded by the constant $\zeta(3/2) = \sum_{k=1}^{\infty}k^{-3/2}$ for any K, where ζ is the Riemann zeta function. Therefore, the two terms are bounded by $\frac{3L}{2}\gamma^2\zeta(3/2)(1+\delta)^2$ and $2L\gamma^2\zeta(3/2)(1+\delta)$ respectively.

2. Bounding the Accumulated Momentum Error. The momentum lag component, $2\sum_{k=0}^{K-1}\gamma^k\mathbb{E}[\|\nabla f(x^k)-m^{k+1}\|_\star]$, is bounded by applying the results of Lemma 3. The structure of the bound

in that lemma is of the form addressed by Lemma 3 in Yang et al. (2023). Applying their result to the two sum-of-products terms in our bound, with our chosen schedules, yields

$$\sum_{k=0}^{K-1} \gamma^k \mathbb{E}[\|\nabla f(x^k) - m^{k+1}\|_\star] \leq \left(C_1 \rho\sigma\gamma\sqrt{\alpha} + C_2 L(1+\delta)\frac{\gamma^2}{\alpha} \right) \log(K+1),$$

where C_1, C_2 are universal numerical constants. This shows that the total accumulated error from momentum and noise grows only logarithmically with the number of iterations.

3. Assembling the Final Bound. Substituting these explicit bounds back into the main inequality:

$$\left(\min_{j<K} \mathbb{E}[\|\nabla f(x^{j+1})\|_\star] \right) \cdot 4\gamma(1-\delta)(K^{1/4}-1) \leq \Delta^0 + L\gamma^2\zeta(3/2)\left(\frac{3}{2}(1+\delta)^2 + 2(1+\delta) \right) + 2\left(C_1\rho\sigma\gamma\sqrt{\alpha} + C_2 L(1+\delta)\frac{\gamma^2}{\alpha} \right)\log(K+1).$$

Dividing by $4\gamma(1-\delta)(K^{1/4}-1)$ gives the final expression for the convergence rate. For large K, the term $(K+1)^{1/4} - 1 \approx K^{1/4}$. This leads to the final bound stated in the theorem, where the dependence on all parameters, including the user-chosen schedule constants γ and α, is made explicit. □

C EXTENSIONS

C.1 (L^0, L^1)-smoothness

We now extend our analysis from the standard L-smooth setting to the more general and realistic (L^0, L^1)-smoothness model, as defined in Riabinin et al. (2025).

Assumption 4 (Local (L^0, L^1)-Smoothness)**.** *The function f is continuously differentiable and is bounded below by $f^* > -\infty$. We assume that f is locally (L^0, L^1)-smooth along the optimization trajectory. Specifically, for any pair of consecutive iterates x^k and x^{k+1} generated by the algorithm, the following inequality holds:*

$$f(x^{k+1}) \leq f(x^k) + \langle \nabla f(x^k), x^{k+1} - x^k \rangle + \frac{L^0 + L^1\|\nabla f(x^k)\|_\star}{2}\|x^{k+1} - x^k\|^2.$$

This local version is sufficient for the analysis of descent methods where the step sizes are bounded, as is the case here. For clarity, we do not explicitly track the radius of the local region, as it is implicitly defined by the maximum possible step size.

Theorem 5 (Iteration Complexity for (L^0, L^1)-Smoothness)**.** *Let the function f satisfy the local (L^0, L^1)-smoothness property (Assumption 4). Consider the deterministic method with an inexact LMO satisfying a uniform error bound $\delta_k \leq \delta < 1$ (Assumption 1), and using the adaptive step size*

$$\gamma_\star^k = \frac{\|\nabla f(x^k)\|_\star (1-\delta_k)}{(L^0 + L^1\|\nabla f(x^k)\|_\star)(1+\delta_k)^2}.$$

To guarantee finding an iterate x^k such that $\|\nabla f(x^k)\|_\star \leq \varepsilon$ after at most K iterations, it is sufficient to run the algorithm for a number of iterations K satisfying

$$K \geq \frac{2\Delta^0(1+\delta)^2}{(1-\delta)^2}\left(\frac{L^0}{\varepsilon^2} + \frac{L^1}{\varepsilon} \right),$$

where $\Delta^0 = f(x^0) - f^*$.

Proof. The proof derives the iteration complexity for the deterministic method under (L^0, L^1)-smoothness by using a step size that is chosen optimally at each iteration.

1. Per-Iteration Descent Inequality. Our analysis begins with the generalized smoothness inequality for (L^0, L^1)-smooth functions, which holds for any step $x^{k+1} = x^k + \gamma^k \hat{d}^k$:

$$f(x^{k+1}) \leq f(x^k) + \langle \nabla f(x^k), x^{k+1} - x^k \rangle + \frac{L^0 + L^1\|\nabla f(x^k)\|_\star}{2}\|x^{k+1} - x^k\|^2.$$

As established in the proof of Theorem 1, we can bound the terms involving the inexact direction $\hat{d}^k$ as

- $\|\gamma^k \hat{d}^k\|^2 \leq (\gamma^k)^2(1+\delta_k)^2$.
- $\left\langle \nabla f(x^k), \hat{d}^k \right\rangle \leq -\|\nabla f(x^k)\|_\star(1-\delta_k)$.

Substituting these into the smoothness inequality gives the per-iteration guarantee

$$f(x^{k+1}) \leq f(x^k) - \gamma^k \|\nabla f(x^k)\|_\star(1-\delta_k) + \frac{L^0 + L^1\|\nabla f(x^k)\|_\star}{2}(\gamma^k)^2(1+\delta_k)^2. \tag{14}$$

2. Optimal Step Size and Resulting Descent. The right-hand side of inequality (14) is a quadratic function of the step size γ^k. To maximize the guaranteed descent at each step, we choose the γ^k that minimizes this expression. The optimal choice is given by

$$\gamma_\star^k = \frac{\|\nabla f(x^k)\|_\star(1-\delta_k)}{(L^0 + L^1\|\nabla f(x^k)\|_\star)(1+\delta_k)^2}.$$

We now substitute this optimal step size back into the descent inequality (14):

$$f(x^{k+1}) \leq f(x^k) - \frac{(\|\nabla f(x^k)\|_\star(1-\delta_k))^2}{2(L^0 + L^1\|\nabla f(x^k)\|_\star)(1+\delta_k)^2}.$$

3. Global Bound and Iteration Complexity. Summing the final descent inequality from $k=0$ to $K-1$ gives the global progress:

$$\sum_{k=0}^{K-1} \frac{(\|\nabla f(x^k)\|_\star(1-\delta_k))^2}{2(L^0 + L^1\|\nabla f(x^k)\|_\star)(1+\delta_k)^2} \leq \sum_{k=0}^{K-1}(f(x^k) - f(x^{k+1})) \leq \Delta^0.$$

To derive a meaningful convergence bound from this sum, we define a progress function $\phi(t)$ that captures the contribution for a gradient norm of magnitude $t = \|\nabla f(x^k)\|_\star$:

$$\phi(t) := \frac{t^2}{L^0 + L^1 t}.$$

This function is monotonically increasing for $t \geq 0$. Assuming a uniform inexactness bound $\delta_k \leq \delta < 1$, the summed inequality can be rewritten as:

$$\frac{(1-\delta)^2}{2(1+\delta)^2} \sum_{k=0}^{K-1} \phi(\|\nabla f(x^k)\|_\star) \leq \Delta^0.$$

Let $g_{\min} = \min_{0\leq j<K} \|\nabla f(x^j)\|_\star$. Since $\phi(t)$ is increasing, we can lower-bound the sum by $K\phi(g_{\min})$:

$$\frac{K(1-\delta)^2}{2(1+\delta)^2}\phi(g_{\min}) \leq \Delta^0 \implies \phi(g_{\min}) \leq \frac{2\Delta^0(1+\delta)^2}{K(1-\delta)^2}.$$

To derive the iteration complexity, we require that the algorithm finds an iterate with gradient norm at most ε, i.e., $g_{\min} \leq \varepsilon$. This is guaranteed if $\phi(\varepsilon)$ satisfies the above bound:

$$\phi(\varepsilon) \leq \frac{2\Delta^0(1+\delta)^2}{K(1-\delta)^2}.$$

Solving for K gives the required number of iterations:

$$K \geq \frac{2\Delta^0(1+\delta)^2}{\phi(\varepsilon)(1-\delta)^2}.$$

Substituting the definition of $\phi(\varepsilon) = \frac{\varepsilon^2}{L^0+L^1\varepsilon}$ yields the final complexity stated in the theorem. □

C.2 Layer-wise analysis

We now extend the analysis to the practical, multi-layered setting considered in Riabinin et al. (2025). This framework is essential for modern deep learning, where models are composed of distinct blocks or layers with heterogeneous geometries. We assume that the parameter vector X is a concatenation of p blocks, $X = [X_1, \ldots, X_p]$, where each layer X_i belongs to its own vector space $\mathcal{X}_i$ equipped with an inner product $\langle \cdot, \cdot \rangle_{(i)}$ and norm $\| \cdot \|_{(i)}$. Each layer is updated independently via its own LMO.

This framework allows us to model a realistic scenario where not only the smoothness but also the LMO inexactness varies across layers. For example, the LMO for a simple bias vector might be solved almost exactly (δ_i is small), while the LMO for a large transformer attention block might have a larger error (δ_i is large).

Assumption 5 (Layer-Wise Smoothness)**.** *The function f is smooth with respect to each layer i with a constant L_i. Specifically, for any iterates X^k and X^{k+1}, the following holds:*

$$f(X^{k+1}) \leq f(X^k) + \sum_{i=1}^{p} \left[\left\langle \nabla_i f(X^k), X_i^{k+1} - X_i^k \right\rangle_{(i)} + \frac{L_i}{2} \| X_i^{k+1} - X_i^k \|_{(i)}^2 \right].$$

We first derive a general bound that holds for any choice of layer-specific, time-varying step sizes $\gamma_{k,i}$ and inexactness levels $\delta_{k,i}$.

Theorem 6 (General Convergence Bound for Layer-Wise Method)**.** *Let Assumption 5 hold. Let the layer-wise inexact LMO for each layer i satisfy Assumption 1 with error $\delta_{k,i} < 1$. Then, for any sequence of layer-specific step sizes $\gamma_{k,i} > 0$, the following bound holds after K iterations:*

$$\sum_{k=0}^{K-1} \sum_{i=1}^{p} \gamma_{k,i} \| \nabla_i f(X^k) \|_{(i)\star} (1 - \delta_{k,i}) \leq \Delta^0 + \frac{1}{2} \sum_{k=0}^{K-1} \sum_{i=1}^{p} L_i (\gamma_{k,i})^2 (1 + \delta_{k,i})^2.$$

Proof. The proof extends the single-layer analysis to the multi-layer setting. We start with the layer-wise smoothness inequality from Assumption 5. The update for each layer is $X_i^{k+1} - X_i^k = \gamma_{k,i} \hat{d}_i^k$. For each layer i inside the summation, we can bound the terms involving its update direction $\hat{d}_i^k$ exactly as in the single-layer case:

- $\| \gamma_{k,i} \hat{d}_i^k \|_{(i)}^2 \leq (\gamma_{k,i})^2 (1 + \delta_{k,i})^2$.
- $\left\langle \nabla_i f(X^k), \hat{d}_i^k \right\rangle_{(i)} \leq - \| \nabla_i f(X^k) \|_{(i)\star} (1 - \delta_{k,i})$.

Substituting these bounds into the main inequality gives the per-iteration descent guarantee for the layer-wise setting:

$$f(X^{k+1}) \leq f(X^k) + \sum_{i=1}^{p} \left[-\gamma_{k,i} \| \nabla_i f(X^k) \|_{(i)\star} (1 - \delta_{k,i}) + \frac{L_i}{2} (\gamma_{k,i})^2 (1 + \delta_{k,i})^2 \right].$$

Rearranging to isolate the measure of progress (the sum of gradient norms) yields

$$\sum_{i=1}^{p} \gamma_{k,i} \| \nabla_i f(X^k) \|_{(i)\star} (1 - \delta_{k,i}) \leq f(X^k) - f(X^{k+1}) + \sum_{i=1}^{p} \frac{L_i}{2} (\gamma_{k,i})^2 (1 + \delta_{k,i})^2.$$

Summing this from $k = 0$ to $K - 1$ and using the telescoping property of the function values on the right-hand side, $\sum_{k=0}^{K-1} (f(X^k) - f(X^{k+1})) = f(X^0) - f(X^K) \leq \Delta^0$, gives the final result stated in the theorem. □

We now analyze the convergence of the layer-wise method when using an adaptive step size for each layer, chosen at each iteration to maximize the guaranteed descent. This approach provides a theoretically-grounded, adaptive learning rate policy for each layer.

Theorem 7 (Iteration Complexity with Layer-Wise Adaptive Step Sizes)**.** *Let Assumption 5 hold. Consider the deterministic layer-wise method where the inexact LMO for each layer i satisfies a uniform error bound $\delta_{k,i} \leq \delta_i < 1$. If at each iteration k, the step size for each layer i is chosen adaptively as*

$$\gamma_{k,i}^* = \frac{\|\nabla_i f(X^k)\|_{(i)\star}(1-\delta_{k,i})}{L_i(1+\delta_{k,i})^2},$$

then to guarantee finding an iterate X^k such that the weighted squared gradient norm $\sum_{i=1}^p \frac{\|\nabla_i f(X^k)\|_{(i)\star}^2}{L_i} \leq \varepsilon^2$ after at most K iterations, it is sufficient to run the algorithm for a number of iterations K satisfying

$$K \geq \frac{2\Delta^0}{\varepsilon^2} \max_{1\leq j\leq p} \left\{ \frac{(1+\delta_j)^2}{(1-\delta_j)^2} \right\}.$$

Proof. The proof begins with the per-iteration descent inequality derived in the proof of the General Convergence Bound theorem:

$$f(X^{k+1}) \leq f(X^k) + \sum_{i=1}^p \left[-\gamma_{k,i}\|\nabla_i f(X^k)\|_{(i)\star}(1-\delta_{k,i}) + \frac{L_i}{2}(\gamma_{k,i})^2(1+\delta_{k,i})^2 \right].$$

The adaptive step size $\gamma_{k,i}^*$ is the value that minimizes the term in the square brackets for each layer i. Substituting this optimal choice into the inequality yields the maximum guaranteed descent at each step

$$f(X^{k+1}) \leq f(X^k) - \sum_{i=1}^p \frac{(\|\nabla_i f(X^k)\|_{(i)\star}(1-\delta_{k,i}))^2}{2L_i(1+\delta_{k,i})^2}.$$

Summing this from $k=0$ to $K-1$ gives the global progress bound

$$\sum_{k=0}^{K-1}\sum_{i=1}^p \frac{\|\nabla_i f(X^k)\|_{(i)\star}^2(1-\delta_{k,i})^2}{2L_i(1+\delta_{k,i})^2} \leq \Delta^0.$$

We assume a uniform (but potentially layer-specific) inexactness bound $\delta_{k,i} \leq \delta_i < 1$. The inequality becomes

$$\frac{1}{2}\sum_{k=0}^{K-1}\sum_{i=1}^p \frac{(1-\delta_i)^2}{(1+\delta_i)^2}\frac{\|\nabla_i f(X^k)\|_{(i)\star}^2}{L_i} \leq \Delta^0.$$

Let $w_i := \frac{(1-\delta_i)^2}{(1+\delta_i)^2}$ be the layer-specific error weight and let $G_k^2 := \sum_{i=1}^p \frac{\|\nabla_i f(X^k)\|_{(i)\star}^2}{L_i}$ be the weighted squared gradient norm. The bound can be written as $\frac{1}{2}\sum_{k=0}^{K-1}\sum_{i=1}^p w_i \frac{\|\nabla_i f(X^k)\|_{(i)\star}^2}{L_i}$. To derive a clear complexity, we can lower-bound the weights by their minimum value, $w_{\min} = \min_{1\leq j\leq p} w_j$:

$$\sum_{i=1}^p w_i \frac{\|\nabla_i f(X^k)\|_{(i)\star}^2}{L_i} \geq w_{\min}\sum_{i=1}^p \frac{\|\nabla_i f(X^k)\|_{(i)\star}^2}{L_i} = w_{\min}G_k^2.$$

Substituting this into the main inequality gives $\frac{w_{\min}}{2}\sum_{k=0}^{K-1} G_k^2 \leq \Delta^0$. Let $G_{\min}^2 = \min_{0\leq j<K} G_j^2$. Since $\sum_{k=0}^{K-1} G_k^2 \geq K \cdot G_{\min}^2$, we have

$$\frac{Kw_{\min}}{2}G_{\min}^2 \leq \Delta^0 \implies G_{\min}^2 \leq \frac{2\Delta^0}{Kw_{\min}}.$$

To guarantee that the weighted squared gradient norm $G_{\min}^2 \leq \varepsilon^2$, we require

$$\frac{2\Delta^0}{Kw_{\min}} \leq \varepsilon^2 \implies K \geq \frac{2\Delta^0}{\varepsilon^2 w_{\min}}.$$

Substituting back the definition of $w_{\min} = \min_j \frac{(1-\delta_j)^2}{(1+\delta_j)^2}$ gives the final complexity

$$K \geq \frac{2\Delta^0}{\varepsilon^2}\frac{1}{\min_j \frac{(1-\delta_j)^2}{(1+\delta_j)^2}} = \frac{2\Delta^0}{\varepsilon^2}\max_j \frac{(1+\delta_j)^2}{(1-\delta_j)^2}.$$

□

Discussion This result for the layer-wise adaptive method is powerful as it provides a concrete, theoretically-grounded learning rate policy for each layer that leads to an optimal convergence rate.

- **Optimal Rate:** The resulting complexity is $\mathcal{O}(1/\varepsilon^2)$, which is the optimal rate for deterministic non-convex optimization. This confirms that the layer-wise adaptive strategy is efficient.
- **Layer-Wise Adaptivity:** The step size $\gamma^*_{k,i}$ for each layer adapts to its own local properties: its current gradient norm $\|\nabla_i f(X^k)\|_{(i)\star}$, its smoothness L_i, and its LMO precision $\delta_{k,i}$. This is a practical and intuitive update rule that naturally assigns smaller steps to sharper or less precisely updated layers.
- **Bottleneck Effect of Inexactness:** The final complexity is degraded by a "worst-layer" factor, $\max_j \frac{(1+\delta_j)^2}{(1-\delta_j)^2}$. This shows that the global convergence speed is bottlenecked by the single layer with the worst combination of update quality and infeasibility cost. A single layer with very high inexactness ($\delta_j \to 1$) will cause this factor to explode, dominating the iteration complexity regardless of how precisely the other layers are handled.

D ADDITIONAL EXPERIMENTS AND DETAILS

This section provides supplementary details and results to support the empirical findings presented in Section 4 of the main paper.

D.1 Experimental setup: details

We provide additional details regarding the experimental setups for both the nanoGPT and CIFAR-10 benchmarks. Further details on model architectures and data processing can be found in the respective code repositories cited in the main text. All experiments were conducted on a system equipped with 4 × NVIDIA A100 GPUs.

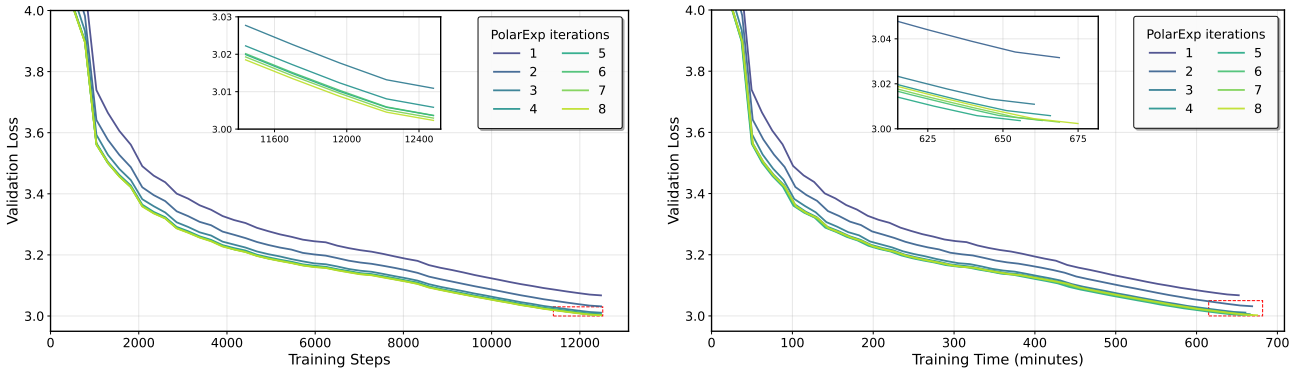
nanoGPT on FineWeb. The hyperparameter grids used for the sweeps shown in Figure 1b are visible on the axes of the plots themselves. The LMO inexactness δ is controlled by the number of iterations of the `PolarExpress` algorithm; fewer iterations correspond to a higher error δ.

CNN on CIFAR-10. For the results presented in Figure 2b, we performed a comprehensive grid search over the following hyperparameters for each level of LMO precision (i.e., for each number of `Newton-Schulz` iterations):

- **Step Size (γ):** A logarithmically spaced grid from 2^{-12} to 2^{-5}.
- **Momentum (α):** A linearly spaced grid from 0.05 to 0.90 in increments of 0.05.

The "Best Accuracy" (solid lines) in Figure 2b corresponds to the maximum test accuracy achieved at a given step size by tuning over the momentum values. The "Worst Accuracy" (dashed lines) corresponds to the minimum test accuracy from the same sweep.

D.2 Additional results for nanoGPT



(a) Improvements saturate after ∼5 `PolarExp` iterations.

(b) Training time grows slightly with the iteration count.

Figure 3: Supplementary results for nanoGPT on FineWeb: convergence behavior (left) and runtime scaling (right) as the number of `PolarExpress` (PolarExp) iterations varies.

Performance Degradation with Increasing Precision. As mentioned in the main text, increasing the LMO precision yields diminishing returns. Figure 3a provides the full validation loss curves for up to 8 iterations of the `PolarExpress` algorithm. The plot clearly shows that while the most significant performance gains are achieved when moving from 1 to 3 iterations, further increases in precision (from 3 to 5, and 5 to 8) continue to improve the convergence speed and final validation loss, albeit with smaller margins. This observation confirms the trend of diminishing returns and empirically validates the degradation factor present in our theoretical bounds.

Table 1 complements these curves by reporting the final validation loss for each setting, highlighting the diminishing returns observed beyond five `PolarExpress` iterations.

D.2.1 LMO Error Analysis.

Aggregate orthogonalization diagnostic. Figure 4 reports an aggregate orthogonalization diagnostic over training, obtained by summing $\|\mathrm{PE}_p(M) - \mathrm{PE}_{100}(M)\|_2$ (PE_p stands for `PolarExpress` ran for p iterations) over the tracked layers at each training step. The figure shows that higher inner-iteration budgets consistently reduce

Table 1: Final validation loss and total training time for NanoGPT on FineWeb across different numbers of `PolarExpress` iterations used for the LMO approximation.

`PolarExpress` iterations	1	2	3	4	5	6	7	8
Validation loss	3.0675	3.0316	3.0109	3.0058	3.0036	3.0037	3.0029	3.0023
Training time (minutes)	652.2	668.8	660.4	665.8	655.7	663.1	668.8	675.0

the aggregate discrepancy and that, for a fixed budget, this quantity remains relatively stable over training apart from occasional transient spikes. This aggregate view is useful for showing that the overall orthogonalization difficulty does not drift dramatically during training. These results provide an explanation for the saturation of performance beyond certain LMO quality, as the theoretical benefits of higher precision diminish once the LMO error reaches sufficiently low levels. However, it hides substantial heterogeneity across tensor families; Figure 5 provides the corresponding block-wise and family-wise breakdown at a representative checkpoint.

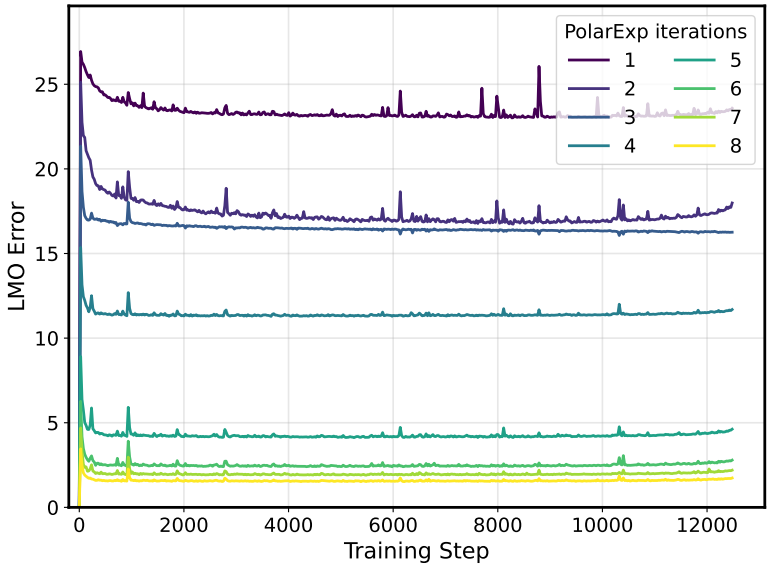


Figure 4: Aggregate `PolarExpress` error over training, defined as the sum over tracked layers of $\|\mathrm{PE}_p(M) - \mathrm{PE}_{100}(M)\|_2$ on the same normalized matrix M. Higher iteration counts maintain lower aggregate orthogonalization error throughout training.

Layer-wise orthogonalization difficulty. Figure 5 complements the aggregate diagnostic in Figure 4 by resolving the orthogonalization error at a representative intermediate (number 6000) training step into transformer block and parameter family. Two patterns are immediate. First, the dominant source of variation is the tensor family rather than the checkpoint: MLP In/Out matrices drop to a low-error band after roughly 5–6 `PolarExpress` iterations in every block, whereas attention matrices remain close to 1 for the first few iterations and typically undergo their main transition only around $k \approx 10$–20. Second, this heterogeneity largely disappears once the inner budget is sufficiently large: by $k \approx 20$, most blocks and families have collapsed to a narrow low-error band. This explains why increasing the `PolarExpress` budget beyond five iterations can still improve NanoGPT performance in Figure 3a: five iterations are already sufficient for the MLP matrices, but several attention matrices are still far from their PE_{100} reference at that budget. The figure also lends empirical support to the layer-wise discussion in Appendix C.2, where the effect of LMO inexactness is predicted to be non-uniform across blocks.

Block 7 appears as an easy outlier because only MLP-type matrices are present in this diagnostic for that block. More generally, the figure shows that a single global summary statistic can hide a strongly bimodal low-budget regime: at small inner-iteration counts, the optimizer already orthogonalizes MLP matrices well, while many attention matrices remain substantially harder.

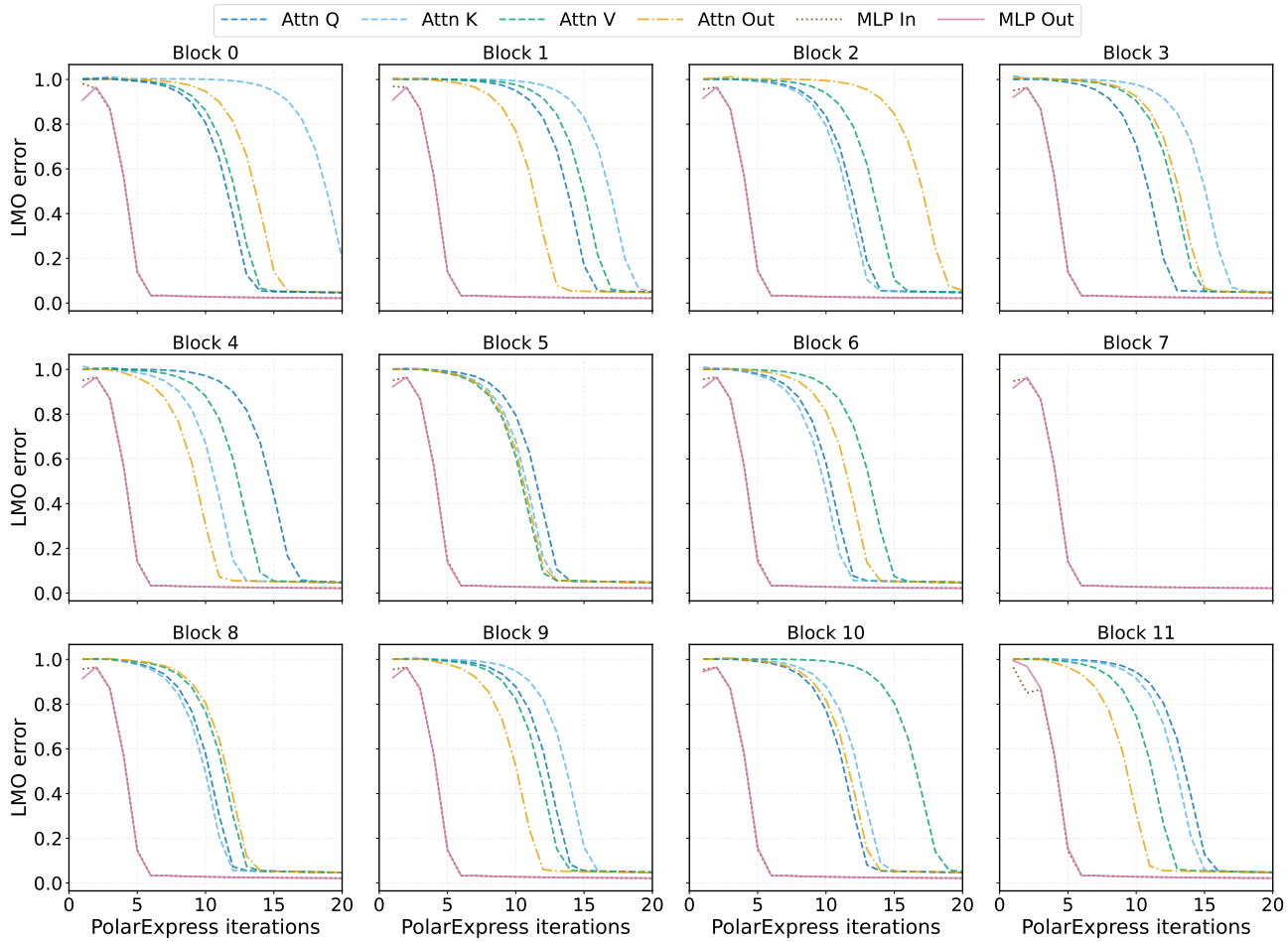


Figure 5: Layer-wise PolarExpress convergence at validation step 6000 for NanoGPT. For each tracked matrix family in each transformer block, we plot the spectral-norm discrepancy $\|\mathrm{PE}_k(M) - \mathrm{PE}_{100}(M)\|_2$ as a function of the inner-iteration count k, where M is the normalized pre-orthogonalization momentum matrix and PE_{100} is used as a high-accuracy reference. MLP matrices converge in only a few iterations, while several attention matrices remain difficult until roughly $k \approx 10$–20, revealing strong heterogeneity across parameter families and blocks.

Why can the plotted error be slightly above 1? Small excursions slightly above 1 at the earliest inner iterations are not contradictory. The plotted quantity is not the error to the exact polar factor, but the difference between two finite-step polynomial iterates, $\|\mathrm{PE}_k(M) - \mathrm{PE}_{100}(M)\|_2$. Moreover, our theory does not assume the approximate direction to remain feasible, so low-budget iterates can transiently have operator norm slightly above 1. In practice, this effect is further moderated by the implementation: before applying PolarExpress, we normalize the matrix by its Frobenius norm with a small ε and a 1.01 safety margin, and the degree-5 coefficient schedule is safety-scaled accordingly to improve numerical stability in bfloat16.

Measurement protocol for orthogonalization diagnostics. For the orthogonalization diagnostics in Figure 4, Figure 5, and Table 2, we record the normalized pre-orthogonalization momentum matrix M during validation and compute

$$\|\mathrm{PE}_k(M) - \mathrm{PE}_{100}(M)\|_2$$

on the same matrix, where PE_{100} serves as a high-accuracy reference. Thus, these measurements should be interpreted as a solver-to-solver calibration of the practical inner routine, rather than as the error to the exact polar factor. In PolarExpress implementation the input is first scaled as

$$M \leftarrow \frac{M}{1.01(\|M\|_F + 10^{-7})},$$

Table 2: Effect of the number of `PolarExpress` iterations on the aggregate orthogonalization discrepancy $e_k(M) = \|\mathrm{PE}_k(M) - \mathrm{PE}_{100}(M)\|_2$, and the final NanoGPT validation loss on FineWeb. Discrepancy statistics are aggregated over all completed checkpoints and saved tensors. This is a solver-to-solver diagnostic relative to a high-accuracy PE_{100} reference on the same normalized matrix, not the error to the exact polar factor.

PE iters	Mean e_k	Std. e_k	Median e_k	Max e_k	Val. loss
1	0.9826	0.0311	1.0011	1.0200	3.0675
2	0.9879	0.0220	1.0016	1.0079	3.0316
3	0.9554	0.0644	1.0012	1.0143	3.0109
4	0.8480	0.2260	0.9964	1.0055	3.0058
5	0.6952	0.4047	0.9895	1.0032	3.0036
6	0.6507	0.4321	0.9797	1.0030	3.0037
7	0.6424	0.4387	0.9616	1.0031	3.0029
8	0.6265	0.4469	0.9272	1.0028	3.0023

which keeps the spectrum inside the convergence region of the polynomial iteration and improves stability in bfloat16.

Table 2 shows that increasing the inner PolarExpress budget consistently reduces the aggregate solver-to-solver discrepancy and improves the final validation loss, but also reveals a strongly heterogeneous low-budget regime: at small k, many attention matrices remain close to their PE_{100} reference only after substantially more iterations than the MLP matrices, as seen in Figure 5.

D.3 Additional results for CIFAR-10

Full Hyperparameter Grids. To complement the summary plots in the main text, Figure 6 presents the full heatmap of final test accuracies across the entire hyperparameter grid for different levels of LMO precision. These plots provide a more detailed view of the optimizer's stability. For a highly precise LMO (e.g., 8 `Newton-Schulz` iterations), a large contiguous region of high performance is visible, indicating robustness to hyperparameter choices. In contrast, for a highly inexact LMO (e.g., 1 `Newton-Schulz` iteration), the high-performance region is smaller and more fragmented, highlighting the optimizer's sensitivity to tuning when the LMO is imprecise.

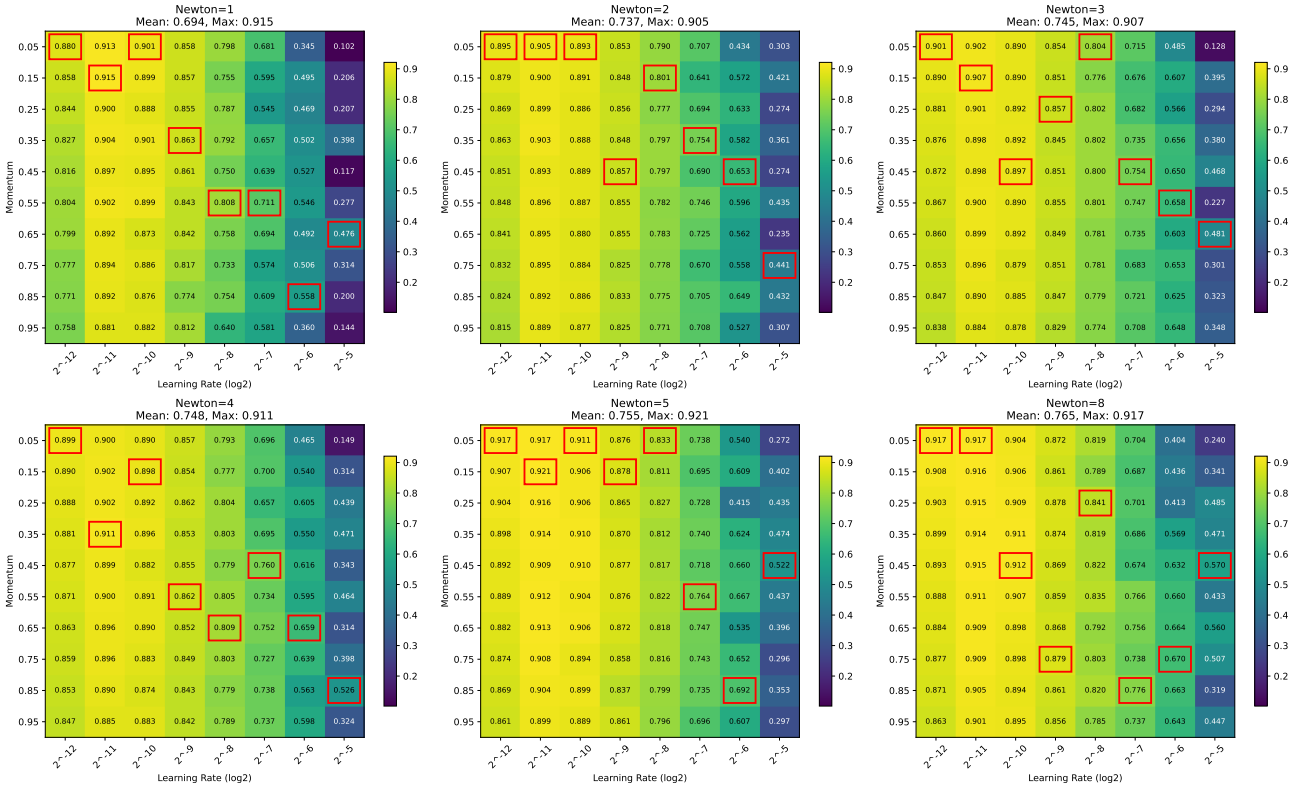


Figure 6: Full final test accuracy heatmaps on CIFAR-10 across a grid of step sizes and momentum values for different numbers of `Newton-Schulz` ("Newton") iterations. Brighter colors indicate higher final test accuracy. The region of high performance is visibly larger and more stable for higher numbers of `Newton-Schulz` iterations.