Convergence Bound and Critical Batch Size of Muon Optimizer

Anonymous authors
Paper under double-blind review

Abstract

Muon, a recently proposed optimizer that leverages the inherent matrix structure of neural network parameters, has demonstrated strong empirical performance, indicating its potential as a successor to standard optimizers such as AdamW. This paper presents theoretical analysis to support its practical success. We provide convergence proofs for Muon across four practical settings, systematically examining its behavior with and without the inclusion of Nesterov momentum and weight decay. We then demonstrate that the addition of weight decay yields strictly tighter theoretical bounds and clarify the interplay between the weight decay coefficient and the learning rate. Finally, we derive a lower bound on the critical batch size for Muon—the batch size that minimizes the stochastic first-order oracle (SFO) complexity of training. Because the resulting formula involves problem-dependent quantities that are not directly observable (gradient variance, target precision, effective rank), it does not predict the critical batch size in absolute terms; rather, it reveals how the hyperparameters β (momentum) and λ (weight decay) govern the qualitative scaling of this value. Our experiments validate these hyperparameter-dependent predictions across workloads including image classification and language modeling.

1 Introduction

Optimization algorithms are fundamental to the training of deep neural networks (DNNs). Since the introduction of stochastic gradient descent (SGD) (Robbins & Monro, 1951), numerous optimizers have been proposed to accelerate convergence. Among these, adaptive gradient algorithms such as Adam (Kingma & Ba, 2015) and its subsequent variant AdamW (Loshchilov & Hutter, 2019) have emerged as the de facto standard in modern deep learning, valued for their rapid convergence and robust performance across a wide range of tasks. A common characteristic of these widely used first-order optimizers is that they treat the weight parameters of neural networks, which are inherently matrices or higher-order tensors, as high-dimensional vectors. However, this vector-based perspective, while effective, disregards the underlying geometric and algebraic structure within the parameter matrices.

The recently proposed Muon optimizer (Jordan et al., 2024) introduces a distinct paradigm that departs from the conventional vector-based viewpoint. The core idea of Muon is to preserve and leverage the intrinsic matrix structure of the network parameters. Instead of using the gradient vector, Muon computes its search direction by orthogonalizing the gradient momentum matrix. Specifically, for a given momentum matrix $C_t \in \mathbb{R}^{m\times n}$, the search direction $O_t \in \mathbb{R}^{m\times n}$ is the orthogonal matrix that is closest in Frobenius norm, found by solving

$$O_t := \operatorname*{argmin}_{O\in\{O\in\mathbb{R}^{m\times n}:O^\top O=I_n\}} \|O - C_t\|_{\mathrm{F}}, \tag{1}$$

where I_n denotes a $n \times n$ identity matrix. As established by classic matrix theory, if the singular value decomposition (SVD) of C_t is $U_t S_t V_t^\top$, the optimal search direction is simply $O_t = U_t V_t^\top$, where $U_t \in \mathbb{R}^{m\times r}, S_t \in \mathbb{R}^{r\times r}, V_t \in \mathbb{R}^{n\times r}$, and $r > 0$ is the rank of C_t. This process of gradient orthogonalization is aimed at finding a search direction that is independent of the gradient's magnitude, potentially leading to

more stable and effective training dynamics. While SVD provides an exact analytical solution, its high computational cost renders it impractical for the large matrices found in modern DNNs. As detailed by Bernstein (2025), the key computational innovation of Muon is the use of the Newton-Schulz iteration (Bernstein & Newhouse, 2024; Higham, 2008; Björck & Bowie, 1971; Kovarik, 1970), a classic and remarkably efficient numerical method, to approximate this orthogonalization. This iterative algorithm enables Muon to compute the search direction without performing an explicit SVD, making it a computationally feasible optimizer for large-scale applications. This elegant fusion of a novel optimization perspective with a powerful numerical technique positions Muon as a promising, theoretically grounded alternative to existing optimizers.

Several studies have reported the strong empirical performance of Muon. Jordan et al. (2024) showed that Muon outperforms Shampoo (Gupta et al., 2018) and SOAP (Vyas et al., 2025), both on a per-step and wall-clock basis. Liu et al. (2025b) showed that by dividing the parameters and updating them with SOAP and Muon, optimization performance equivalent to or better than SOAP can be achieved while substantially reducing memory usage. Liu et al. (2025a) demonstrated that Muon is effective for training large-scale LLMs and suggested it has the potential to replace AdamW as the standard optimizer. AI et al. (2025) showed that Muon expands AdamW's Pareto frontier on the compute-time plane, enlarging the practitioner's flexibility in resource allocation. However, the theoretical understanding of Muon's convergence behavior remains underdeveloped, and a formal justification for its strong performance over AdamW is still lacking. This work aims to bridge this gap by providing a rigorous convergence analysis of Muon.

Furthermore, since the batch size is a critical hyperparameter for managing computational costs in large-scale training, we also consider Muon's critical batch size. This is defined as the batch size that minimizes the computational cost of training. In other words, the critical batch size is the point at which further increases in batch size yield diminishing returns in hardware throughput (i.e., the number of samples processed per unit time). By understanding and utilizing the critical batch size, one can maximize GPU utilization, thereby shortening training times and reducing overall computing costs. Therefore, to theoretically understand Muon and maximize its performance, analyzing its critical batch size is essential. Following previous studies, we aim to clarify the critical batch size of Muon.

Our main contributions are as follows:

- We present a convergence analysis for four variants of Muon—with and without Nesterov momentum and with and without weight decay (Theorems 3.1 and 3.2). For any variant, we showed that the upper bound on the average expected gradient norm is as follows:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right] = \mathcal{O}\left(\frac{1}{\eta T} + \sqrt{\frac{(1-\beta)r}{b}} + n\eta\right),$$

  where $W_t \in \mathbb{R}^{m\times n}$, $\eta > 0$ is a learning rate, $\beta \in [0,1)$ is momentum, b is the batch size, and $r \leq n$ is maximum rank of $C_t - \nabla f(W_t)$. For detailed definitions, see Section 2 and Theorems 3.1 and 3.2. The variant combining both Nesterov momentum and weight decay is of particular interest as it mirrors common practical settings. Our analysis thus offers direct insights into Muon's real-world behavior.

- We prove that incorporating weight decay yields tighter theoretical bounds on both the parameter and gradient norms (Propositions 3.1 and 3.2) and present experimental results to support this finding. We also show that for Muon to converge, the learning rate η and weight decay coefficient λ must satisfy $\eta \leq \frac{1}{\lambda}$, a condition supported by our experimental results (see Figure 1).

- We derive a lower bound on the critical batch size for the four Muon variants (Proposition 4.3). For example, for Muon with Nesterov momentum and weight decay we show

$$b^{\star}_{\mathrm{Muon}} > \frac{9(1-\beta)(1+\sqrt{2}\beta)^2 r\sigma^2}{(1-\lambda)^2\epsilon^2},$$

  where $\beta \in (0,1]$ is momentum, $\lambda > 0$ is the weight decay coefficient, $\sigma^2 > 0$ is the variance of the stochastic gradient, r is the maximum rank of $C_t - \nabla f(W_t)$, and $\epsilon > 0$ is the stopping threshold.

Because the formula contains problem-dependent quantities (σ^2, r, ϵ) that are not directly controlled by the practitioner, it does not predict the absolute value of the critical batch size. Instead, it identifies the qualitative scaling law: how the hyperparameters β and λ govern $b^\star$, and how the ratio between variants depends only on β and λ (the unknown quantities cancel). Our experiments validate these hyperparameter-dependent predictions (see Figures 3–4).

2 Preliminaries

2.1 Notations and Definition

Let $\mathbb{N}$ be the set of nonnegative integers. For $p \in \mathbb{N}\backslash\{0\}$, define $[p] := \{1, 2, \ldots, p\}$. Let $\mathbb{R}^d$ be a d-dimensional Euclidean space. We use lowercase letters for scalars (e.g., $a \in \mathbb{R}$), bold lowercase letters for vectors (e.g., $\boldsymbol{a} \in \mathbb{R}^d$), and uppercase letters for matrices (e.g., $A \in \mathbb{R}^{m\times n}$). $\boldsymbol{a}^\top \in \mathbb{R}^{1\times d}$ and $A^\top \in \mathbb{R}^{n\times m}$ denote the transposes of $\boldsymbol{a} \in \mathbb{R}^d$ and $A \in \mathbb{R}^{m\times n}$, respectively. For a square matrix $A = (a_{ij}) \in \mathbb{R}^{n\times n}$, the trace is defined as $\mathrm{tr}(A) := \sum_{i=1}^n a_{ii}$. For all vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, the Euclidean inner product is defined as $\langle \boldsymbol{x}, \boldsymbol{y}\rangle_2 := \boldsymbol{x}^\top \boldsymbol{y}$ and the Euclidean norm is defined as $\|\boldsymbol{x}\|_2 := \sqrt{\langle \boldsymbol{x}, \boldsymbol{x}\rangle_2}$. For all matrices $A, B \in \mathbb{R}^{m\times n}$, the Frobenius inner product is defined as $\langle A, B\rangle_{\mathrm{F}} := \mathrm{tr}(A^\top B)$, the Frobenius norm defined as $\|A\|_{\mathrm{F}} := \sqrt{\langle A, A\rangle_{\mathrm{F}}}$, and the operator norm defined as $\|A\|_{\mathrm{op}} := \max_{\|\boldsymbol{x}\|_2\le 1} \|A\boldsymbol{x}\|_2$. The model is parameterized by a matrix $W \in \mathbb{R}^{m\times n}$ ($m \ge n$), which is optimized by minimizing the empirical loss function $f(W) := \frac{1}{N}\sum_{i\in[N]} f_i(W)$, where $N \in \mathbb{R}$ is the number of training samples and $f_i(W)$ denotes the loss associated with the i-th training sample $\boldsymbol{z}_i$ ($i \in [N]$). We define $W^\star := \mathrm{argmin}_{W\in\mathbb{R}^{m\times n}} f(W)$. Let ξ be a random variable that does not depend on $W \in \mathbb{R}^{m\times n}$, and let $\mathbb{E}_\xi[X]$ denote the expectation with respect to ξ of a random variable X. $\xi_{t,i}$ is a random variable generated from the i-th sampling at time t, and $\xi_{t,i}$ and $\xi_{t,j}$ are independent ($i \ne j$). $\boldsymbol{\xi}_t := (\xi_{t,1}, \xi_{t,2}, \ldots, \xi_{t,b})^\top$ is independent of sequence $(W_k)_{k=0}^t \subset \mathbb{R}^{m\times n}$ generated by Muon (Algorithm 1), where b ($\le N$) is the batch size. The independence of $\boldsymbol{\xi}_0, \boldsymbol{\xi}_1, \ldots$ allows us to define the total expectation $\mathbb{E}$ as $\mathbb{E} = \mathbb{E}_{\boldsymbol{\xi}_0}\mathbb{E}_{\boldsymbol{\xi}_1}\cdots\mathbb{E}_{\boldsymbol{\xi}_t}$. Let $\mathsf{G}_\xi(W)$ be the stochastic gradient of $f(\cdot)$ at $W \in \mathbb{R}^{m\times n}$. The mini-batch $\mathcal{S}_t$ consists of b samples at time t, and the mini-batch stochastic gradient of $f(W_t)$ for $\mathcal{S}_t$ is defined as $\nabla f_{\mathcal{S}_t}(W_t) := \frac{1}{b}\sum_{i\in[b]} \mathsf{G}_{\xi_{t,i}}(W_t) = \frac{1}{b}\sum_{i\in\mathcal{S}_t} \nabla f_i(W_t)$.

Algorithm 1 presents the most common variant of Muon, which incorporates Nesterov momentum and weight decay. Our implementation of Nesterov momentum and decoupled weight decay (Loshchilov & Hutter, 2019) follows the original formulations of (Jordan et al., 2024). Muon optimizers with both Nesterov momentum and weight decay are often used in practice (e.g., Jordan et al., 2024; AI et al., 2025). NewtonSchulz5($\cdot$) receives C_t, sets $X_0 := C_t/\|C_t\|_{\mathrm{F}}$, performs the following iterations from $k = 0$ to 4, and returns X_5.

$$X_{k+1} := aX_k + b(X_kX_k^\top)X_k + c(X_kX_k^\top)^2X_k,$$

where $a = 3.4445, b = -4.7750, c = 2.0315$. The sequence $(X_k)_{k\in\mathbb{N}}$ converges to O_t defined as Eq. (1) (Bernstein & Newhouse, 2024; Higham, 2008). Our theoretical analysis assumes that $O_t :=$ NewtonSchulz5(C_t) satisfies Eq. (1). Therefore, there may be a gap between the Muon we consider theoretically and its practical implementation. On the other hand, according to Shen et al. (2025), the experimental performance of the SVD-based Muon is equivalent to that of the Newton-Schulz-based version, with the main difference being the higher computational cost of the SVD procedure. That is, experimentally, no gap has been demonstrated. To quantify the approximation gap, note that the Newton-Schulz iteration achieves exponential convergence: if $\tilde{O}_t^{(k)}$ denotes the k-th iterate, then $\|\tilde{O}_t^{(k)} - O_t\|_{\mathrm{F}} \le c\cdot\rho^{2^k}$ for constants $c > 0$ and $\rho \in (0, 1)$ that depend on the condition

Algorithm 1 Muon

Require: $\eta, \lambda > 0, \beta \in [0, 1), M_{-1} := \boldsymbol{0}, W_0 \in \mathbb{R}^{m\times n}$
  for $t = 0$ to $T - 1$ **do**
    $M_t := \beta M_{t-1} + (1-\beta)\nabla f_{\mathcal{S}_t}(W_t)$
    if (Nesterov = True) **then**
      $C_t := \beta M_t + (1-\beta)\nabla f_{\mathcal{S}_t}(W_t)$
    else
      $C_t := M_t$
    end if
    $O_t :=$ NewtonSchulz5(C_t)
    if (weight decay = True) **then**
      $W_{t+1} := (1-\eta\lambda)W_t - \eta O_t$
    else
      $W_{t+1} := W_t - \eta O_t$
    end if
  end for
  return W_T

number of $C_t/\|C_t\|_{\mathrm{F}}$ (Higham, 2008). With $k = 5$ iterations, the approximation error is negligible in practice. The contraction rate ρ worsens as the condition number $\kappa(C_t/\|C_t\|_{\mathrm{F}})$ increases. In principle, early in training or for ill-conditioned momentum matrices, k=5 iterations may not yet reach full machine-precision accuracy. However, two factors mitigate this concern: (i) the pre-normalization step $X_0 := C_t/\|C_t\|_{\mathrm{F}}$ ensures all singular values lie in $[0, 1]$, bounding ρ independently of the gradient magnitude; and (ii) the superlinear (quadratic) convergence of the Newton-Schulz iteration means that even for moderate ρ, the error after k=5 steps is $\mathcal{O}(\rho^{32})$, which is extremely small. Empirically, Shen et al. (2025) confirmed that SVD-based and Newton-Schulz-based Muon yield identical optimization performance. To propagate this error through the analysis, one replaces O_t with $\tilde{O}_t^{(5)}$ in the descent lemma, which introduces an additional per-step term $\eta\|\nabla f(W_t)\|_{\mathrm{F}} \cdot \|\tilde{O}_t^{(5)} - O_t\|_{\mathrm{F}} + L\eta^2\|\tilde{O}_t^{(5)} - O_t\|_{\mathrm{F}}$. Since this error is exponentially small, it does not affect the asymptotic convergence rate and is therefore omitted from our main results for clarity.

2.2 Assumptions

We make the following standard assumptions:

Assumption 2.1. *The function $f\colon \mathbb{R}^{m\times n} \to \mathbb{R}$ is L-smooth, i.e., for all $A, B \in \mathbb{R}^{m\times n}$,*

$$\|\nabla f(A) - \nabla f(B)\|_{\mathrm{F}} \le L\|A - B\|_{\mathrm{F}}.$$

Assumption 2.2. (i) *For all t and all i,*

$$\mathbb{E}_{\xi_{t,i}}\left[\mathsf{G}_{\xi_{t,i}}(W_t)\right] = \nabla f(W_t).$$

(ii) *There exists a nonnegative constant σ^2 such that, for all t and all i,*

$$\mathbb{E}_{\xi_{t,i}}\left[\|\mathsf{G}_{\xi_{t,i}}(W_t) - \nabla f(W_t)\|_{\mathrm{F}}^2\right] \le \sigma^2.$$

3 Analysis of Muon's convergence

3.1 Muon without weight decay

We now present a convergence analysis of Muon (Algorithm 1) without weight decay. The proofs of Theorem 3.1(i) and (ii) are in Appendix B.

Theorem 3.1. *Suppose Assumptions 2.1 and 2.2 hold. Then, for all $t \in \mathbb{N}$,*

(i) for Muon without Nesterov and without Weight Decay,

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right] &\le \frac{f(W_0)}{\eta T} + \frac{8\sqrt{r}\Delta}{(1-\beta)T} + 2\sqrt{2(1-\beta)}\sqrt{\frac{r\sigma^2}{b}} + \frac{4L\sqrt{nr}\eta}{1-\beta} + \frac{Ln\eta}{2} \\ &= \mathcal{O}\left(\frac{1}{\eta T} + \sqrt{\frac{(1-\beta)r}{b}} + n\eta\right),\end{aligned}$$

(ii) for Muon with Nesterov and without Weight Decay,

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right] &\le \frac{f(W_0)}{\eta T} + \frac{8\sqrt{r}\beta\Delta}{(1-\beta)T} + 2(1+\sqrt{2}\beta)\sqrt{1-\beta}\sqrt{\frac{r\sigma^2}{b}} + \frac{4L\beta\sqrt{nr}\eta}{1-\beta} + \frac{Ln\eta}{2} \\ &= \mathcal{O}\left(\frac{1}{\eta T} + \sqrt{\frac{(1-\beta)r}{b}} + n\eta\right),\end{aligned}$$

where $\Delta := \|M_0 - \nabla f(W_0)\|_{\mathrm{F}}$ and $r := \max_{0\le t\le T-1} \operatorname{rank}(C_t - \nabla f(W_t))$.

Theorem 3.1 shows that Muon, both with and without Nesterov momentum, converges to a neighborhood of a stationary point, and achieves same upper bounds on convergence. Note that some terms in the upper bound are slightly smaller when Nesterov momentum is used. Theorem 3.1 suggests that setting the learning rate small and the momentum and batch size large results in a smaller upper bounds. For example, using $\eta := 1/\sqrt{T}$ yields $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right] \leq D\left(\frac{1}{\sqrt{T}} + \sqrt{\frac{(1-\beta)r}{b}}\right)$, where $D > 0$ is non-negative constant.

Interpretation of the convergence rate and the role of rank r. The $\mathcal{O}(1/T)$ rate in Theorem 3.1 matches the standard rate for nonconvex SGD and Adam-type methods, meaning that Muon's theoretical worst-case rate is neither faster nor slower than its competitors in terms of T. The distinguishing feature of Muon's bound lies in the *variance term* $\sqrt{(1-\beta)r/b}$, which depends on the rank $r := \max_t \operatorname{rank}(C_t - \nabla f(W_t))$ rather than the full dimension mn. In practical deep networks, the effective rank of gradient matrices is often much smaller than the full dimension (i.e., $r \ll n$), particularly in over-parameterized regimes where gradients exhibit low-rank structure. Recent empirical evidence supports this: Ahn et al. (2025) showed that a rank-fraction of 1/4 (i.e., $r \approx n/4$) suffices to match full-rank Muon performance in LLMs up to 3B parameters, and that larger models tolerate even lower rank fractions. This suggests that Muon's bound may be tighter than bounds for vector-based optimizers whose variance terms scale with the full parameter count. We emphasize, however, that this is a property of the *upper bound*, not a proof of strict superiority; the actual convergence behavior depends on problem-specific constants that are difficult to compare across optimizers. The empirical advantage of Muon over AdamW observed in our experiments (Figures 2–5) is consistent with this interpretation but may also arise from other mechanisms (e.g., implicit regularization from gradient orthogonalization) not captured by the worst-case bound.

3.2 Muon with weight decay

The following proposition establishes a key result for Muon (Algorithm 1) with weight decay. The proofs of Propositions 3.1 and 3.2 are in Appendix C.

Proposition 3.1. *Suppose Assumptions 2.1 and 2.2 hold, and that Muon is run with $\eta \leq \frac{1}{\lambda}$. Then, for all $t \in \mathbb{N}$,*

$$\|W_t\|_{\mathrm{F}} \leq \begin{cases} (1-\eta\lambda)^t\|W_0\|_{\mathrm{F}} + \frac{\sqrt{n}}{\lambda} & \text{if } \eta < \frac{1}{\lambda}, \\ \frac{\sqrt{n}}{\lambda} & \text{if } \eta = \frac{1}{\lambda}. \end{cases}$$

Proposition 3.2. *Suppose Assumptions 2.1 and 2.2 hold, and that Muon is run with $\eta \leq \frac{1}{\lambda}$. Then, for all $t \in \mathbb{N}$,*

$$\|\nabla f(W_t)\|_{\mathrm{F}} \leq \begin{cases} L(1-\eta\lambda)^t\|W_0\|_{\mathrm{F}} + \frac{L}{\lambda} + L\|W^\star\|_{\mathrm{F}} & \text{if } \eta < \frac{1}{\lambda}, \\ \frac{L}{\lambda} + L\|W^\star\|_{\mathrm{F}} & \text{if } \eta = \frac{1}{\lambda}. \end{cases}$$

Proposition 3.1 establishes that when $\eta \leq \frac{1}{\lambda}$, weight decay ensures the parameter norm remains almost surely bounded. Furthermore, the upper bound decreases monotonically with t, converging to $\frac{\sqrt{n}}{\lambda}$ as $t \to \infty$. The bound is minimized uniformly across all t when $\eta = \frac{1}{\lambda}$. Proposition 3.2 extends the result of Proposition 3.1 to the full gradient norm, which is likewise almost surely bounded. This bound decreases monotonically with t, converging to $\frac{L}{\lambda} + L\|W^\star\|_{\mathrm{F}}$ as $t \to \infty$. From these results, Corollary 3.1 establishes an almost surely bound of Muon. In both cases, the upper bounds are minimized when $\eta = \frac{1}{\lambda}$.

Corollary 3.1. *Suppose Assumptions 2.1 and 2.2 hold and $\eta \leq \frac{1}{\lambda}$. Then, for all $T \in \mathbb{N}$,*

$$\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla f(W_t)\|_{\mathrm{F}} \leq \begin{cases} \frac{L\|W_0\|_{\mathrm{F}}}{\eta\lambda T} + \frac{L}{\lambda} + L\|W^\star\|_{\mathrm{F}} & \text{if } \eta < \frac{1}{\lambda}, \\ \frac{L}{\lambda} + L\|W^\star\|_{\mathrm{F}} & \text{if } \eta = \frac{1}{\lambda}. \end{cases}$$

While these results suggest that a larger weight decay λ yields a tighter bound, the condition $\eta \leq \frac{1}{\lambda}$ necessitates a smaller learning rate η. These desirable properties stem from the fact that Muon's search

direction is inherently bounded. A key advantage of this feature is that our analysis does not rely on the common-and often restrictive-assumption of bounded gradients.

The following is a convergence analysis of Muon (Algorithm 1) with weight decay (the proofs of Theorems 3.2(i) and (ii) are in Appendix C).

Theorem 3.2. *Suppose Assumptions 2.1 and 2.2 hold, and that Muon is run with* $\eta \leq \frac{1}{\lambda}$. *If* $\lambda < \min\left\{\frac{1}{\|W_0\|_{\mathrm{op}}}, 1\right\}$, *then, for all* $t \in \mathbb{N}$,

(i) for Muon without Nesterov and with Weight Decay,

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right] \leq \frac{f(W_0)}{(1-\lambda)\eta T} + \frac{8\sqrt{r}\Delta}{(1-\beta)T} + \frac{2\sqrt{2(1-\beta)}}{1-\lambda}\sqrt{\frac{r\sigma^2}{b}} + \frac{4L\sqrt{rn}\eta}{(1-\beta)(1-\lambda)} + \frac{(1+\lambda^2)Ln\eta}{1-\lambda}$$
$$= \mathcal{O}\left(\frac{1}{\eta T} + \sqrt{\frac{(1-\beta)r}{b}} + n\eta\right),$$

(ii) for Muon with Nesterov and with Weight Decay,

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right] \leq \frac{f(W_0)}{(1-\lambda)\eta T} + \frac{8\beta\sqrt{r}\Delta}{(1-\beta)T} + \frac{2(1+\sqrt{2}\beta)\sqrt{1-\beta}}{1-\lambda}\sqrt{\frac{r\sigma^2}{b}} + \frac{4L\beta\sqrt{rn}\eta}{(1-\beta)(1-\lambda)} + \frac{(1+\lambda^2)Ln\eta}{1-\lambda}$$
$$= \mathcal{O}\left(\frac{1}{\eta T} + \sqrt{\frac{(1-\beta)r}{b}} + n\eta\right),$$

where $\Delta := \|M_0 - \nabla f(W_0)\|_{\mathrm{F}}$, $r := \max_{0\leq t\leq T-1} \mathrm{rank}(C_t - \nabla f(W_t))$.

Similar conclusions follow from Theorem 3.2, which again demonstrate a modest advantage from incorporating Nesterov momentum. These results build on Propositions 3.1 and 3.2 and therefore inherit the assumption that $\eta \leq \frac{1}{\lambda}$. In other words, for Muon with weight decay to attain the stated convergence rate, it must satisfy $\eta \leq \frac{1}{\lambda}$. Practically speaking, since the weight decay coefficient λ is typically less than 1, this assumption is realistic and does not materially constrain the choice of learning rate. Furthermore, Theorem 3.2 requires that the weight decay coefficient satisfy $\lambda < \min\left\{\frac{1}{\|W_0\|_{\mathrm{op}}}, 1\right\}$, which recommends using a small λ. In practice, λ is often used with values much smaller than 1, and this assumption can also be considered realistic. Compared to Theorem 3.1, combining Muon and weight decay allows us to multiply several terms in the upper bound by $\frac{1}{1-\lambda}$. When using sufficiently small λ, we have $\frac{1}{1-\lambda} \approx 1$, so we see that even with weight decay, the upper bound does not grow excessively large. As in Theorem 3.1, these bounds represent convergence to a neighborhood of a stationary point, suggesting that setting the learning rate small and the momentum and batch size large results in a smaller upper bounds.

4 Analysis of Muon's critical batch size

Our theoretical analysis characterizes the critical batch size as a function of the gradient variance σ^2 and optimization hyperparameters. While explicitly modeling the dependence of σ^2 on model width or depth is beyond the scope of this single-matrix analysis, our results establish the fundamental relationship $b^\star \propto \sigma^2$. This suggests that scaling behaviors observed in larger models are mediated through changes in their gradient noise properties. Indeed, recent large-scale empirical studies report that Muon remains efficient at increasingly large batch sizes for large language models (Liu et al., 2025a); our theory supports this observation, predicting that if larger models entail distinct gradient variance characteristics, the critical batch size will shift accordingly. With this motivation, we now formalize the notion of the critical batch size used in our analysis.

We next introduce the concept of the critical batch size, defined as the batch size that minimizes computational complexity. This complexity is measured in terms of the stochastic first-order oracle (SFO) complexity,

which is the total number of stochastic gradient computations. Since the optimizer computes b stochastic gradients per step, an optimizer that runs for T steps with batch size b incurs a total of Tb SFO complexity. Empirically, for batch sizes up to a certain threshold $b^\star$ (the critical batch size), the number of training steps T required to train a DNN scales inversely with b (Shallue et al., 2019; Ma et al., 2018; McCandlish et al., 2018). Beyond $b^\star$, increasing the batch size yields diminishing returns in reducing T. The critical batch size is therefore the batch size that minimizes SFO complexity Tb. Prior work has shown that $b^\star$ depends on both the optimizer (Zhang et al., 2019) and dataset size (Zhang et al., 2025) and has established a theoretical framework for proving its existence and estimating its lower bound (Sato & Iiduka, 2023; Imaizumi & Iiduka, 2024). Concurrently, Naganuma et al. (2026) derived a non-Euclidean gradient noise scale tailored to spectral descent methods (including Muon) and proposed an adaptive batch-size schedule based on this noise scale. Their analysis characterizes the critical batch size of the spectral gradient step itself but does not consider the effects of Nesterov momentum or weight decay. In contrast, our work adopts the SFO-minimization framework and aims to clarify the relationship between the Muon hyperparameters β and λ and its critical batch sizes.

4.1 Relationship between batch size and number of steps needed for training

Suppose Assumptions 2.1 and 2.2 hold. Then, by Theorems 3.1 and 3.2, the following inequality holds:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right] \leq \frac{X}{T} + \frac{Y}{\sqrt{b}} + Z,$$

where $X, Y, Z > 0$ are nonnegative constants. Let $\epsilon > 0$ be an arbitrarily fixed threshold.

$$\exists T, \exists b : \frac{X}{T} + \frac{Y}{\sqrt{b}} + Z < \epsilon, \tag{2}$$

The relationship between b and the number of steps T_b satisfying Eq. (2) is as follows:

Proposition 4.1. *Suppose Assumptions 2.1 and 2.2 hold and let Muon be the optimizer under consideration. Then, T_b defined by*

$$T_b := \frac{X\sqrt{b}}{(\epsilon - Z)\sqrt{b} - Y} < T \quad \textit{for} \quad \sqrt{b} > \frac{Y}{\epsilon - Z}, \tag{3}$$

satisfies Eq. (2). In addition, the function T_b defined by Eq. (3) is monotone decreasing and convex for $\sqrt{b} > \frac{Y}{\epsilon - Z}$.

Proof. According to Eq.(3), Muon satisfies Eq.(2). For $\sqrt{b} > \frac{Y}{\epsilon - Z}$, we have

$$\frac{\mathrm{d}T_b}{\mathrm{d}b} = \frac{-XY}{2\sqrt{b}\left\{(\epsilon - Z)\sqrt{b} - Y\right\}^2} \leq 0, \quad \frac{\mathrm{d}^2 T_b}{\mathrm{d}b^2} = \frac{XY\left\{3(\epsilon - Z)\sqrt{b} - Y\right\}}{4b\sqrt{b}\left\{(\epsilon - Z)\sqrt{b} - Y\right\}^3} \geq 0.$$

Therefore, T_b is monotone decreasing and convex for $\sqrt{b} > \frac{Y}{\epsilon - Z}$. This completes the proof. □

Proposition 4.1 implies that the required number of steps T_b decreases as $\sqrt{b}$ increases. In fact, the reduction in the required number of steps achieved by increasing the batch size has a ceiling. The next section will show that a critical batch size exists that minimizes computational complexity.

4.2 Existence of a critical batch size

The critical batch size minimizes the computational complexity for training. Here, we use SFO complexity as a measure of computational complexity. Since the stochastic gradient is computed b times per step, SFO

complexity is defined as

$$T_b b = \frac{Xb\sqrt{b}}{(\epsilon - Z)\sqrt{b} - Y}. \tag{4}$$

The following theorem guarantees the existence of critical batch sizes that are global minimizers of $T_b b$ defined by Eq. (4).

Proposition 4.2. *Suppose that Assumptions 2.1 and 2.2 hold and consider Muon. Then, there exists*

$$b^\star_{Muon} > \frac{9Y^2}{4\epsilon^2} \tag{5}$$

such that $b^\star_{Muon}$ minimizes SFO complexity $T_b b$.

Proof. From Eq.(5), we have that, for $\sqrt{b} > \frac{Y}{\epsilon - Z}$,

$$\frac{\mathrm{d}T_b b}{\mathrm{d}b} = \frac{X\sqrt{b}\left\{2(\epsilon - Z)\sqrt{b} - 3Y\right\}}{2\left\{(\epsilon - Z)\sqrt{b} - Y\right\}^2}, \quad \frac{\mathrm{d}^2 T_b b}{\mathrm{d}b^2} = \frac{XY\left\{3Y - (\epsilon - Z)\sqrt{b}\right\}}{4\sqrt{b}\left\{(\epsilon - Z)\sqrt{b} - Y\right\}^3}.$$

Since $\frac{\mathrm{d}^2 T_b b}{\mathrm{d}b^2} > 0$ holds when $\sqrt{b} < \frac{3Y}{\epsilon - Z}$ and $\frac{\mathrm{d}^2 T_b b}{\mathrm{d}b^2} \leq 0$ holds when $\sqrt{b} \geq \frac{3Y}{\epsilon - Z}$, $T_b b$ is a convex function for $\frac{Y}{\epsilon - Z} < \sqrt{b} < \frac{3Y}{\epsilon - Z}$. In addition, from $\frac{\mathrm{d}T_b b}{\mathrm{d}b} = 0$, we find that $\sqrt{b^\star_{\mathrm{Muon}}} = \frac{3Y}{2(\epsilon - Z)}$ is a global minimizer of $T_b b$. Therefore, we have

$$\sqrt{b^\star_{\mathrm{Muon}}} > \frac{3Y}{2\epsilon}, \text{ i.e., } b^\star_{\mathrm{Muon}} > \frac{9Y^2}{4\epsilon^2}$$

This completes the proof. □

In Proposition 4.2, we obtained a lower bound for the critical batch size using $\frac{3Y}{2(\epsilon - Z)} > \frac{3Y}{2\epsilon}$. We provide additional context regarding the tightness of this inequality. In any variant we are considering, the term Z is expressed using the parameter dimension n and the learning rate η. For example, in the case of Muon with Nesterov momentum and weight decay, $Z = \frac{4L\beta\sqrt{rn}\eta}{(1-\beta)(1-\lambda)} + \frac{(1+\lambda^2)Ln\eta}{1-\lambda} = \mathcal{O}(n\eta)$. Therefore, by choosing a sufficiently small learning rate η, the term Z also becomes sufficiently small, and this inequality can be considered tight. On the basis of Theorems 3.1 and 3.2 and Proposition 4.2, we derive the following proposition, which gives $b^\star_{\mathrm{Muon}}$.

Proposition 4.3. *Suppose Assumptions 2.1 and 2.2 hold. Then, for a given precision ϵ, the lower bound of the critical batch size for Muon is shown in Table 1.*

Table 1: Approximate lower bound of critical batch size $b^\star_{\mathrm{Muon}}$ computed with $\beta = 0.95$ and $\lambda = 0.0625$.

	w/o weight decay	w/ weight decay
w/o Nesterov	$\frac{18(1-\beta)r\sigma^2}{\epsilon^2} \approx 0.9 \times \frac{r\sigma^2}{\epsilon^2}$	$\frac{18(1-\beta)r\sigma^2}{(1-\lambda)^2\epsilon^2} \approx 1.0 \times \frac{r\sigma^2}{\epsilon^2}$
w/ Nesterov	$\frac{9(1-\beta)(1+\sqrt{2}\beta)^2 r\sigma^2}{\epsilon^2} \approx 2.47 \times \frac{r\sigma^2}{\epsilon^2}$	$\frac{9(1-\beta)(1+\sqrt{2}\beta)^2 r\sigma^2}{(1-\lambda)^2\epsilon^2} \approx 2.81 \times \frac{r\sigma^2}{\epsilon^2}$

The results in Table 1 indicate that, for a given accuracy ϵ, combining Nesterov momentum significantly increases the critical batch size for Muon. In contrast, the increase in critical batch size achieved by combining weight decay is modest when using sufficiently small values of λ. Furthermore, in all cases, Y^2 becomes smaller as $\beta \to 1$ (e.g., $Y^2 = (1-\beta)(1+\sqrt{2}\beta)^2 \to 0$), suggesting that the larger the momentum β, the smaller the critical batch size $b^\star_{\mathrm{Muon}}$. We stress that the formulas in Table 1 are lower bounds derived from

the convergence upper bound and contain the problem-dependent quantities σ^2, ϵ, and r. Accordingly, they should be read as *qualitative scaling laws* that identify which hyperparameters govern $b^\star$ and how, rather than as numerically predictive estimates. When comparing two Muon variants on the same workload, the unknown quantities cancel and the ratio depends only on β and λ (see Section 5). In the next section, we provide numerical experiments focusing on this relationship between the critical batch sizes and hyperparameters β and λ (See Figures 3–6).

5 Numerical Experiments

We evaluate Muon on three workloads: (i) ResNet-18 on CIFAR-10, (ii) VGG-16 on CIFAR-100, and (iii) Llama3.1 (160M) on the C4 corpus. We first analyze convergence and critical batch size on CIFAR-10 with ResNet-18, then report language–modeling results on C4. Results for VGG-16 on CIFAR-100 are summarized in Appendix F, where we observe the same qualitative trends.

Experimental Setup For CIFAR-10 and CIFAR-100 experiments, unless otherwise stated, we tuned the learning rate by grid search at a base batch size of 512 and applied square-root scaling for Muon and AdamW; for Momentum SGD we tried both square-root and linear scaling. Each configuration was run five times with different seeds and we report mean and standard deviation. For C4 dataset [1] we trained Llama3.1 (160M) with sequence length 2048 and batch sizes from 64 to 8192. SFO complexity is measured as *steps* $\times$ *batch size*. Further details of the experimental protocol are in Appendix D.

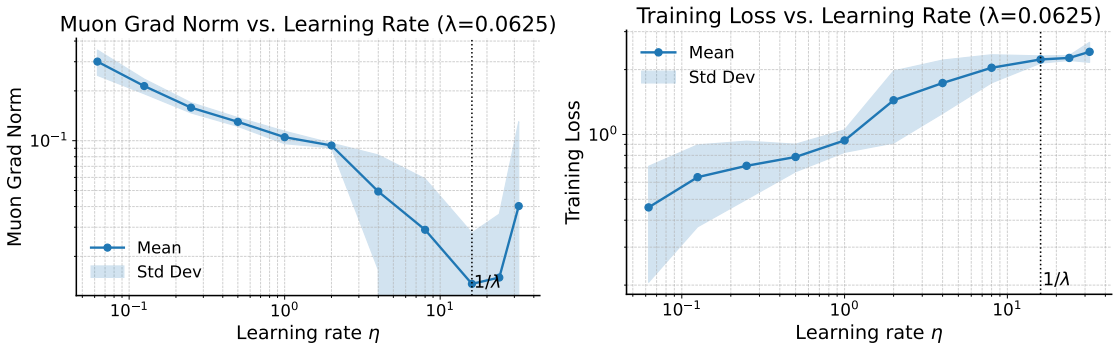

Figure 1: Empirical validation of the stability condition in Proposition 3.2. Final gradient norm (left) and training loss (right) for ResNet-18 on CIFAR-10 with Muon at λ=0.0625. The dashed line shows η=1/λ. Training is most stable near this value.

In the vision workloads, we follow common practice and use a hybrid optimizer (Muon on matrix-shaped parameter blocks and AdamW on the remaining parameters). This reflects the standard deployment of Muon: since the original release (Jordan et al., 2024), Muon has been applied exclusively to matrix-shaped (2D) parameters while a separate optimizer (typically AdamW) handles biases, normalization layers, and embedding vectors. We therefore adopt the same hybrid configuration in our practical-scale experiments to ensure our results are directly relevant to practitioners.

Two-level validation strategy: full Muon vs. hybrid Muon. Our theory analyzes a single matrix parameter updated entirely by Muon, whereas practical implementations use a hybrid optimizer (Muon on matrix-shaped blocks, AdamW on the rest). To bridge this gap, we employ a two-level strategy: (i) a *controlled full-Muon setting* (Appendix E) that exactly matches the theoretical assumptions, confirming the convergence proxy $X/T + Y/\sqrt{b} + Z$ ($R^2 \approx 0.914$) and CBS scaling $b^\star \propto 1/\varepsilon^2$; and (ii) *practical-scale hybrid experiments* (ResNet/CIFAR, VGG/CIFAR, Llama/C4) showing that qualitative predictions (CBS ordering, β–CBS monotonicity) transfer to the hybrid regime. This transfer is expected because the Muon component dominates ($>$ 99.8% of parameters for ResNet-18; Table 5), gradient noise scales are highly correlated across

[1] https://huggingface.co/datasets/allenai/c4

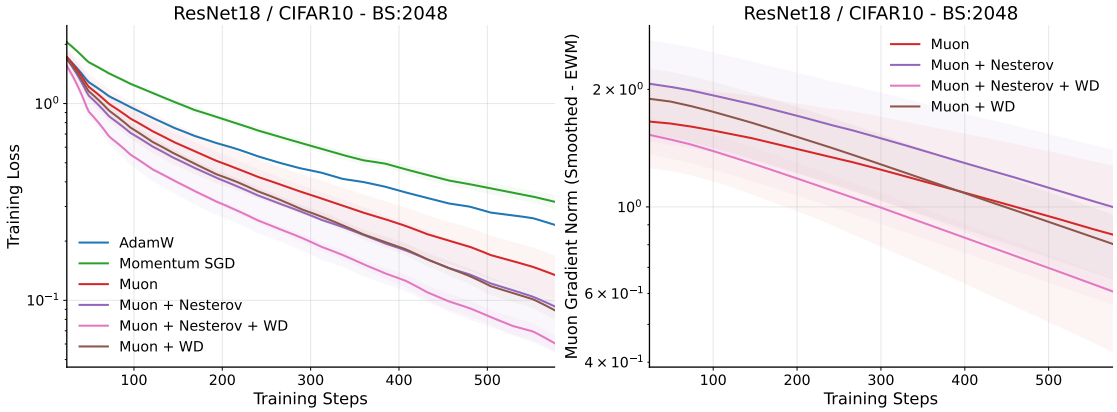

Figure 2: Convergence rate comparison for ResNet-18 on CIFAR-10 with batch size 2048. Training loss (left) and smoothed gradient norm (right) over steps. Muon with Nesterov momentum and weight decay converges the fastest.

layer types (Gray et al., 2024), and the CBS predictions depend only on hyperparameter ratios, not problem-specific constants.

Theory-aligned stopping proxy: gradient-norm threshold. Our theory upper-bounds the average expected full-gradient norm. To better align experiments with this proxy, we additionally report a gradient-norm–based stopping metric. For each run, we track the Frobenius norm of the (mini-batch) gradient, $g_t := \|\nabla f_{\mathcal{S}_t}(W_t)\|_{\mathrm{F}}$, and define a smoothed estimate $\tilde{g}_t$ via an exponential moving average (EMA) over steps.[2] Given a target threshold ϵ, we define the stopping time $T_\varepsilon(b)$ as the first step such that $\tilde{g}_t \leq \epsilon$, and report both steps $T_\varepsilon(b)$ and SFO complexity $b \cdot T_\varepsilon(b)$. We emphasize that our original loss/accuracy targets remain useful for practitioner-facing comparisons, while the gradient-norm criterion is introduced specifically to validate the theory-aligned proxy.

Convergence Analysis We empirically validated the stability condition from Proposition 3.2. Figure 1 shows final gradient norm and training loss for ResNet-18 on CIFAR-10 across learning rates η at fixed weight decay λ=0.0625. The vertical dashed line marks the threshold η=1/λ=16.0. The lowest gradient norm occurs near this threshold; for larger η training becomes unstable. The same behavior holds for other values of λ (Appendix F). This result validates a key practical implication of Propositions 3.1 and 3.2: the condition $\eta \leq 1/\lambda$ is not merely a theoretical convenience but a sharp stability boundary. Practitioners tuning Muon with weight decay should therefore set the learning rate at or below $1/\lambda$.

We next compared the four Muon variants with AdamW and Momentum SGD. Figure 2 shows that Muon with Nesterov momentum and weight decay attains the fastest decrease in both loss and gradient norm. This ordering is consistent with Theorems 3.1 and 3.2: although all four variants share the same asymptotic rate $\mathcal{O}\Big(1/(\eta T) + \sqrt{(1-\beta)r/b} + n\eta\Big)$, the leading constants are slightly smaller when weight decay is included (due to the factor $1/(1-\lambda)$ tightening the effective bound).

Critical Batch Size We measured the number of steps and SFO needed to reach 90% test accuracy and 95% training accuracy. Figure 3 shows that Muon scales better with batch size than the baselines. SFO is the lowest for Muon over the entire range, and Nesterov shifts the SFO-minimizing batch size to larger values. Momentum SGD requires more steps within the same schedule but eventually reaches comparable accuracy (Appendix F).

[2] For large-scale workloads, computing the full gradient is impractical; we therefore use the mini-batch gradient norm as an unbiased proxy for the full gradient and report EMA-smoothed curves for stability. In a controlled toy setting (Appendix E), we also compute the full-batch gradient norm to directly match the theoretical quantity.

Table 1 predicts that, among the four Muon variants, adding Nesterov momentum increases the CBS coefficient more than adding weight decay alone. In Figure 3 (Right), the SFO-minimizing batch sizes of the four Muon variants are close to each other, with slight differences that are directionally consistent with the theoretical ordering. Across all tested batch sizes, Muon achieves fewer steps and lower SFO than AdamW.

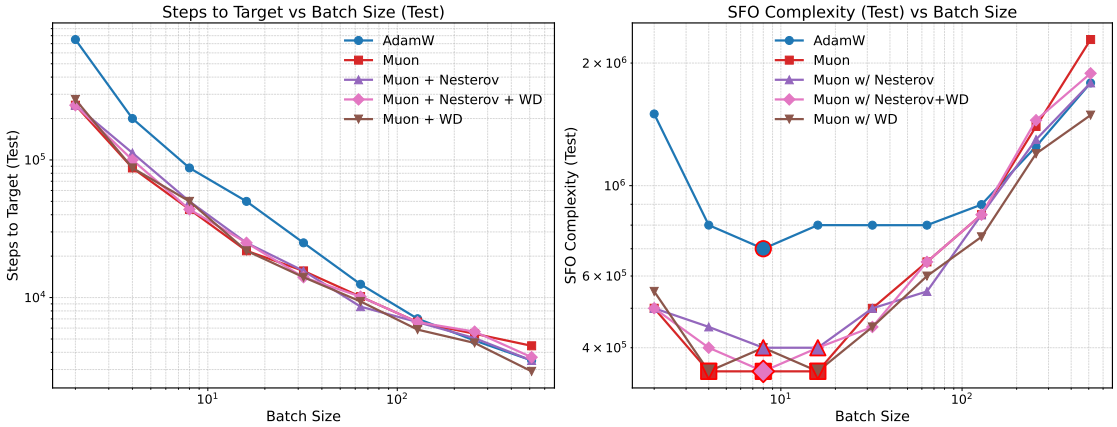

Figure 3: Batch-size scaling and SFO on ResNet-18/CIFAR-10. (Left) Steps to reach 90% test accuracy. (Right) SFO to reach 95% training accuracy. Muon achieves fewer steps and lower SFO than AdamW across all tested batch sizes.

Role of rank r and unknown quantities in the CBS formula. The CBS formulas in Table 1 contain several problem-dependent quantities that are not directly controlled by the practitioner: the gradient variance σ^2, the target precision ϵ, and the maximum rank $r := \max_{0 \le t \le T-1} \operatorname{rank}(C_t - \nabla f(W_t))$, where $r \le n$. Because the primary goal of our analysis is to clarify how the *hyperparameters* β and λ govern the critical batch size, the quantities σ^2, ϵ, and r play the role of task-dependent constants that cancel when comparing different Muon variants on the same workload. For instance, the ratio of CBS with Nesterov to CBS without Nesterov is $(1 + \sqrt{2}\beta)^2/2$, which depends only on β. Our experiments in Figures 3–6 are designed to test precisely these hyperparameter-dependent predictions and do not require knowledge of r or σ^2. We note that recent work by Ahn et al. (2025) provides indirect evidence that $r \ll n$ in practice: their Dion optimizer achieves near-full-rank Muon performance with only $r \approx n/4$ in LLMs up to 3B parameters, suggesting that the low-rank structure assumed in our bound is realistic.

Tightness of the CBS lower bound. Proposition 4.2 provides a lower bound on the critical batch size via $b^\star_{\text{Muon}} > 9Y^2/(4\epsilon^2)$, where the gap arises from bounding $\epsilon - Z$ by ϵ. As noted in Section 4.2, $Z = \mathcal{O}(n\eta)$ and becomes negligible for sufficiently small learning rates. In our controlled toy experiment (Appendix E), the fitted value of Z is close to zero ($\hat{Z} \approx -0.017$, effectively negligible), and the simplified model without Z still achieves $R^2 \approx 0.853$, empirically supporting that the lower bound is reasonably tight in practice.

Critical batch size under the gradient-norm stopping proxy (controlled full Muon). As the first level of our validation strategy, we run a controlled experiment where Muon is applied to *all* parameters (full Muon) on a Teacher-Student Tanh Regression task (Appendix E). This setting exactly matches the theoretical assumptions: a single matrix parameter updated entirely by Muon, with no hybrid optimizer confound. We define the empirical SFO as $b \cdot T_\varepsilon(b)$ where $T_\varepsilon(b)$ is the first time the EMA-smoothed gradient norm drops below ϵ. We observe a clear U-shaped SFO—batch curve and a well-defined minimizer $b^\star$ (e.g., $b^\star$=32 for ϵ=0.08), providing evidence that a critical batch size exists when using the theory-aligned gradient-norm criterion.

Critical batch size under the gradient-norm stopping proxy (controlled full Muon). We further quantify how well the theoretical decomposition matches observations by fitting

$$\bar{g}(T, b) \approx \frac{X}{T} + \frac{Y}{\sqrt{b}} + Z$$

to the measured average gradient norms collected across multiple (T, b) pairs in the controlled full-Muon MLP setting. A linear regression in the features $(1/T, 1/\sqrt{b}, 1)$ achieves a high goodness-of-fit (e.g., R^2=0.914), supporting that the proxy captures the dominant scaling with T and b. From the fitted coefficients, we obtain $\hat{Y}$ and $\hat{Z}$ and compute a *predicted* critical batch size via

$$b^\star_{\text{pred}} = \frac{9\hat{Y}^2}{4(\epsilon - \hat{Z})^2},$$

which matches the empirically SFO-minimizing batch size in the regime $\epsilon > \hat{Z}$. Moreover, when ϵ is not too close to the floor $\hat{Z}$, we observe the predicted scaling $b^\star \propto 1/(\epsilon - \hat{Z})^2$, consistent with the theoretical prediction from Proposition 4.2.

Effect of β on Muon's Critical Batch Size Theory in Section 4.2 predicts that the critical batch size decreases as $\beta \to 1$. Figure 4 confirms this trend for ResNet-18/CIFAR-10, regardless of weight decay or Nesterov. Quantitatively, the CBS formulas in Table 1 contain the factor $(1-\beta)$ (or $(1-\beta)(1+\sqrt{2}\beta)^2$ with Nesterov), both of which vanish as $\beta \to 1$, driving the critical batch size toward zero. In Figure 4, the SFO-minimizing batch size shifts leftward as β increases from 0.7 to 0.999, consistent with this monotone relationship. Practically, this suggests that high-momentum configurations can be trained efficiently at relatively small batch sizes, reducing the hardware resources required.

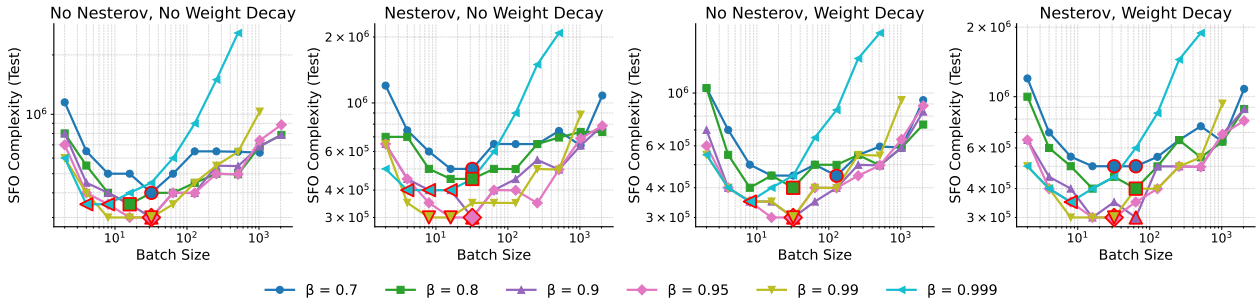

Figure 4: Dependence of SFO and critical batch size on β for ResNet-18/CIFAR-10. The critical batch size consistently decreases as β increases, in line with Section 4.2.

Language-Model Workload: C4 on Llama3.1 (160M) We now test whether the findings hold for large-language-model training. We train Llama3.1 (160M) (18 layers, hidden dimension 768, 12 attention heads; approximately 160M parameters) on the C4 corpus with batch sizes ranging from 32 to 4096 (details in Appendix D). Muon is applied to all matrix-shaped Transformer parameters (approximately 127M parameters), while the remaining parameters (embeddings, layer norms) are updated by AdamW (Table 5). We define the stopping criterion as the number of steps to reach a target training loss, and report SFO complexity = steps $\times$ batch size. Figure 5 reports steps to reach the target loss and SFO versus batch size. Muon generally requires fewer steps and achieves lower SFO than AdamW, particularly at small to moderate batch sizes. At the largest batch sizes, the Muon variants show increasing SFO, narrowing the gap with AdamW. In this setting, adding Nesterov momentum or weight decay does not yield consistent gains. This observation is not at odds with the theory: Table 1 predicts that Nesterov increases the CBS (allowing efficient use of larger batches), but it does not guarantee improved *final loss* when the training budget is fixed. The step/SFO advantage of Muon over AdamW is consistent with the convergence rate established in Theorems 3.1 and 3.2.

Momentum Sweeps on C4 We varied β on C4 to examine the critical batch size. Figure 6 shows that a moderate value, $\beta \approx 0.95$, gives the best loss and SFO. As β decreases, the critical batch size increases; as β increases toward 1, the critical batch size decreases, but extreme values are suboptimal. These observations are consistent with Section 4.2 and mirror the vision results. The fact that the same β–CBS relationship holds across both vision (Figure 4) and language modeling (Figure 6) workloads supports the generality of the theoretical prediction, even though the two settings differ substantially in model architecture, loss landscape, and the ratio of Muon-updated to AdamW-updated parameters.

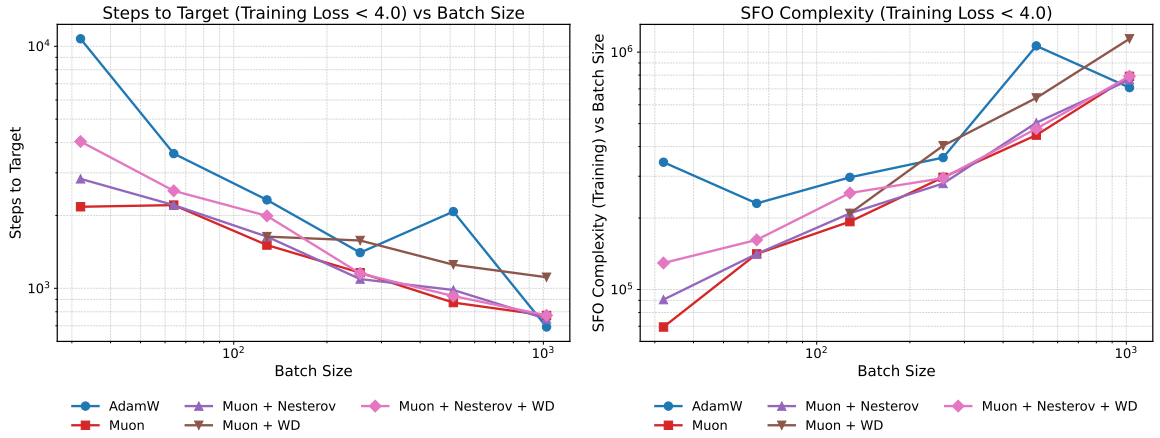

Figure 5: Batch-size scaling on C4 with Llama3.1 (160M). Steps to reach the target training loss (left) and SFO complexity (right) versus batch size. Muon generally achieves lower steps and SFO than AdamW at small to moderate batch sizes. Nesterov momentum and weight decay provide little additional benefit for this workload.

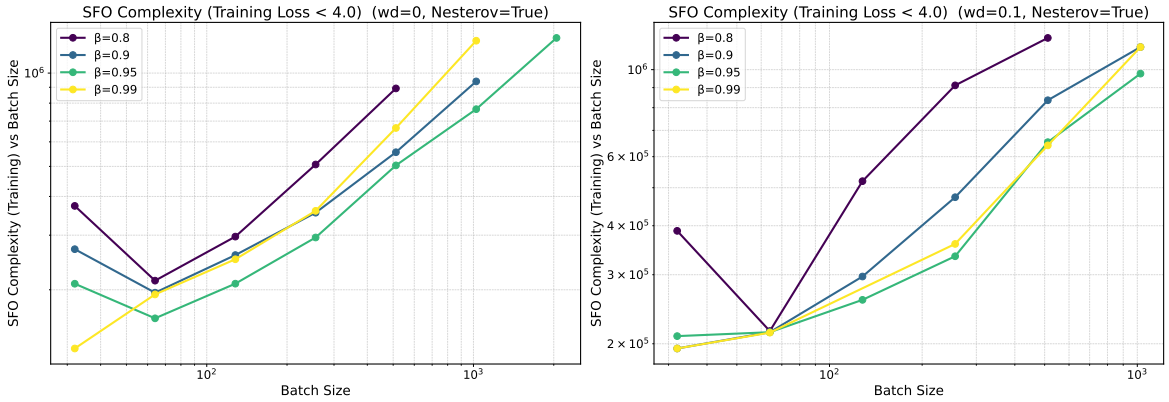

Figure 6: Effect of momentum β on C4 with Llama3.1 (160M). SFO complexity across batch sizes for different β values, without weight decay (left) and with weight decay (right). The best trade-off occurs near β=0.95. The critical batch size decreases with larger β.

6 Related Works

Several studies have investigated the theoretical properties and convergence behavior of Muon. Bernstein & Newhouse (2024) connected Muon to momentum applied in the steepest descent direction under a spectral norm constraint. Li & Hong (2025) provided a pioneering analysis assuming Frobenius norm Lipschitz smoothness. Pethick et al. (2025) studied Muon in the context of optimization methods that use linear minimization oracles for norm balls, and established a convergence rate. Connections to other optimizers have also been explored. Shah et al. showed that Shampoo and SOAP reduce to Muon under simplifying assumptions (AI et al., 2025). Kovalev (2025) proposed and analyzed a stochastic non-Euclidean trust-region method that includes Muon as a special case. Similarly, An et al. (2025) proposed an adaptive structured gradient optimization algorithm that matches Muon in its momentum-free variant. Other studies have explored specific properties and extensions of Muon. Petrov et al. (2025) proposed and analyzed a zeroth-order version . Chen et al. demonstrated Muon's compatibility with the Lion-$\mathcal{K}$ algorithm (Chen et al., 2024) and showed that Muon with weight decay implicitly solves an optimization problem with a spectral norm constraint (Chen et al., 2025). Shen et al. (2025) presented a comprehensive analysis of Muon's convergence rate in comparison to gradient descent. Lau et al. (2025) introduced PolarGrad, a unifying framework for matrix-aware preconditioned methods including Muon, and established convergence rates. The core concepts

of gradient orthogonalization and dualization, which are central to Muon, were introduced in the foundational works by Carlson et al. (2015) and Flynn (2017). On the practical side, Naganuma et al. (2026) derived a non-Euclidean gradient noise scale for spectral descent (including Muon) and used it to build an adaptive batch-size schedule, providing a complementary, empirically driven perspective on the critical batch size of Muon; however, their analysis does not account for the effects of momentum β or weight decay λ. The convergence rates of major related works are summarized in Table 2. Our contribution is the first study to analyze the Muon gradient norm using both Nesterov momentum and weight decay.

The framework for assumptions and proofs presented by Shen et al. (2025) is the most similar to our own. The crucial difference between our study and all previous studies, including this one, is that we also consider most common variant of Muon, which incorporates Nesterov momentum and weight decay. Technically, the technique of transforming $\langle C_t, O_t \rangle_{\mathrm{F}}$ using the dual norm and inverse triangle inequality follows the proof of Pethick et al. (2025) (see proof of Theorem B.1 in Appendix B).

Table 2: Comparison of convergence rates in related works. $\|\cdot\|_\star$ denotes an arbitrary norm, and $\|\cdot\|_*$ denotes the nuclear norm. Each result has been rewritten to conform to our notation. $S(W_t)$ is the KKT score function defined as $S(W) := \|\nabla f(W)\|_* + \lambda \langle W, \nabla f(W) \rangle$.

Related work	Measure	Convergence Rate	Nesterov momentum	Weight decay
Pethick et al. (2025)	$\mathbb{E}\left[\|\nabla f(W_T)\|_\star\right]$	$\mathcal{O}\left(\frac{1}{T} + \eta\right)$	$\times$	$\times$
Li & Hong (2025)	$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right]$	$\mathcal{O}\left(\frac{1}{T} + \frac{1}{\sqrt{b}} + n\right)$	$\times$	$\times$
Kovalev (2025)	$\min_{0\le t\le T-1} \mathbb{E}\left[\|\nabla f(W_t)\|_*\right]$	$\mathcal{O}\left(\frac{1}{T} + \eta + \sqrt{\beta}\right)$	$\times$	$\times$
Shen et al. (2025)	$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla f(W_t)\|_*\right]$	$\mathcal{O}\left(\frac{1}{T} + \frac{1}{\sqrt{b}} + r\eta\right)$	$\times$	$\times$
Chen et al. (2025)	$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[S(W_t)\right]$	$\mathcal{O}\left(\frac{1}{T} + \frac{1}{\sqrt{b}} + n\right)$	✓	✓
Lau et al. (2025)	$\frac{1}{T}\min_{0\le t\le T-1} \mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right]$	$\mathcal{O}\left(\frac{1}{T} + \eta + \sqrt{r}\right)$	$\times$	$\times$
Ours (Theorem 3.2(ii))	$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right]$	$\mathcal{O}\left(\frac{1}{T} + \frac{1}{\sqrt{b}} + n\eta\right)$	✓	✓

7 Conclusion

Through a comprehensive theoretical analysis of the Muon optimizer, we established convergence guarantees for four practical configurations (with and without Nesterov momentum and with and without weight decay). Our primary theoretical contribution is demonstrating the crucial role of weight decay. We proved that it enforces a strict decrease in parameter and gradient norms, a clear advantage over the standard Muon configuration. This theoretical insight, along with the necessary condition relating the learning rate and weight decay coefficient, was empirically validated by our experimental results. Additionally, we derived the critical batch size for Muon, revealing its dependence on fundamental hyperparameters such as momentum and weight decay. Collectively, our findings provide both a deeper theoretical understanding of Muon and actionable guidance for practitioners aiming to leverage this promising optimizer in large-scale settings.

Finally, we note three limitations. First, our single-matrix analysis does not directly model the layer-wise heterogeneity of gradient statistics in deep networks. However, Gray et al. (2024) show that gradient noise scales of different layers are highly correlated, suggesting that our theory captures the fundamental functional dependencies of the critical batch size on hyperparameters. Second, in practice Muon is used as a hybrid optimizer (Muon on matrix-shaped parameters, AdamW on the rest), whereas our theory assumes full Muon. We address this gap through a two-level validation strategy: a controlled full-Muon experiment that exactly matches the theoretical assumptions, and practical-scale hybrid experiments that confirm the qualitative predictions transfer to the hybrid regime (Section 5). Third, our experimental baselines are limited to AdamW and Momentum SGD. While other matrix-aware optimizers such as Shampoo (Gupta et al., 2018) and SOAP (Vyas et al., 2025) are closely related to Muon (Jordan et al., 2024), our primary goal is to validate the *theoretical predictions* (convergence ordering, CBS dependencies) rather than to benchmark Muon against all competitors; we therefore compare against the most widely used baseline (AdamW) and defer broader comparisons to future work.

References

Kwangjun Ahn, Byron Xu, Natalie Abreu, and John Langford. Dion: Distributed orthonormalized updates. *arXiv*, https://arxiv.org/abs/2504.05295, 2025.

Essential AI, 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*, https://arxiv.org/abs/2505.02222, 2025.

Kang An, Yuxing Liu, Rui Pan, Yi Ren, Shiqian Ma, Donald Goldfarb, and Tong Zhang. ASGO: Adaptive structured gradient optimization. *arXiv*, https://arxiv.org/abs/2503.20762, 2025.

Jeremy Bernstein. Deriving Muon. https://jeremybernste.in/writing/deriving-muon, 2025. Accessed: 2025-7-1.

Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology. *arXiv*, https://arxiv.org/abs/2409.20325, 2024.

Å. Björck and C. Bowie. An iterative algorithm for computing the best estimate of an orthogonal matrix. *SIAM Journal on Numerical Analysis*, 8(2):358–364, 1971.

David Carlson, Volkan Cevher, and Lawrence Carin. Stochastic spectral descent for restricted boltzmann machines. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, volume 38, pp. 111–119, 2015.

Lizhang Chen, Bo Liu, Kaizhao Liang, and Qiang Liu. Lion secretly solves a constrained optimization: As lyapunov predicts. In *International Conference on Learning Representations*, 2024.

Lizhang Chen, Jonathan Li, and Qiang Liu. Muon optimizes under spectral norm constraints. *arXiv*, https://arxiv.org/abs/2506.15054, 2025.

Thomas Flynn. The duality structure gradient descent algorithm: analysis and applications to neural networks. *arXiv*, https://arxiv.org/abs/1708.00523, 2017.

Gavia Gray, Shane Bergsma, Joel Hestness, et al. Normalization layer per-example gradients are sufficient to predict gradient noise scale in transformers. *Advances in Neural Information Processing Systems*, 37: 93510–93539, 2024.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 1837–1845, 2018.

Nicholas J. Higham. *Functions of matrices - theory and computation*. SIAM, 2008.

Kento Imaizumi and Hideaki Iiduka. Iteration and stochastic first-order oracle complexities of stochastic gradient descent using constant and decaying learning rates. *Optimization*, pp. 1–24, 2024.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. https://kellerjordan.github.io/posts/muon/, 2024. Accessed: 2025-7-3.

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

Dmitry Kovalev. Understanding gradient orthogonalization for deep learning via non-euclidean trust-region optimization. *arXiv*, https://arxiv.org/abs/2503.12645, 2025.

Zdislav Kovarik. Some iterative methods for improving orthonormality. *SIAM Journal on Numerical Analysis*, 7(3):386–389, 1970.

Tim Tsz-Kit Lau, Qi Long, and Weijie Su. PolarGrad: A class of matrix-gradient optimizers from a unifying preconditioning perspective. *arXiv*, https://arxiv.org/abs/2505.21799, 2025.

Jiaxiang Li and Mingyi Hong. A note on the convergence of Muon and further. *arXiv*, https://arxiv.org/abs/2502.02900, 2025.

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*, https://arxiv.org/abs/2502.16982, 2025a.

Liming Liu, Zhenghao Xu, Zixuan Zhang, Hao Kang, Zichong Li, Chen Liang, Weizhu Chen, and Tuo Zhao. COSMOS: A hybrid adaptive optimizer for memory-efficient training of llms. *arXiv*, https://arxiv.org/abs/2502.17410, 2025b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Siyuan Ma, Raef Bassily, and Mikhail Belkin. The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning. *Proceedings of the 35th International Conference on Machine Learning*, 80:3331–3340, 2018.

Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv*, https://arxiv.org/abs/1812.06162, 2018.

Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Stochastic conditional gradient methods: From convex minimization to submodular maximization. *Journal of Machine Learning Research*, 21(105):1–49, 2020.

Hiroki Naganuma, Shagun Gupta, Youssef Briki, Ioannis Mitliagkas, Irina Rish, Parameswaran Raman, and Hao-Jun Michael Shi. Adaptive batch sizes using non-Euclidean gradient noise scales for stochastic sign and spectral descent. *arXiv*, https://arxiv.org/abs/2602.03001, 2026.

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*, 2025.

Egor Petrov, Grigoriy Evseev, Aleksey Antonov, Andrey Veprikov, Pavel Plyusnin, Nikolay Bushkov, Stanislav Moiseev, and Aleksandr Beznosikov. Leveraging coordinate momentum in SignSGD and Muon: Memory-optimized zero-order. *arXiv*, https://arxiv.org/abs/2506.04430, 2025.

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

Naoki Sato and Hideaki Iiduka. Existence and estimation of critical batch size for training generative adversarial networks with two time-scale update rule. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 30080–30104, 2023.

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

Wei Shen, Ruichuan Huang, Minhui Huang, Cong Shen, and Jiawei Zhang. On the convergence analysis of Muon. *arXiv*, https://arxiv.org/abs/2505.23737, 2025.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham M. Kakade. SOAP: improving and stabilizing shampoo using adam for language modeling. In *International Conference on Learning Representations*, 2025.

Guodong Zhang, Lala Li, Zachary Nado, James Martens, Sushant Sachdeva, George E. Dahl, Christopher J. Shallue, and Roger B. Grosse. Which algorithmic choices matter at which batch sizes? insights from a noisy quadratic model. In *Advances in Neural Information Processing Systems*, volume 32, pp. 8194–8205, 2019.

Hanlin Zhang, Depen Morwani, Nikhil Vyas, Jingfeng Wu, Difan Zou, Udaya Ghai, Dean P. Foster, and Sham M. Kakade. How does critical batch size scale in pre-training? In *International Conference on Learning Representations*, 2025.

A Tools for Proof of All Theorems

The results presented in this section are not new and are given simply for reference and completeness.

Lemma A.1. *Suppose Assumption 2.2(ii) hold for all $t \in \mathbb{N}$; then,*

$$\mathbb{E}_{\boldsymbol{\xi}_t}\left[\|\nabla f_{\mathcal{S}_t}(W_t) - \nabla f(W_t)\|_{\mathrm{F}}^2\right] \leq \frac{C^2}{b}.$$

Proof. Assumption 2.2(ii) guarantee that

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\xi}_t}\left[\|\nabla f_{\mathcal{S}_t}(W_t) - \nabla f(W_t)\|_{\mathrm{F}}^2\right] &= \mathbb{E}_{\boldsymbol{\xi}_t}\left[\left\|\frac{1}{b}\sum_{i=1}^{b}\mathsf{G}_{\xi_{t,i}}(W_t) - \nabla f(W_t)\right\|_{\mathrm{F}}^2\right] \\
&= \mathbb{E}_{\boldsymbol{\xi}_t}\left[\left\|\frac{1}{b}\sum_{i=1}^{b}\mathsf{G}_{\xi_{t,i}}(W_t) - \frac{1}{b}\sum_{i=1}^{b}\nabla f(W_t)\right\|_{\mathrm{F}}^2\right] \\
&= \mathbb{E}_{\boldsymbol{\xi}_t}\left[\left\|\frac{1}{b}\sum_{i=1}^{b}\left(\mathsf{G}_{\xi_{t,i}}(W_t) - \nabla f(W_t)\right)\right\|_{\mathrm{F}}^2\right] \\
&= \frac{1}{b^2}\mathbb{E}_{\boldsymbol{\xi}_t}\left[\left\|\sum_{i=1}^{b}\left(\mathsf{G}_{\xi_{t,i}}(W_t) - \nabla f(W_t)\right)\right\|_{\mathrm{F}}^2\right] \\
&= \frac{1}{b^2}\mathbb{E}_{\boldsymbol{\xi}_t}\left[\sum_{i=1}^{b}\left\|\mathsf{G}_{\xi_{t,i}}(W_t) - \nabla f(W_t)\right\|_{\mathrm{F}}^2\right] \\
&\leq \frac{\sigma^2}{b}.
\end{aligned}$$

This completes the proof. □

The following lemma was established by (Mokhtari et al., 2020). In their setting, the algorithm without momentum corresponds to the case in which $\beta_1 = 1$, whereas in ours it corresponds to $\beta_1 = 0$. As a result, the statements may appear slightly different.

Lemma A.2. *Suppose Assumptions 2.1 and 2.2 hold. Then for all $t \in \mathbb{N}$,*

$$\sum_{t=0}^{T-1}\mathbb{E}\left[\|M_t - \nabla f(W_t)\|_{\mathrm{F}}\right] \leq \frac{4}{1-\beta}\|M_0 - \nabla f(W_0)\|_{\mathrm{F}} + \sqrt{\frac{2(1-\beta)\sigma^2}{b}}T + \frac{2L\eta\sqrt{n}}{1-\beta}T.$$

Proof. From the definition of M_t,

$$\begin{aligned}
\|M_t - \nabla f(W_t)\|_{\mathrm{F}}^2 &= \|\beta M_{t-1} + (1-\beta)\nabla f_{\mathcal{S}_t}(W_t) - \nabla f(W_t)\|_{\mathrm{F}}^2 \\
&= \|\beta(M_{t-1} - \nabla f(W_{t-1})) + \beta(\nabla f(W_{t-1}) - \nabla f(W_t)) + (1-\beta)(\nabla f_{\mathcal{S}_t}(W_t) - \nabla f(W_t))\|_{\mathrm{F}}^2 \\
&= \beta^2\|M_{t-1} - \nabla f(W_{t-1})\|_{\mathrm{F}}^2 + \beta^2\|\nabla f(W_{t-1}) - \nabla f(W_t)\|_{\mathrm{F}}^2 + (1-\beta)^2\|\nabla f_{\mathcal{S}_t}(W_t) - \nabla f(W_t)\|_{\mathrm{F}}^2 \\
&\quad + 2\beta^2\langle M_{t-1} - \nabla f(W_{t-1}), \nabla f(W_{t-1}) - \nabla f(W_t)\rangle_{\mathrm{F}} \\
&\quad + 2\beta(1-\beta)\langle M_{t-1} - \nabla f(W_{t-1}), \nabla f_{\mathcal{S}_t}(W_t) - \nabla f(W_t)\rangle_{\mathrm{F}} \\
&\quad + 2\beta(1-\beta)\langle \nabla f(W_{t-1}) - \nabla f(W_t), \nabla f_{\mathcal{S}_t}(W_t) - \nabla f(W_t)\rangle_{\mathrm{F}}
\end{aligned}$$

Therefore, by taking the expectation,

$$\begin{aligned}
\mathbb{E}\left[\|M_t - \nabla f(W_t)\|_{\mathrm{F}}^2\right] &= \beta^2\mathbb{E}\left[\|M_{t-1} - \nabla f(W_{t-1})\|_{\mathrm{F}}^2\right] + \beta^2\mathbb{E}\left[\|\nabla f(W_{t-1}) - \nabla f(W_t)\|_{\mathrm{F}}^2\right] \\
&\quad + (1-\beta)^2\mathbb{E}\left[\|\nabla f_{\mathcal{S}_t}(W_t) - \nabla f(W_t)\|_{\mathrm{F}}^2\right] + 2\beta^2\mathbb{E}\left[\langle M_{t-1} - \nabla f(W_{t-1}), \nabla f(W_{t-1}) - \nabla f(W_t)\rangle_{\mathrm{F}}\right].
\end{aligned}$$

Here, by the Peter-Paul inequality, for all $\epsilon > 0$, we have

$$\langle M_{t-1} - \nabla f(W_{t-1}), \nabla f(W_{t-1}) - \nabla f(W_t)\rangle_{\mathrm{F}} \leq \frac{\epsilon}{2}\|M_{t-1} - \nabla f(W_{t-1})\|_{\mathrm{F}}^2 + \frac{1}{2\epsilon}\|\nabla f(W_{t-1}) - \nabla f(W_t)\|_{\mathrm{F}}^2.$$

Therefore, we obtain

$$\begin{aligned}\mathbb{E}\left[\|M_t - \nabla f(W_t)\|_{\mathrm{F}}^2\right] &= \beta^2(1+\epsilon)\mathbb{E}\left[\|M_{t-1} - \nabla f(W_{t-1})\|_{\mathrm{F}}^2\right] + \beta^2\left(1+\frac{1}{\epsilon}\right)\mathbb{E}\left[\|\nabla f(W_{t-1}) - \nabla f(W_t)\|_{\mathrm{F}}^2\right] \\ &\quad + (1-\beta)^2\mathbb{E}\left[\|\nabla f_{\mathcal{S}_t}(W_t) - \nabla f(W_t)\|_{\mathrm{F}}^2\right].\end{aligned}$$

In addition, from Assumption 2.1,

$$\|\nabla f(W_{t-1}) - \nabla f(W_t)\|_{\mathrm{F}}^2 \leq L^2\|W_{t-1} - W_t\|_{\mathrm{F}}^2 = L^2\eta^2\|O_t\|_{\mathrm{F}}^2 \leq L^2\eta^2 n.$$

Hence, from Lemma A.1,

$$\mathbb{E}\left[\|M_t - \nabla f(W_t)\|_{\mathrm{F}}^2\right] \leq \beta^2(1+\epsilon)\mathbb{E}\left[\|M_{t-1} - \nabla f(W_{t-1})\|_{\mathrm{F}}^2\right] + \beta^2\left(1+\frac{1}{\epsilon}\right)L^2\eta^2 n + \frac{(1-\beta)^2\sigma^2}{b}.$$

Then, letting $\epsilon := \frac{1-\beta}{2}$, we have

$$\begin{aligned}\mathbb{E}\left[\|M_t - \nabla f(W_t)\|_{\mathrm{F}}^2\right] &\leq \frac{\beta^2(3-\beta)}{2}\mathbb{E}\left[\|M_{t-1} - \nabla f(W_{t-1})\|_{\mathrm{F}}^2\right] + \frac{\beta^2(3-\beta)}{1-\beta}L^2\eta^2 n + \frac{(1-\beta)^2\sigma^2}{b} \\ &\leq \frac{1+\beta}{2}\mathbb{E}\left[\|M_{t-1} - \nabla f(W_{t-1})\|_{\mathrm{F}}^2\right] + \frac{2}{1-\beta}L^2\eta^2 n + \frac{(1-\beta)^2\sigma^2}{b} \\ &\leq \left(\frac{1+\beta}{2}\right)^t\|M_0 - \nabla f(W_0)\|_{\mathrm{F}}^2 + \left\{\frac{2L^2\eta^2 n}{1-\beta} + \frac{(1-\beta)^2\sigma^2}{b}\right\}\sum_{k=0}^{t-1}\left(\frac{1+\beta}{2}\right)^k \\ &\leq \left(\frac{1+\beta}{2}\right)^t\|M_0 - \nabla f(W_0)\|_{\mathrm{F}}^2 + \left\{\frac{2L^2\eta^2 n}{1-\beta} + \frac{(1-\beta)^2\sigma^2}{b}\right\}\frac{2}{1-\beta} \\ &= \left(\frac{1+\beta}{2}\right)^t\|M_0 - \nabla f(W_0)\|_{\mathrm{F}}^2 + \frac{4L^2\eta^2 n}{(1-\beta)^2} + \frac{2(1-\beta)\sigma^2}{b}.\end{aligned}$$

Therefore, summing over t, we have

$$\sum_{t=0}^{T-1}\mathbb{E}\left[\|M_t - \nabla f(W_t)\|_{\mathrm{F}}^2\right] \leq \frac{2}{1-\beta}\|M_0 - \nabla f(W_0)\|_{\mathrm{F}}^2 + \frac{2(1-\beta)\sigma^2}{b}T + \frac{4L^2\eta^2 n}{(1-\beta)^2}T.$$

Finally, from the properties of variance and expectation,

$$\begin{aligned}\mathbb{E}\left[\|M_t - \nabla f(W_t)\|_{\mathrm{F}}\right] &\leq \sqrt{\mathbb{E}\left[\|M_t - \nabla f(W_t)\|_{\mathrm{F}}^2\right]} \\ &\leq \sqrt{\left(\frac{1+\beta}{2}\right)^t}\|M_0 - \nabla f(W_0)\|_{\mathrm{F}} + \sqrt{\frac{2(1-\beta)\sigma^2}{b}} + \frac{2L\eta\sqrt{n}}{(1-\beta)}.\end{aligned}$$

Hence, we have

$$\begin{aligned}\sum_{t=0}^{T-1}\mathbb{E}\left[\|M_t - \nabla f(W_t)\|_{\mathrm{F}}\right] &\leq \frac{\sqrt{2}}{\sqrt{2}-\sqrt{1+\beta}}\|M_0 - \nabla f(W_0)\|_{\mathrm{F}} + \sqrt{\frac{2(1-\beta)\sigma^2}{b}}T + \frac{2L\eta\sqrt{n}}{1-\beta}T \\ &\leq \frac{4}{1-\beta}\|M_0 - \nabla f(W_0)\|_{\mathrm{F}} + \sqrt{\frac{2(1-\beta)\sigma^2}{b}}T + \frac{2L\eta\sqrt{n}}{1-\beta}T\end{aligned}$$

This completes the proof. □

B Proof of Theorems for Muon without weight decay

Theorem B.1 (Auxiliary Theorem for Muon without weight decay)**.** *Suppose Assumptions 2.1 and 2.2 hold. Then for all $t \in \mathbb{N}$,*

$$\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right] \leq \frac{f(W_0) - f(W_T)}{\eta} + 2\sqrt{r} \sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla f(W_t) - C_t\|_{\mathrm{F}}\right] + \frac{L\eta}{2} nT,$$

where $\mathrm{rank}(C_t - \nabla f(W_t)) =: r_t \leq \max_{0 \leq t \leq T-1} r_t =: r$.

Proof. From Assumption 2.1,

$$\begin{aligned} f(W_{t+1}) &\leq f(W_t) + \langle \nabla f(W_t), W_{t+1} - W_t \rangle_{\mathrm{F}} + \frac{L}{2}\|W_{t+1} - W_t\|_{\mathrm{F}}^2 \\ &= f(W_t) - \eta \langle \nabla f(W_t), O_t \rangle_{\mathrm{F}} + \frac{L}{2}\eta^2 \|O_t\|_{\mathrm{F}}^2 \\ &\leq (W_t) - \eta \langle C_t, O_t \rangle_{\mathrm{F}} - \eta \langle \nabla f(W_t) - C_t, O_t \rangle_{\mathrm{F}} + \frac{L\eta^2 n}{2}. \end{aligned}$$

Using the definition $O_t := \operatorname{argmin}_{O \in \{O \in \mathbb{R}^{m \times n} : O^\top O = I_n\}} \|O - C_t\|_{\mathrm{F}}$, we obtain $O_t := \operatorname{argmax}_{O \in \{O \in \mathbb{R}^{m \times n} : O^\top O = I_n\}} \langle C_t, O \rangle_{\mathrm{F}}$. Then,

$$\langle C_t, O_t \rangle_{\mathrm{F}} = \max_{O : O^\top O = I_n} \langle C_t, O \rangle_{\mathrm{F}} = \max_{O : \|O\|_{\mathrm{op}} \leq 1} \langle C_t, O \rangle_{\mathrm{F}} =: \|C_t\|_*,$$

where $\|\cdot\|_*$ denotes the dual norm. Applying the reverse triangle inequality and the relation $\|A\|_{\mathrm{F}} \leq \|A\|_* \leq \sqrt{\mathrm{rank}(A)}\|A\|_{\mathrm{F}}$, we have

$$\begin{aligned} -\langle C_t, O_t \rangle_{\mathrm{F}} &= -\|C_t\|_* \\ &= -\|C_t - \nabla f(W_t) + \nabla f(W_t)\|_* \\ &\leq \|C_t - \nabla f(W_t)\|_* - \|\nabla f(W_t)\|_* \\ &\leq \sqrt{\mathrm{rank}(C_t - \nabla f(W_t))}\|C_t - \nabla f(W_t)\|_{\mathrm{F}} - \|\nabla f(W_t)\|_{\mathrm{F}} \\ &\leq \sqrt{r}\|C_t - \nabla f(W_t)\|_{\mathrm{F}} - \|\nabla f(W_t)\|_{\mathrm{F}}, \end{aligned} \tag{6}$$

where $\mathrm{rank}(C_t - \nabla f(W_t)) =: r_t \leq \max_{0 \leq t \leq T-1} r_t =: r \ (\leq n)$. In addition, since O_t has orthonormal columns ($O_t^\top O_t = I_n$), we have $\|O_t\|_{\mathrm{op}} = 1$. Therefore, by the duality between the nuclear norm and the operator norm ($|\langle A, B \rangle_{\mathrm{F}}| \leq \|A\|_* \|B\|_{\mathrm{op}}$),

$$\begin{aligned} |\langle \nabla f(W_t) - C_t, O_t \rangle_{\mathrm{F}}| &\leq \|\nabla f(W_t) - C_t\|_* \|O_t\|_{\mathrm{op}} \\ &\leq \sqrt{r_t}\, \|\nabla f(W_t) - C_t\|_{\mathrm{F}}, \end{aligned}$$

where we used $\|A\|_* \leq \sqrt{\mathrm{rank}(A)}\|A\|_{\mathrm{F}}$. Therefore,

$$f(W_{t+1}) \leq f(W_t) + 2\eta\sqrt{r}\|C_t - \nabla f(W_t)\|_{\mathrm{F}} - \eta\|\nabla f(W_t)\|_{\mathrm{F}} + \frac{L\eta^2}{2} n.$$

By rearranging the terms and taking expectation, we obtain

$$\mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right] \leq \frac{f(W_t) - f(W_{t+1})}{\eta} + 2\sqrt{r}\mathbb{E}\left[\|\nabla f(W_t) - C_t\|_{\mathrm{F}}\right] + \frac{L\eta}{2} n.$$

This completes the proof. □

B.1 Proof of Theorem 3.1(i)

Proof. From $C_t := M_t$, together with Lemma A.2 and Theorem B.1, we find that

$$\begin{aligned}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right] &\leq \frac{f(W_0)-f(W_T)}{\eta} + 2\sqrt{r}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(W_t)-M_t\|_{\mathrm{F}}\right] + \frac{L\eta}{2}nT \\ &\leq \frac{f(W_0)-f(W_T)}{\eta} + 2\sqrt{r}\left\{\frac{4}{1-\beta}\|M_0-\nabla f(W_0)\|_{\mathrm{F}} + \sqrt{\frac{2(1-\beta)\sigma^2}{b}}T + \frac{2L\eta\sqrt{n}}{1-\beta}T\right\} + \frac{L\eta}{2}nT.\end{aligned}$$

By taking the average over $t = 0, \ldots, T-1$ and applying expectation, we obtain

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right] &\leq \frac{f(W_0)-f(W_T)}{\eta T} + \frac{8\sqrt{r}\|M_0-\nabla f(W_0)\|_{\mathrm{F}}}{(1-\beta)T} + 2\sqrt{\frac{2(1-\beta)r\sigma^2}{b}} + \frac{4L\eta\sqrt{nr}}{1-\beta} + \frac{L\eta}{2}n \\ &= \mathcal{O}\left(\frac{1}{\eta T} + \sqrt{\frac{r(1-\beta)}{b}} + \eta n\right).\end{aligned}$$

This completes the proof. □

B.2 Proof of Theorem 3.1(ii)

Proof. From the definition of C_t, we have

$$\begin{aligned}\|C_t - \nabla f(W_t)\|_{\mathrm{F}} &= \|\beta M_t + (1-\beta)\nabla f_{\mathcal{S}_t}(W_t) - \nabla f(W_t)\|_{\mathrm{F}} \\ &= \|\beta(M_t - \nabla f(W_t)) + (1-\beta)(\nabla f_{\mathcal{S}_t}(W_t) - \nabla f(W_t))\|_{\mathrm{F}} \\ &\leq \beta\|M_t - \nabla f(W_t)\|_{\mathrm{F}} + (1-\beta)\|\nabla f_{\mathcal{S}_t}(W_t) - \nabla f(W_t)\|_{\mathrm{F}}\end{aligned}$$

According to Theorem B.1, we find that

$$\begin{aligned}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right] &\leq \frac{f(W_0)-f(W_T)}{\eta} + 2\sqrt{r}\sum_{t=0}^{T-1}\mathbb{E}\left[\|C_t - \nabla f(W_t)\|_{\mathrm{F}}\right] + \frac{L\eta}{2}nT \\ &\leq \frac{f(W_0)-f(W_T)}{\eta} + 2\sqrt{r}\left\{\beta\sum_{t=0}^{T-1}\mathbb{E}\left[\|M_t - \nabla f(W_t)\|_{\mathrm{F}}\right] + (1-\beta)\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f_{\mathcal{S}_t}(W_t) - \nabla f(W_t)\|_{\mathrm{F}}\right]\right\} \\ &\quad + \frac{L\eta}{2}nT.\end{aligned}$$

From Lemmas A.1 and A.2, we have

$$\begin{aligned}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right] &\leq \frac{f(W_0)-f(W_T)}{\eta} + 2\beta\sqrt{r}\left\{\frac{4}{1-\beta}\|M_0-\nabla f(W_0)\|_{\mathrm{F}} + \sqrt{\frac{2(1-\beta)\sigma^2}{b}}T + \frac{2L\eta\sqrt{n}}{1-\beta}T\right\} \\ &\quad + 2(1-\beta)\sqrt{r}\cdot\sqrt{\frac{\sigma^2}{b}}T + \frac{L\eta}{2}nT.\end{aligned}$$

By taking the average over $t = 0, \ldots, T-1$ and applying expectation, we obtain

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right] &\leq \frac{f(W_0)-f(W_T)}{\eta T} + \frac{8\beta\sqrt{r}\|M_0-\nabla f(W_0)\|_{\mathrm{F}}}{(1-\beta)T} \\ &\quad + 2(1+\sqrt{2}\beta)\sqrt{1-\beta}\sqrt{\frac{r\sigma^2}{b}} + \frac{4L\eta\sqrt{nr}\beta}{1-\beta} + \frac{L\eta}{2}n \\ &= \mathcal{O}\left(\frac{1}{T} + \beta\sqrt{\frac{r(1-\beta)}{b}} + \eta n\right),\end{aligned}$$

where we use $2(\beta\sqrt{2(1-\beta)} + (1-\beta)) \leq 2(1+\sqrt{2}\beta)\sqrt{1-\beta}$. This completes the proof. □

C Proof of Theorems for Muon with weight decay

C.1 Proof of Proposition 3.1

Proof. From the definition of W_t and the condition $\eta \leq \frac{1}{\lambda}$, we have

$$\begin{aligned}
\|W_t\|_{\mathrm{F}} &= \|(1-\eta\lambda)W_{t-1} - \eta O_t\|_{\mathrm{F}} \\
&\leq (1-\eta\lambda)\|W_{t-1}\|_{\mathrm{F}} + \eta\sqrt{n} \\
&\leq (1-\eta\lambda)^t\|W_0\|_{\mathrm{F}} + \eta\sqrt{n}\sum_{k=0}^{t-1}(1-\eta\lambda)^k \\
&\leq (1-\eta\lambda)^t\|W_0\|_{\mathrm{F}} + \frac{\sqrt{n}}{\lambda} \\
&\leq (1-\eta\lambda)^t\|W_0\|_{\mathrm{F}} + \frac{\sqrt{n}}{\lambda}.
\end{aligned}$$

This completes the proof. □

C.2 Proof of Proposition 3.2

Proof. According to Assumption 2.1,

$$\|\nabla f(W_t) - \nabla f(W^\star)\|_{\mathrm{F}} \leq L\|W_t - W^\star\|_{\mathrm{F}}.$$

Therefore, from Proposition 3.1 and the fact that $\nabla f(W^\star) = \mathbf{0}$, we have

$$\begin{aligned}
\|\nabla f(W_t)\|_{\mathrm{F}} &\leq L\|W_t - W^\star\|_{\mathrm{F}} \\
&\leq L\|W_t\|_{\mathrm{F}} + L\|W^\star\|_{\mathrm{F}} \\
&\leq L(1-\eta\lambda)^t\|W_0\|_{\mathrm{F}} + \frac{L\sqrt{n}}{\lambda} + L\|W^\star\|_{\mathrm{F}}.
\end{aligned}$$

This completes the proof. □

Lemma C.1. *Suppose Assumptions 2.1 and 2.2 hold, and that Muon is run with $\eta \leq \frac{1}{\lambda}$. Then, for all $t \in \mathbb{N}$,*

$$\|W_t\|_{\mathrm{op}} \leq (1-\eta\lambda)^t\|W_0\|_{\mathrm{op}} + \frac{1-(1-\eta\lambda)^t}{\lambda}.$$

In addition, if $\lambda < \frac{1}{\|W_0\|_{\mathrm{op}}}$ holds, then

$$\|W_t\|_{\mathrm{op}} < 1.$$

Proof. From the definition of W_t and $\|O_t\|_{\mathrm{op}} = 1$, we have

$$\begin{aligned}
\|W_t\|_{\mathrm{op}} &\leq (1-\eta\lambda)\|W_{t-1}\|_{\mathrm{op}} + \eta\|O_t\|_{\mathrm{op}} \\
&= (1-\eta\lambda)^t\|W_0\|_{\mathrm{op}} + \eta\sum_{j=0}^{t-1}(1-\eta\lambda)^j \\
&= (1-\eta\lambda)^t\|W_0\|_{\mathrm{op}} + \frac{1-(1-\eta\lambda)^t}{\lambda}.
\end{aligned}$$

In addition, if $\lambda < \frac{1}{\|W_0\|_{\mathrm{op}}}$, i.e., $\|W_0\|_{\mathrm{op}} < \frac{1}{\lambda}$,

$$\|W_t\|_{\mathrm{op}} < (1-\eta\lambda)^t\frac{1}{\lambda} + \frac{1-(1-\eta\lambda)^t}{\lambda} = 1.$$

This completes the proof. □

Theorem C.1 (Auxiliary Theorem for Muon with weight decay)**.** *Suppose Assumptions 2.1 and 2.2 hold, and that Muon is run with* $\eta \leq \frac{1}{\lambda}$*. If* $\lambda < \min\left\{\frac{1}{\|W_0\|_{\mathrm{op}}}, 1\right\}$*, then for all* $t \in \mathbb{N}$*,*

$$\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right] \leq \frac{f(W_0) - f(W_T)}{(1-\lambda)\eta} + \frac{1 - \lambda + (1+\lambda)\sqrt{r}}{1-\lambda} \sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla f(W_t) - C_t\|_{\mathrm{F}}\right] + \frac{(1+\lambda^2)L\eta n T}{1-\lambda}$$

where $\mathrm{rank}(C_t - \nabla f(W_t)) =: r_t \leq \max_{0 \leq t \leq T-1} r_t =: r$*.*

Proof. According to Assumption 2.1,

$$\begin{aligned}
f(W_{t+1}) &\leq f(W_t) + \langle \nabla f(W_t), W_{t+1} - W_t \rangle_{\mathrm{F}} + \frac{L}{2}\|W_{t+1} - W_t\|_{\mathrm{F}}^2 \\
&= f(W_t) - \eta \langle \nabla f(W_t), O_t + \lambda W_t \rangle_{\mathrm{F}} + \frac{L\eta^2}{2}\|O_t + \lambda W_t\|_{\mathrm{F}}^2 \\
&= f(W_t) - \eta \langle C_t, O_t + \lambda W_t \rangle_{\mathrm{F}} - \eta \langle \nabla f(W_t) - C_t, O_t + \lambda W_t \rangle_{\mathrm{F}} + \frac{L\eta^2}{2}\|O_t + \lambda W_t\|_{\mathrm{F}}^2.
\end{aligned}$$

First, we bound the first inner product term. From Eq.(6) and Lemma C.1, we have

$$-\langle C_t, O_t + \lambda W_t \rangle_{\mathrm{F}} = -\langle C_t, O_t \rangle_{\mathrm{F}} - \lambda \langle C_t, W_t \rangle_{\mathrm{F}} = -\|C_t\|_* + \lambda \|C_t\|_* \|W_t\|_{\mathrm{op}} \leq -(1-\lambda)\|C_t\|_*.$$

From the reverse triangle inequality, we have $-\|C_t\|_{\mathrm{F}} \leq \|C_t - \nabla f(W_t)\|_{\mathrm{F}} - \|\nabla f(W_t)\|_{\mathrm{F}}$. Hence,

$$-\eta \langle C_t, O_t + \lambda W_t \rangle_{\mathrm{F}} \leq (1-\lambda)\eta \|C_t - \nabla f(W_t)\|_{\mathrm{F}} - (1-\lambda)\eta \|\nabla f(W_t)\|_{\mathrm{F}}.$$

Next, we bound the second inner product term. From Lemma C.1 and $\|O_t\|_{\mathrm{op}} = 1$, we have

$$\begin{aligned}
-\langle \nabla f(W_t) - C_t, O_t + \lambda W_t \rangle_{\mathrm{F}} &= -\langle \nabla f(W_t) - C_t, O_t \rangle_{\mathrm{F}} - \lambda \langle \nabla f(W_t) - C_t, W_t \rangle_{\mathrm{F}} \\
&\leq \|\nabla f(W_t) - C_t\|_* \|O_t\|_{\mathrm{op}} + \lambda \|\nabla f(W_t) - C_t\|_* \|W_t\|_{\mathrm{op}} \\
&\leq \sqrt{r}\|\nabla f(W_t) - C_t\|_{\mathrm{F}} + \lambda\sqrt{r}\|\nabla f(W_t) - C_t\|_{\mathrm{F}} \\
&= (1+\lambda)\sqrt{r}\|\nabla f(W_t) - C_t\|_{\mathrm{F}}.
\end{aligned}$$

Finally, we bound the last term. From Lemma C.1, we have $\|W_t\|_{\mathrm{F}} \leq \sqrt{n}\|W_t\|_{\mathrm{op}} \leq \sqrt{n}$. Hence,

$$\frac{L\eta^2}{2}\|O_t + \lambda W_t\|_{\mathrm{F}}^2 = L\eta^2\|O_t\|_{\mathrm{F}}^2 + L\eta^2\lambda^2\|W_t\|_{\mathrm{F}}^2 \leq L\eta^2 n + L\eta^2\lambda^2 n = (1+\lambda^2)L\eta^2 n.$$

From all of the above, we obtain

$$\begin{aligned}
f(W_{t+1}) &\leq f(W_t) + (1-\lambda)\eta\|C_t - \nabla f(W_t)\|_{\mathrm{F}} - (1-\lambda)\eta\|\nabla f(W_t)\|_{\mathrm{F}} \\
&\quad + \eta(1+\lambda)\sqrt{r}\|\nabla f(W_t) - C_t\|_{\mathrm{F}} + (1+\lambda^2)L\eta^2 n \\
&\leq f(W_t) + \eta\left\{1 - \lambda + (1+\lambda)\sqrt{r}\right\}\|\nabla f(W_t) - C_t\|_{\mathrm{F}} - (1-\lambda)\eta\|\nabla f(W_t)\|_{\mathrm{F}} + (1+\lambda^2)L\eta^2 n.
\end{aligned}$$

By rearranging the terms and taking the expectation, we obtain

$$\mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right] \leq \frac{f(W_t) - f(W_{t+1})}{(1-\lambda)\eta} + \frac{1 - \lambda + (1+\lambda)\sqrt{r}}{1-\lambda}\mathbb{E}\left[\|\nabla f(W_t) - C_t\|_{\mathrm{F}}\right] + \frac{(1+\lambda^2)L\eta n}{1-\lambda}.$$

This completes the proof. □

C.3 Proof of Theorem 3.2(i)

Proof. From $C_t := M_t$ and Lemma A.2 and Theorem C.1, we find that

$$\begin{aligned}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right] &\leq \frac{f(W_0)-f(W_T)}{(1-\lambda)\eta} + \hat{\lambda}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(W_t)-M_t\|_{\mathrm{F}}\right] + \frac{(1+\lambda^2)L\eta nT}{1-\lambda} \\ &\leq \frac{f(W_0)-f(W_T)}{(1-\lambda)\eta} + \hat{\lambda}\left\{\frac{4}{1-\beta}\|M_0-\nabla f(W_0)\|_{\mathrm{F}} + \sqrt{\frac{2(1-\beta)\sigma^2}{b}}T + \frac{2L\eta\sqrt{n}}{1-\beta}T\right\} \\ &\quad + \frac{(1+\lambda^2)L\eta nT}{1-\lambda},\end{aligned}$$

where $\hat{\lambda} := \frac{1-\lambda+(1+\lambda)\sqrt{r}}{1-\lambda}$. By taking the average over $t = 0, \ldots, T-1$, we obtain

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right] &\leq \frac{f(W_0)-f(W_T)}{(1-\lambda)\eta T} + \frac{8\sqrt{r}\|M_0-\nabla f(W_0)\|}{(1-\beta)T} + \frac{2\sqrt{2(1-\beta)}}{1-\lambda}\sqrt{\frac{r\sigma^2}{b}} \\ &\quad + \frac{4L\eta\sqrt{rn}}{(1-\beta)(1-\lambda)} + \frac{(1+\lambda^2)L\eta n}{(1-\lambda)} \\ &= \mathcal{O}\left(\frac{1}{\eta T} + \sqrt{\frac{r(1-\beta)}{b}} + \eta n\right),\end{aligned}$$

where we use $\hat{\lambda} \leq \frac{2\sqrt{r}}{1-\lambda}$. This completes the proof. □

C.4 Proof of Theorem 3.2(ii)

Proof. From the definition of C_t,

$$\|C_t - \nabla f(W_t)\|_{\mathrm{F}} \leq \beta\|M_t - \nabla f(W_t)\|_{\mathrm{F}} + (1-\beta)\|\nabla f_{\mathcal{S}_t}(W_t) - \nabla f(W_t)\|_{\mathrm{F}}.$$

Therefore, according to Lemma A.1 and Theorem C.1, we find that

$$\begin{aligned}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right] &\leq \frac{f(W_0)-f(W_T)}{(1-\lambda)\eta} + \frac{(1+\lambda^2)L\eta nT}{1-\lambda} + \hat{\lambda}(1-\beta)\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f_{\mathcal{S}_t}(W_t) - \nabla f(W_t)\|_{\mathrm{F}}\right] \\ &\quad + \hat{\lambda}\beta\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(W_t) - M_t\|_{\mathrm{F}}\right] \\ &\leq \frac{f(W_0)-f(W_T)}{(1-\lambda)\eta} + \frac{(1+\lambda^2)L\eta nT}{1-\lambda} + \frac{\hat{\lambda}(1-\beta)\sigma}{\sqrt{b}}T \\ &\quad + \hat{\lambda}\beta\left\{\frac{4}{1-\beta}\|M_0-\nabla f(W_0)\|_{\mathrm{F}} + \sqrt{\frac{2(1-\beta)\sigma^2}{b}}T + \frac{2L\eta\sqrt{n}}{1-\beta}T\right\},\end{aligned}$$

where $\hat{\lambda} := \frac{1-\lambda+(1+\lambda)\sqrt{r}}{1-\lambda}$. By taking the average over $t = 0, \ldots, T-1$, we obtain

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(W_t)\|_{\mathrm{F}}\right] &\leq \frac{f(W_0)-f(W_T)}{(1-\lambda)\eta T} + \frac{8\beta\sqrt{r}\|M_0-\nabla f(W_0)\|_{\mathrm{F}}}{(1-\beta)(1-\lambda)T} + \frac{2(1+\sqrt{2}\beta)\sqrt{1-\beta}}{1-\lambda}\sqrt{\frac{r\sigma^2}{b}} \\ &\quad + \frac{(1+\lambda^2)L\eta n}{1-\lambda} + \frac{4L\beta\eta\sqrt{rn}}{(1-\beta)(1-\lambda)} \\ &= \mathcal{O}\left(\frac{1}{\eta T} + \sqrt{\frac{r(1-\beta)}{b}} + \eta n\right),\end{aligned}$$

where we use $(\beta\sqrt{2(1-\beta)} + (1-\beta)) \leq (1+\sqrt{2}\beta)\sqrt{1-\beta}$ and $\hat{\lambda} \leq \frac{2\sqrt{r}}{1-\lambda}$. This completes the proof. □

D Experimental Details

This section details the experimental setup for evaluating the optimizers. Our code will be available at `https://anonymous.4open.science/r/critical_batchsize_muon-0D38/README.md`.

Workloads and General Setup (Vision) All CIFAR experiments used ResNet-18 on CIFAR-10 and VGG-16 on CIFAR-100, trained from scratch. Training was performed for a fixed number of samples corresponding to 100 epochs at a batch size of 512. For any batch size B, the number of epochs was scaled as $E_B = 100 \times (512/B)$ to keep the number of seen samples constant. Each configuration was repeated with five random seeds, and we report mean and standard deviation of test accuracy.

Workload and System Setup (Language Modeling: C4/ Llama3.1 (160M)) For the language–modeling workload we trained Llama3.1 (160M) on the C4 dataset with sequence length 2048. We used the `torchtitan` codebase with PyTorch FSDP (full sharding) and activation checkpointing (`mode=full`). Gradient clipping with max norm 1.0 was applied. Due to sharding, Muon requires an additional *all-gather of gradients* before the optimizer step, which introduces one extra collective communication per step. We therefore report SFO-based metrics (steps $\times$ batch size) in addition to loss to separate algorithmic efficiency from system overhead.

Unless otherwise noted, the training budget was fixed to 3.2B tokens. At a base global batch size (GBS) of 128 this corresponds to 12,208 steps; for other GBS values the number of steps scales inversely to keep the token budget constant. We ran on up to 8 H100 GPUs with per-GPU micro-batch size 8 and no tensor, pipeline, or context parallelism (`TP=PP=CP=1`).

Learning-Rate Schedules Vision workloads used a grid search at base batch size 512 and square-root scaling to other batch sizes for Muon and AdamW. Momentum SGD was tested with both square-root and linear scaling. For Llama3 160M, we used linear warmup followed by cosine decay. The base configuration uses 2000 warmup steps; in sweeps we also parameterized warmup as 10% of total steps to maintain a comparable schedule across different GBS.

Hyperparameter Tuning Protocol (Vision) We performed an extensive grid search over base learning rates and weight decay. For $B \neq 512$, learning rates were scaled from the base value using $\eta_B = \eta_{512}\sqrt{B/512}$ for AdamW and Muon; for Momentum SGD we report both $\sqrt{B/512}$ and $B/512$ scaling. The shared search space is summarized in Table 3, and optimizer-specific spaces in Table 4.

Table 3: Shared hyperparameters for vision experiments.

Hyperparameter	Value / Search Space
Model Architecture	ResNet-18, VGG-16
Dataset	CIFAR-10, CIFAR-100
Batch Size (B)	{2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096}
Base Epochs (at $B = 512$)	100
Random Seeds	5

Optimizer Configuration (Vision) For the hybrid Muon optimizer, we applied the Muon update to all convolutional layers in ResNet-18 (excluding the input stem). Biases, batch-norm parameters, and the final linear layer were updated by AdamW. The number of parameters updated by each component is listed in Table 5. For standard AdamW and Momentum SGD, the respective optimizer was applied to all parameters.

Search Space (Language Modeling) For C4/ Llama3.1 (160M), we performed sweeps using grid search. GBS took values in $\{64, 256, 1024, 4096\}$ when resources permitted. We tuned Muon and the AdamW baseline per GBS without automatic LR scaling.

Table 4: Optimizer-specific hyperparameter search spaces (vision). The base learning rate (η_{512}) is specified at a reference batch size of 512.

Hyperparameter	AdamW	Momentum SGD	Muon
Searched Parameters			
Base Learning Rate (η_{512})	{0.01, 0.001, 0.0001}	{0.001, 0.0005, 0.0001}	
- Muon component	—	—	{0.01, 0.005, 0.001}
- AdamW component	—	—	{0.01, 0.001, 0.0001}
Momentum	—	0.9	{0.7, 0.8, 0.9, 0.95, 0.99, 0.999}
Weight Decay (λ)	{0.1, 0.01, 0.001, 0.0001, 0}	{0.1, 0.01, 0.001, 0.0001, 0}	{0.1, 0.01, 0.001, 0.0001, 0}
Fixed Parameters			
Learning Rate Scaling	$\sqrt{B/512}$	$\sqrt{B/512}$ (and $B/512$)	$\sqrt{B/512}$
Adam β_1, β_2	0.9, 0.999 (default)	—	0.9, 0.999 (default)
Adam ϵ	1e-8 (default)	—	1e-8 (default)

Table 5: Number of parameters updated by each optimizer component in the Muon setup for ResNet-18, VGG-16 and Llama3.1 (160M).

Model	Muon params	AdamW params
ResNet-18	11,157,504	16,458
VGG-16	14,712,896	19,302,500
Llama3.1 (160M)	127,401,984	49,180,416

Table 6: Shared configuration for C4/ Llama3.1 (160M).

Item	Value
Model	Llama3.1 (160M) (dim = 768, $n_{\text{layers}} = 18$, $n_{\text{heads}} = 12$)
Dataset	C4
Sequence length	2048
Global batch size (GBS)	$\{32, 64, 128, 256, 512, 1024, 2048, 4096\}$
Total tokens	3.2×10^9
Base steps (GBS=128)	12,208
Schedule	linear warmup then cosine decay (decay ratio 0.5)
Warmup	2000 steps (base) or 10% of total steps (sweep)
Gradient clipping	max-norm = 1.0
Parallelism	FSDP; TP=1, PP=1, CP=1
Hardware	up to 8 H100 GPUs; local batch size = 8

Table 7: Optimizer search spaces for C4/ Llama3.1 (160M). Component-wise LRs are tuned per GBS.

Hyperparameter	Muon	AdamW (baseline)
Muon LR	{0.005, 0.01, 0.02, 0.04}	—
Muon momentum β	{0.8, 0.9, 0.95, 0.99}	—
Muon Nesterov	{True, False}	—
Muon weight decay	{0, 0.1}	—
AdamW LR (component or baseline)	{0.001, 0.002, 0.004, 0.008}	{0.001, 0.002, 0.004, 0.008}
AdamW weight decay	0.1	0.1
AdamW β_1, β_2	0.9, 0.999	0.9, 0.999
AdamW ϵ	1×10^{-8}	1×10^{-8}

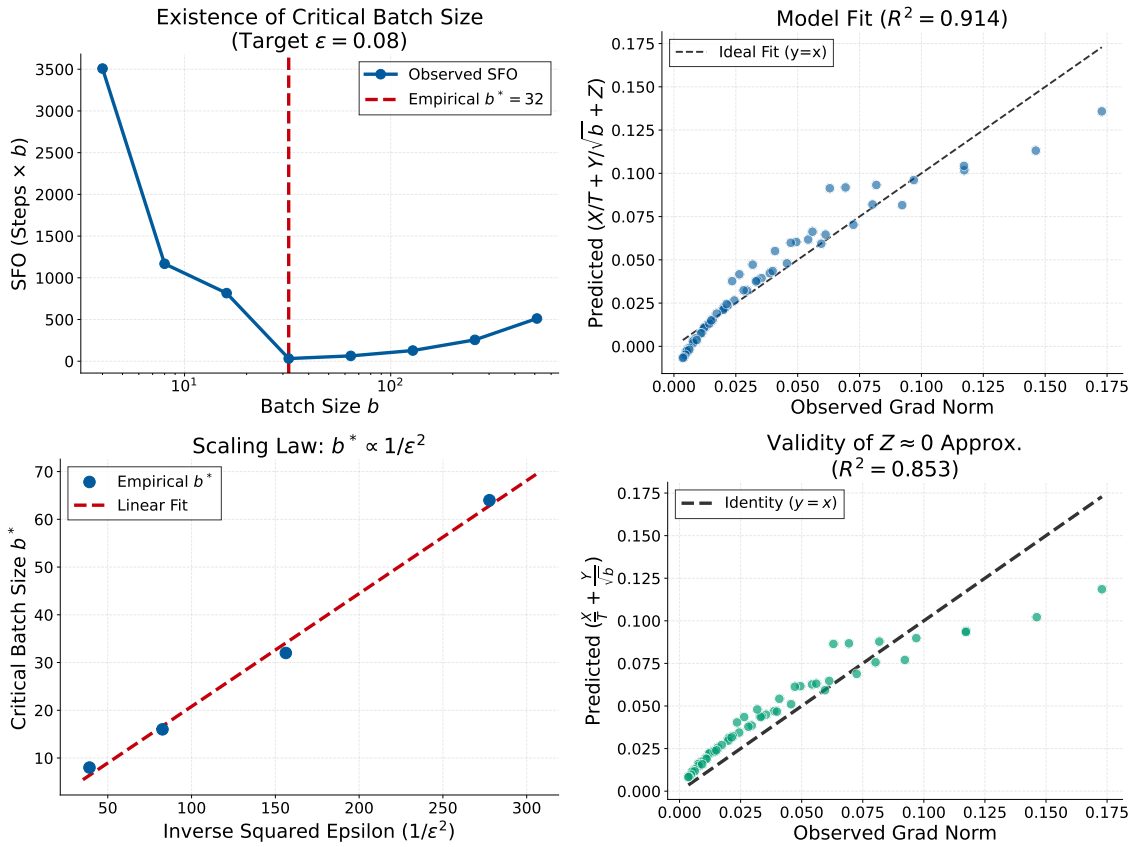

Figure 7: **Validation of critical batch size theory on a controlled full-Muon task. (Top Left)** SFO vs. batch size b required to reach a target gradient norm $\varepsilon = 0.08$. A clear empirical optimum (critical batch size) is observed at $b^\star = 32$. **(Top Right)** Goodness-of-fit for the theoretical proxy $\bar{g}(T, b) \approx X/T + Y/\sqrt{b} + Z$ against observed gradient norms ($R^2 \approx 0.914$). **(Bottom Left)** The empirical critical batch size $b^\star$ scales with the inverse squared target precision $1/\varepsilon^2$, consistent with the theoretical prediction $b^\star \propto 1/\varepsilon^2$ from Proposition 4.2. **(Bottom Right)** Goodness-of-fit for the theoretical proxy $\bar{g}(T, b) \approx X/T + Y/\sqrt{b}$ against observed gradient norms ($R^2 \approx 0.853$).

E Additional Diagnostics: Gradient-norm proxy and full-Muon toy setting

Toy task and full-Muon configuration. As described in Section 5, we use a controlled full-Muon experiment to provide an exact test of the theoretical predictions, free from the hybrid optimizer confound. We conduct experiments on a controlled synthetic task where Muon is applied to all parameters. We employ a **Teacher-Student Tanh Regression** problem. A fixed teacher matrix $W^\star \in \mathbb{R}^{m\times n}$ and a student matrix $W \in \mathbb{R}^{m\times n}$ are initialized with entries drawn from $\mathcal{N}(0, 1/n)$. At each step t, we sample inputs $x_t \sim \mathcal{N}(0, I_n)$ and generate labels $y_t = \tanh(x_t(W^\star)^\top) + \xi_t$, where $\xi_t \sim \mathcal{N}(0, \sigma^2 I)$ represents label noise. The student minimizes the loss $\mathcal{L}(W) = \frac{1}{2}\|\tanh(x_t W^\top) - y_t\|^2$. We set the dimensions to $m = 256, n = 128$, and the noise level to $\sigma = 0.1$. Unlike the large-scale experiments, here Muon (Algorithm 1) is applied to the entire matrix W with learning rate $\eta = 0.05$, momentum $\beta = 0.95$ (Nesterov), and 5 Newton-Schulz iterations.

Measured quantities and stopping rule. To rigorously validate the theoretical convergence bound $\mathbb{E}[\|\nabla\mathcal{L}\|] \lesssim \frac{X}{T} + \frac{Y}{\sqrt{b}} + Z$, we track the *cumulative mean* of the gradient norm, denoted as $\bar{g}_t = \frac{1}{t}\sum_{i=1}^{t}\|\nabla\mathcal{L}(W_i)\|_F$. We define the convergence step $T_\varepsilon(b)$ as the first step t where $\bar{g}_t \le \varepsilon$. The stochastic

first-order oracle (SFO) complexity is computed as SFO $= b \cdot T_\varepsilon(b)$. We perform a grid search over batch sizes $b \in \{4, 8, \ldots, 512\}$.

Results and Validity of Approximations. We confirm three key findings in this controlled setting: (1) **Existence of Critical Batch Size:** As shown in Figure 7 (Top Left), the SFO complexity exhibits a clear convex shape with respect to b, identifying an empirical optimum $b^\star$. (2) **Model Fit:** The observed data fits the theoretical proxy $\frac{X}{T} + \frac{Y}{\sqrt{b}} + Z$ with high accuracy ($R^2 \approx 0.914$), validating our convergence analysis. (3) **Justification for $Z \approx 0$:** To address the theoretical concern regarding the non-vanishing term Z, we compared the full fit against a simplified model $\frac{X}{T} + \frac{Y}{\sqrt{b}}$ (forcing $Z = 0$). Figure 7 (Bottom Right) demonstrates that the simplified model still predicts the observed gradient norms with reasonable accuracy ($R^2 \approx 0.853$), supporting the practical validity of the $Z \approx 0$ approximation. This empirically justifies our derivation of the critical batch size $b^\star \propto 1/\varepsilon^2$ by treating Z as negligible during the primary optimization phase.

F Additional Results (Vision)

This section provides supplementary results to support the analysis in the main text.

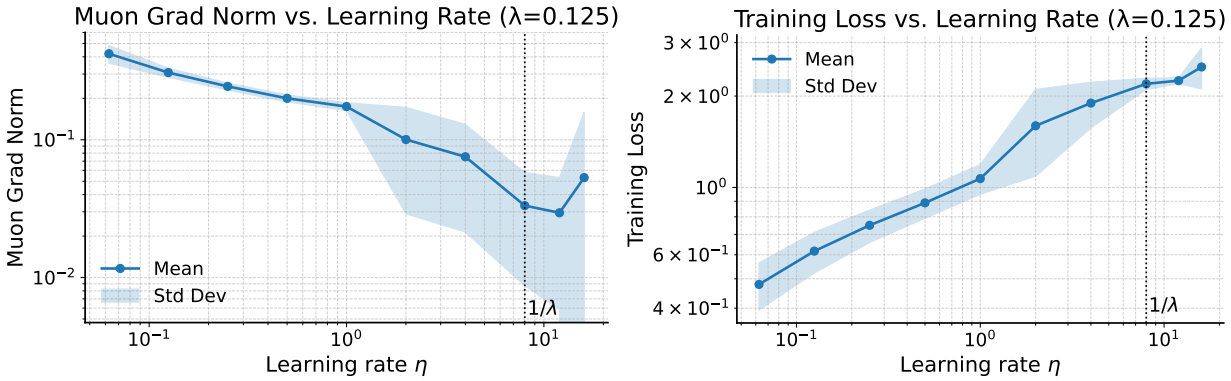

Figure 8: Empirical validation of the stability condition derived in Proposition 3.2. The plots show the final gradient norm (left) and training loss (right) for ResNet-18 on CIFAR-10, trained with Muon using various learning rates (η) and a fixed weight decay $\lambda = 0.125$. The vertical dashed lines mark the theoretical stability threshold $\eta = 1/\lambda$. Training was most stable and achieved the best performance near this threshold, consistent with our theoretical analysis.

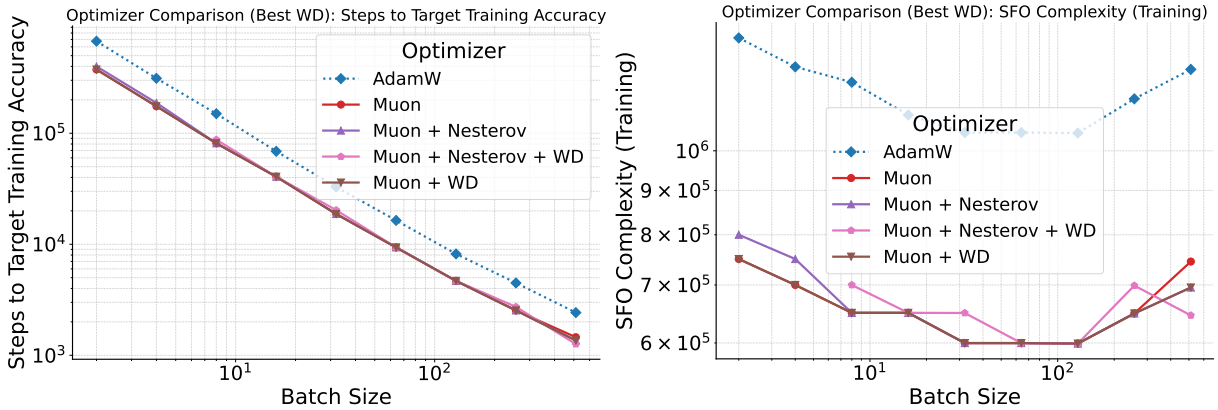

Figure 9: Analysis of batch size scaling and SFO complexity for ResNet-18 on CIFAR-10. (Left) Number of steps required to reach target training accuracy (95%). (Right) SFO complexity to reach the same accuracy. Muon exhibited superior scaling to large batch sizes, and its critical batch size (which minimizes SFO complexity (training)) was smaller than that of AdamW.

Convergence (Vision) We conducted experiments under various weight decay configurations to examine whether the upper bound of the learning rate is determined by the weight decay ($\lambda = 0.125$), similar to the analysis presented in Figure 1 of the main text ($\lambda = 0.0625$). As shown in Figure 8, the results are consistent with the earlier findings, confirming that the gradient norm begins to increase once the learning rate exceeds $1/\lambda$.

Figure 10 compares convergence rates for ResNet-18 on CIFAR-10 across three batch sizes. This complements Figure 2 by providing a more detailed view of the effect of batch size.

Critical Batch Size (Vision) In the main text, we defined the critical batch size as the smallest batch size that reaches the test target accuracy in the fewest steps. For comparison, we also report results using the training target accuracy in Figure 9. While the overall trends remain similar, the gap between AdamW and Muon becomes significantly larger in this setting.

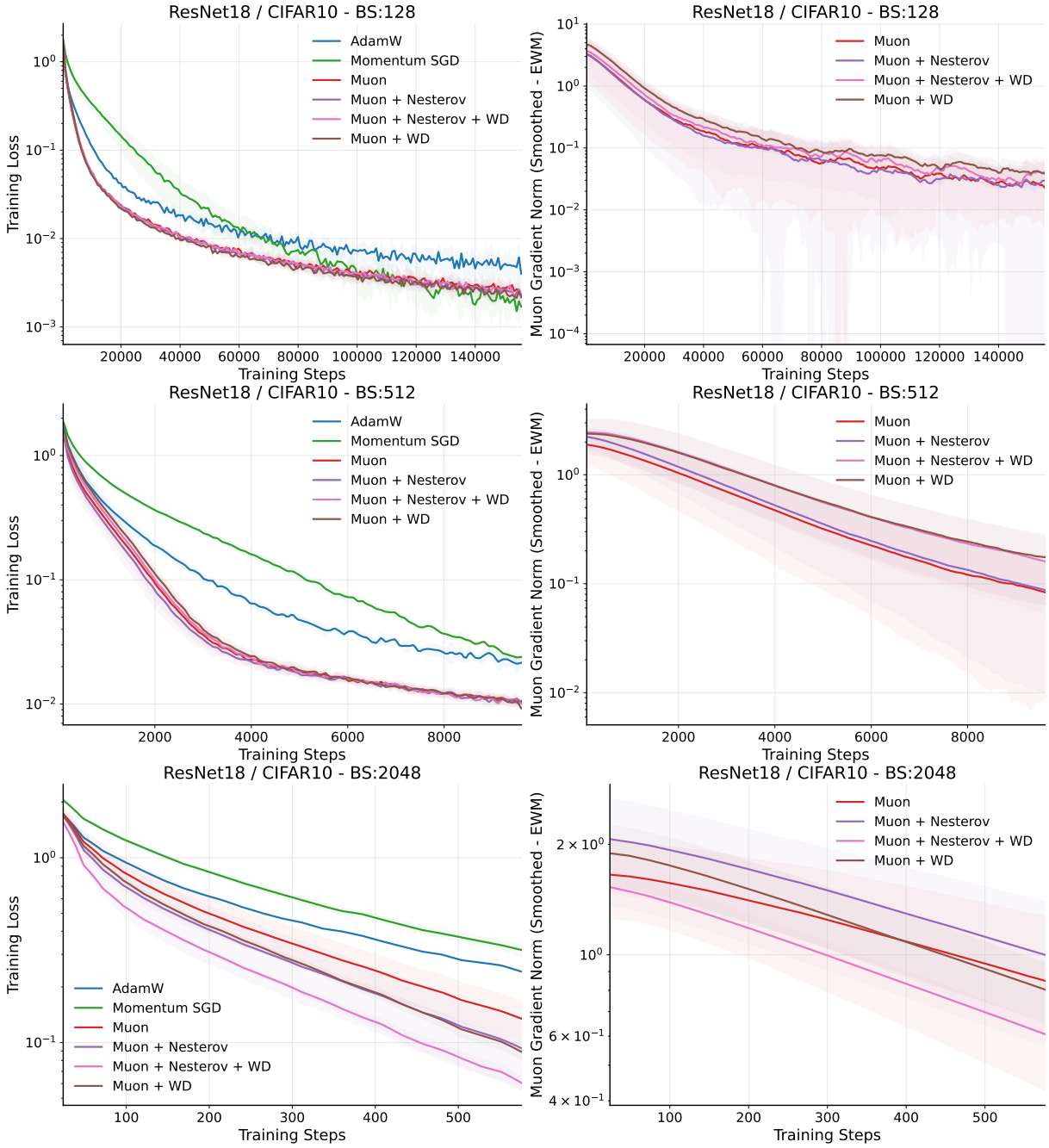

Figure 10: Convergence rate comparison for ResNet-18 on CIFAR-10 across three batch sizes (128, 512, and 2048 from top to bottom). Each row compares training loss (left) and smoothed gradient norm (right) for the Muon variants and baselines. These results supplement Figure 2 and confirm that the observed performance trends hold across a range of batch sizes.

In the main text, Figure 3 presents results exclusively for AdamW and Muon. For a broader comparison, results for Momentum SGD are included and shown in Figure 11.

Additional Vision Workload Due to space limits, the main text focuses on ResNet-18/CIFAR-10. Figure 12 reports VGG-16/CIFAR-100, which shows the same qualitative behavior.

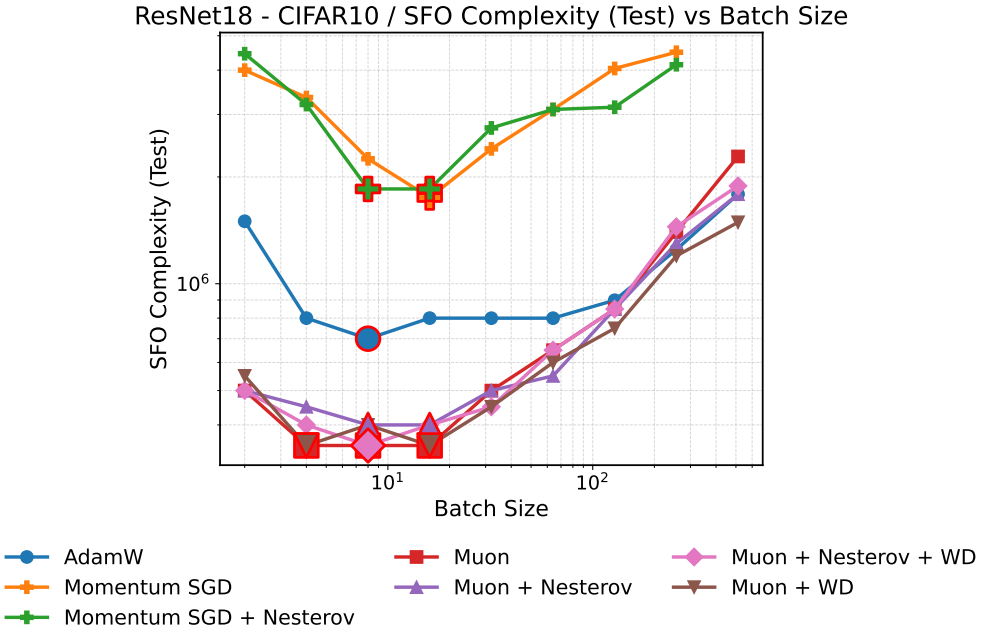

Figure 11: Optimizer comparison via analysis of batch size scaling and SFO complexity (test) for ResNet-18 on CIFAR-10.

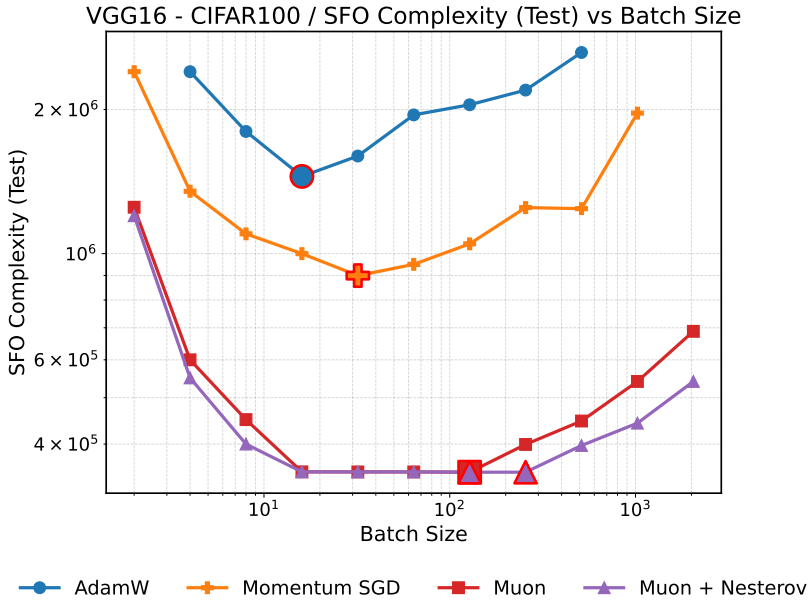

Figure 12: Optimizer comparison via analysis of batch size scaling and SFO complexity (test) for VGG-16 on CIFAR-100.

Ablation Study Finally, we conducted an ablation study to examine the relationship between the weight decay parameter β and the learning rate across four batch sizes. Figure 13 illustrates how Muon's weight decay and learning rate affect the loss for each batch size. For the training loss (top row), lower weight decay consistently corresponds to reduced loss. Similarly, the smallest learning rates within the explored range are preferred. For the test loss (bottom row), smaller learning rates yield better results. However, a clear inflection point emerges in weight decay at around 10^{-1}, or approximately 10^{-2} for larger batch sizes, indicating that these weight decay settings minimize the test loss.

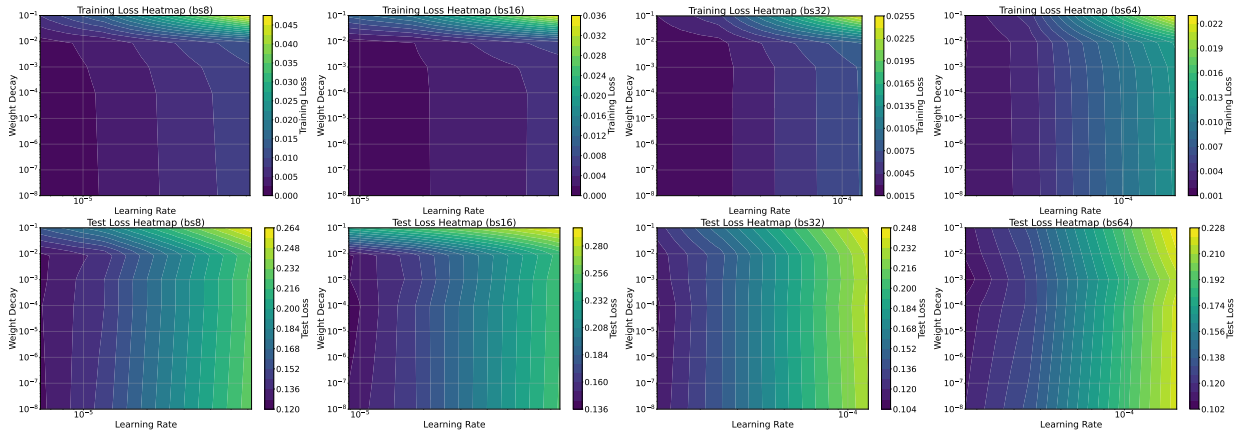

Figure 13: Ablation study on the impact of weight decay and learning rate on loss for Muon across different batch sizes (8, 16, 32, and 64, from left to right). Each column compares training loss (top) and test loss (bottom) across various configurations of weight decay and learning rate. The training loss consistently favors smaller weight decay values and the lowest learning rates within the explored range. For test loss, optimal performance is achieved at weight decay values around 10^{-1}, shifting toward approximately 10^{-2} at larger batch sizes.

G Additional Results (Language Modeling)

We provide supplemental plots for the LLM experiments discussed in the main text. Figure 14 shows final loss and SFO complexity versus batch size. Muon is consistently better than AdamW and the gap widens at large batch sizes. Nesterov momentum and weight decay do not produce systematic gains in this setting.

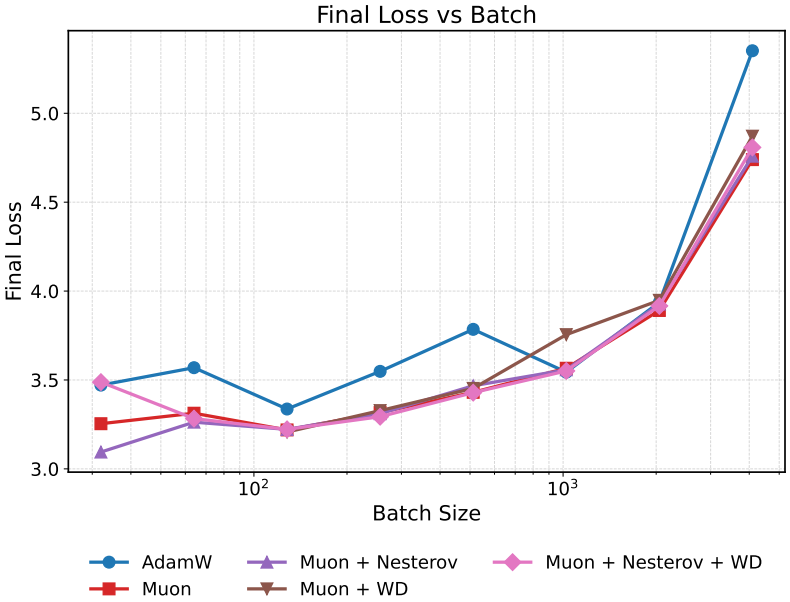

Figure 14: C4/ Llama3.1 (160M). Final training loss versus batch size. Muon outperforms AdamW across all batch sizes, with a larger margin at bigger batches.

To examine the role of momentum, we swept β under two configurations: with and without Nesterov (weight decay fixed at zero unless indicated). Figure 15 shows that the best trade-off is near $\beta = 0.95$. As β increases, the critical batch size decreases, but very small or very large values of β degrade both loss and SFO.

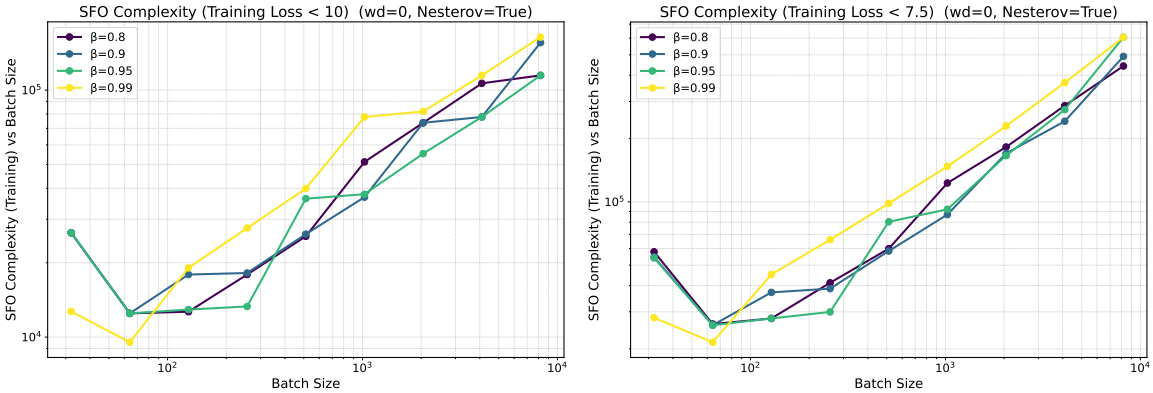

Figure 15: Effect of momentum β on C4/ Llama3.1 (160M). SFO for target loss of 10.0 (left) and SFO for target loss of 7.5 (right) across batch sizes under (WD=0, Nesterov=True). The optimum is near $\beta = 0.95$; excessive or too small momentum harms both metrics.

Training Curves for Muon on Llama3.1 (160M) To further illustrate Muon's optimization dynamics on large-language-model workloads, we report full training curves for Llama3.1 (160M) trained on C4 at multiple batch sizes. Figure 16 plots the training loss against the number of steps for Muon and AdamW. Across all batch

sizes, Muon exhibits faster loss reduction in early training and consistently reaches lower final loss within the same token budget. These results complement Figures 5 and 6, demonstrating that the advantages of Muon are visible not only in SFO complexity but also in the raw optimization trajectory.

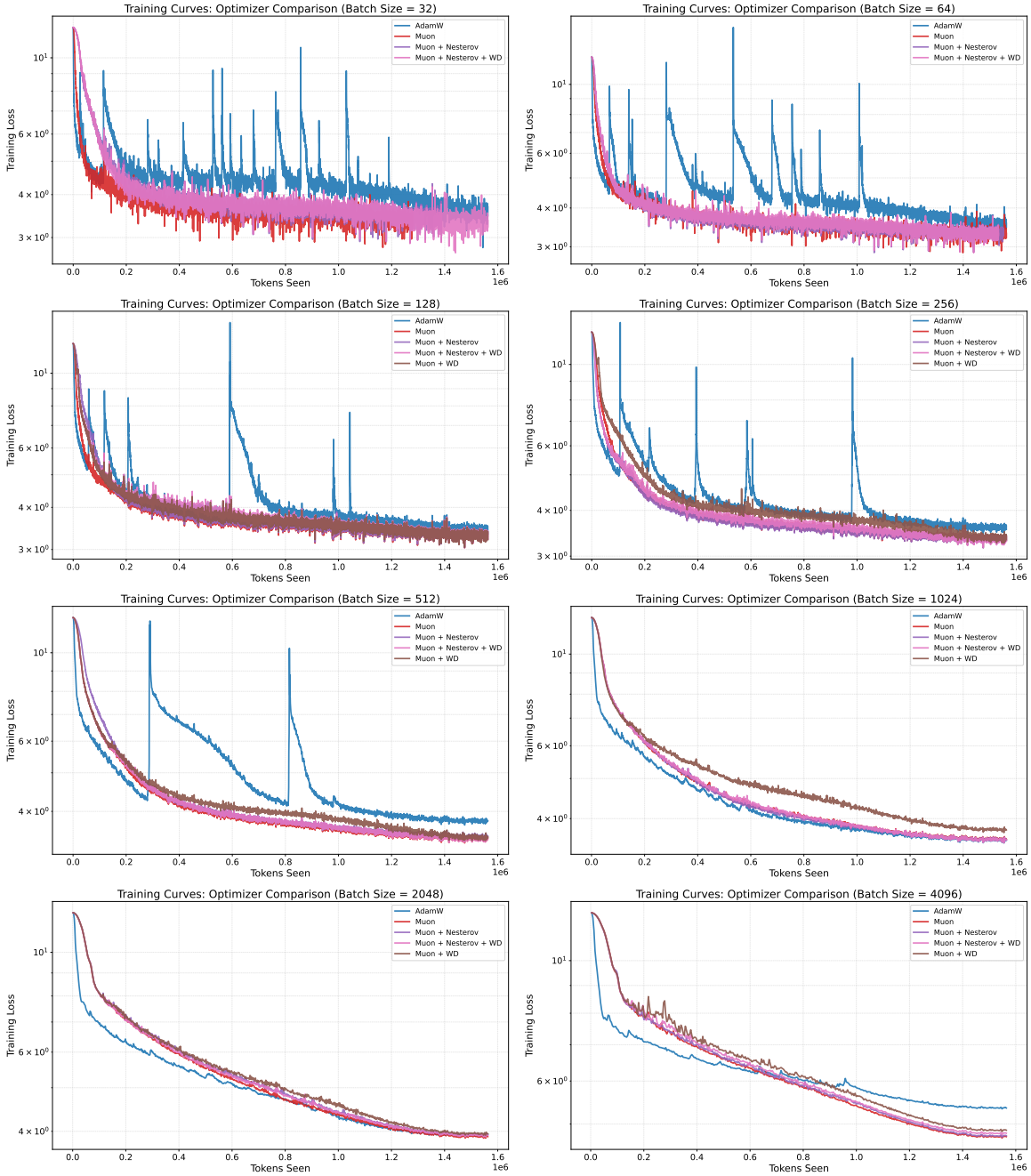

Figure 16: Training loss curves for Muon vs. AdamW on Llama3.1 (160M)/ C4. Each plot corresponds to a different global batch size (32 to 4096, top-left to bottom-right). Muon variants consistently achieve faster loss reduction and lower final loss across all batch sizes.

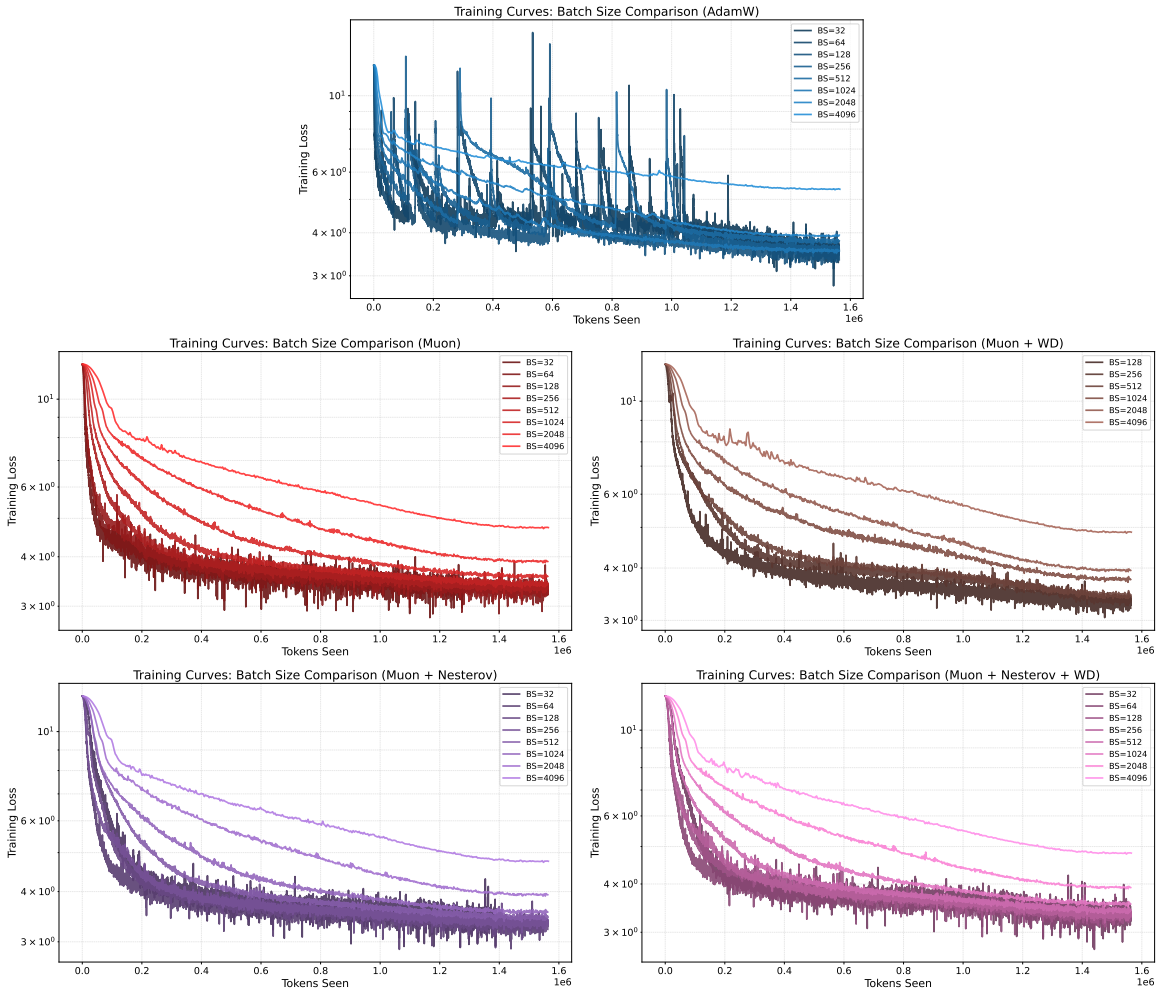

Figure 17: Training loss curves grouped by optimizer on Llama3.1 (160M)/ C4. Each plot corresponds to AdamW, Muon, Muon+WD, Muon+Nesterov, Muon+Nesterov+WD (top to bottom). Smaller batch sizes consistently achieve lower final loss across all optimizers.