

MUON²: Boosting MUON via Adaptive Second-Moment Preconditioning

Ziyue Liu¹, Ruijie Zhang¹, Zhengyang Wang¹, Yequan Zhao¹, Yupeng Su¹,
Zi Yang², Zheng Zhang¹

¹University of California at Santa Barbara; ²University at Albany, SUNY
{ziyueliu, zzhang01}@ucsb.edu

Abstract

MUON has emerged as a promising optimizer for large-scale foundation model pre-training by exploiting the matrix structure of neural network updates through iterative orthogonalization. However, the orthogonalization quality of MUON hinges on the number of Newton–Schulz (NS) iterations performed, which poses efficiency challenges due to its non-trivial computation and communication cost. We propose MUON², an extension of MUON, to improve both quality and efficiency by applying Adam-style adaptive second-moment preconditioning before orthogonalization. Our key insight is that the core challenge of polar approximation in MUON lies in the ill-conditioned momentum matrix, of which the spectrum is substantially improved by MUON², leading to faster convergence toward a practically sufficient orthogonalization. We further characterize the practical orthogonalization quality via directional alignment, under which MUON² demonstrates dramatic improvement over MUON at each polar step. Across GPT, LLaMA, and Mixture-of-Experts pre-training experiments up to 13B parameters, MUON² (and its memory-efficient variant MUON²-F that preserves most of its benefits) consistently outperforms MUON and its variants while reducing NS iterations by 40%, and saves up to 1/4 training time over MUON when achieving the same loss.

1 Introduction

The rapid progress of modern large-scale neural networks has been driven by the continual expansion of model capacity and training data (Hoffmann et al., 2022; Kaplan et al., 2020). This paradigm has enabled the emergence of highly capable foundation models across language, vision, and multi-modal domains (Achiam et al., 2023; Grattafiori et al., 2024; Team et al., 2023). However, the increasing scale of these systems has made pre-training extremely resource-intensive, requiring

vast computational budgets and long training durations. As a result, numerous efforts have been devoted to improve the efficiency of the pre-training system, spanning across model architectures (Adler et al., 2024; Liu et al., 2025b; Zhang et al., 2025), infrastructures (Shoeybi et al., 2019; Rajbhandari et al., 2020; Wang et al., 2025), and optimization algorithms (Gupta et al., 2018; Vyas et al., 2024; Jordan et al., 2024).

Among existing optimization methods, adaptive first-order optimizers such as Adam (Kingma and Ba, 2014) and AdamW (Loshchilov and Hutter, 2017) remain the de facto choice for training large models due to their robustness and ease of use. Nevertheless, their overlook of the underlying matrix structure of neural network parameters, have motivated substantial research into alternative optimization strategies (Martens and Grosse, 2015; Gupta et al., 2018; Vyas et al., 2024).

MUON (Jordan et al., 2024) has emerged as a breakthrough that explicitly exploits the matrix structure of neural network gradients, without the full cost of computing second-order statistics. MUON approximates a polar decomposition of the momentum via the Newton–Schulz iteration to efficiently orthogonalize the update to mitigate gradient rank collapse and improve optimization dynamics in large models. Various studies have shown improved stability and overall performance by deploying MUON in large-scale foundation model pre-training (Liu et al., 2025a; Shah et al., 2025; Zeng et al., 2025; Team et al., 2025).

Building on these successes, MUON remains an active area of research, with a number of recent works proposing variants that improve different aspects of the optimizer (Khaled et al., 2025; Li et al., 2025; Si et al., 2025; Amsel et al., 2026; Boissin et al., 2025; Ahn et al., 2025; Zhang et al., 2026). Most of these approaches explore modifications to MUON’s update rules, yet few has devoted to tackle the core challenge: a non-trivial amount

of Newton–Schulz (NS) iterations per update that determines the orthogonalization quality and introduces computation and communication overhead. This raises a natural question: can we simultaneously enhance the quality and reduce the burden of MUON’s orthogonalization procedure?

Contributions In this work, we investigate this question and introduce MUON², a simple yet effective modification of MUON that leverages adaptive second-moment scaling as an effective preconditioner for MUON’s orthogonalization step. Our key observation is that applying Adam-style per-parameter scaling prior to the orthogonalization significantly improves the spectral properties of the momentum matrix. Empirically, this produces a more favorable singular value distribution that simultaneously improves the convergence of Newton–Schulz iterations and the final model performance. These improvements lead to an optimizer that is both computationally lighter and empirically stronger. We summarize our contributions:

- We propose MUON², a novel generalization of MUON optimizer that preconditions the momentum matrix via Adam-style adaptive scaling prior to MUON’s orthogonalization. This simple yet effective approach simultaneously boosts model performance and training efficiency.
- To justify MUON², we identify the challenge of polar approximation lies in its input matrix’s ill-conditioned spectrum and demonstrate that MUON² significantly improves the input matrix conditioning, achieving superior directional alignment to the true orthogonalized update and substantially reducing polar iterations.
- We also propose MUON²-F, a memory-efficient version of MUON² with a factorized second-moment preconditioner. This variant effectively reduces the memory overhead of saving the full second-moment while preserving most of MUON²’s performance gain.
- We conduct comprehensive experiments on pre-training GPT, LLaMA and Mixture-of-Experts models for up to 13B parameters. Experiments show that MUON² family consistently outperforms MUON and its variants with 40% fewer Newton–Schulz iterations, and reduce 1/4 training time for achieving the same loss.

2 Related Work

Coordinate-wise Adaptive Methods. Despite ig-

noring underlying matrix structures, adaptive first-order optimizers remain the dominant choice for large-scale training. Adagrad (Duchi et al., 2011) introduced per-parameter adaptive scaling based on historical gradients, while Adam(W) (Kingma and Ba, 2014; Loshchilov and Hutter, 2017) further incorporate exponential moving averages of first and second moments. To reduce memory overhead, Adafactor (Shazeer and Stern, 2018) factorizes second-moment statistics, while more recent variants such as Adam-mini (Zhang et al., 2024) and GaLore (Zhao et al., 2024) simplify or compress optimizer states while retaining performance.

Matrix-Structured Methods. An alternative line of work leverages matrix structure for improved conditioning. Shampoo (Gupta et al., 2018) applies Kronecker-factored second-order preconditioning, while SOAP (Vyas et al., 2024) stabilizes it with adaptive scaling. MUON (Li et al., 2025) operates directly on matrix-valued momentum, approximating its polar factor via iterative orthogonalization.

Variants of MUON. Recent works extend MUON along multiple directions. PolarExpress (Amsel et al., 2026) and Turbo-Muon (Boissin et al., 2025) improve the convergence of polar approximation, while NorMuon (Li et al., 2025) and AdaMuon (Si et al., 2025) incorporate second-moment statistics into the update rule. Dion (Ahn et al., 2025) explores low-rank orthogonalization for scalability in distributed settings. These methods improve either the efficiency or effectiveness of MUON, rather than jointly addressing both.

3 The MUON² Optimizer

3.1 Introducing MUON²

We introduce MUON², a novel generalization of MUON that integrates second-moment preconditioning prior to the orthogonalization step. The algorithm is summarized in Algorithm 1.

Given a parameter matrix $\mathbf{W}_t \in \mathbb{R}^{n \times m}$ and gradient

$$\mathbf{G}_t = \nabla_{\mathbf{W}_t} \mathcal{L}(\mathbf{W}_t), \quad (1)$$

MUON first constructs a momentum estimate

$$\mathbf{M}_t = \beta_1 \mathbf{M}_{t-1} + (1 - \beta_1) \mathbf{G}_t. \quad (2)$$

MUON² augments this step with a second-moment accumulator

$$\mathbf{V}_t = \beta_2 \mathbf{V}_{t-1} + (1 - \beta_2)(\mathbf{G}_t \odot \mathbf{G}_t), \quad (3)$$

Algorithm 1 The MUON² Optimizer

Require: 2D weights $\mathbf{W}_t \in \mathbb{R}^{n \times m}$, objective $\mathcal{L}$, learning rate η , momentum coefficients β_1, β_2 , Newton–Schulz steps K , numerical constant ϵ

Ensure: Updated weights $\mathbf{W}_{t+1}$

```

1:  $\mathbf{M}_0 \leftarrow \mathbf{0}, \mathbf{V}_0 \leftarrow \mathbf{0}$ 
2: for  $t \leftarrow 1, 2, \dots$  do
3:    $\mathbf{G}_t \leftarrow \nabla_{\mathbf{W}_t} \mathcal{L}(\mathbf{W}_t)$ 
4:    $\mathbf{M}_t \leftarrow \beta_1 \mathbf{M}_{t-1} + (1 - \beta_1) \mathbf{G}_t$ 
5:    $\mathbf{V}_t \leftarrow \beta_2 \mathbf{V}_{t-1} + (1 - \beta_2)(\mathbf{G}_t \odot \mathbf{G}_t)$ 
6:    $\widetilde{\mathbf{M}}_t \leftarrow \mathbf{M}_t \odot (\sqrt{\mathbf{V}_t} + \epsilon \mathbf{1})$ 
7:    $\mathbf{O}_t \leftarrow \text{Newton-Schulz}(\widetilde{\mathbf{M}}_t, K)$ 
8:    $\mathbf{W}_{t+1} \leftarrow \mathbf{W}_t - \eta \sqrt{m/n} \mathbf{O}_t$ 
9: end for
```

which produces a preconditioned momentum

$$\widetilde{\mathbf{M}}_t = \mathbf{M}_t \odot (\sqrt{\mathbf{V}_t} + \epsilon \mathbf{1}), \quad (4)$$

where $\odot$ and $\oslash$ denote element-wise multiplication and division, respectively.

The preconditioned matrix is then orthogonalized using K steps of the Newton–Schulz (NS) iteration

$$\mathbf{O}_t = \text{NewtonSchulz}(\widetilde{\mathbf{M}}_t, K). \quad (5)$$

Finally the parameter update is

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \eta \sqrt{\frac{m}{n}} \mathbf{O}_t. \quad (6)$$

where the $\sqrt{\frac{m}{n}}$ learning rate factor is proposed by (Bernstein, 2025) for better scalability.

Compared with MUON, the only modification introduced by MUON² is the second-moment scaling prior to orthogonalization, as shown in Eq. (4). As we will show in the following sections, this simple yet effective modification substantially improves the spectral properties of the matrix entering the NS iteration, enabling better and faster convergence of the orthogonalization procedure.

3.2 Revisiting Polar Approximation in MUON

A central component of MUON is the use of the Newton–Schulz iteration to approximate the polar factor of a matrix.¹ Existing theoretical analyses of polar methods (Amsel et al., 2026; Chen and Chow, 2014; Grishina et al., 2025) typically measure approximation quality through *exact orthogonality*,

¹Strictly speaking, this map is an *orthonormalization*; we follow common usage and write “orthogonalization”.

for example via quantities such as

$$\|\mathbf{Q}^\top \mathbf{Q} - \mathbf{I}\|, \quad (7)$$

which quantifies how close the approximate factor $\mathbf{Q}$ is to an orthogonal matrix².

However, we argue that despite of its unwavering mathematical correctness, this notion of quality does not fully characterize the role polar approximation plays in MUON-family optimizers. In particular, the original MUON work (Jordan et al., 2024) explicitly uses an *inexact* orthogonalization that roughly maps singular values to $[1 - \epsilon, 1 + \epsilon]$. And surprisingly, ϵ can be as large as ~ 0.3 without harming the performance of MUON.

Let’s pivot to another example showcasing why Eq. (7) may fail as a practically effective metric. Consider a scenario where all singular values are projected to an exact constant $c \in (0, 1]$. Under this construction, the resulting matrix can be written as

$$\mathbf{Q} = c \mathbf{U} \mathbf{V}^\top, \quad (8)$$

where $\mathbf{U} \mathbf{V}^\top$ is the exact polar factor from singular value decomposition (SVD)

$$\mathbf{G} = \mathbf{U} \Sigma \mathbf{V}^\top, \quad \mathbf{Q}_* = \mathbf{U} \mathbf{V}^\top. \quad (9)$$

And Eq. (7) becomes

$$\|\mathbf{Q}^\top \mathbf{Q} - \mathbf{I}\| = |c^2 - 1| \cdot \|\mathbf{I}\|. \quad (10)$$

Therefore, the orthogonality error depends entirely on the deviation of c from 1. In particular, even if the matrix preserves the exact singular directions, any global scaling $c \neq 1$ leads to a large orthogonality error despite $\mathbf{Q}$ being perfectly aligned with the true polar factor.

We remark that for optimization, the approximate orthogonalized matrix is not used as a standalone object but rather as the *update direction*, where

$$\Delta \mathbf{W} = -\eta \mathbf{Q}, \quad (11)$$

and any scaling factor c that $\mathbf{Q}$ may possess will be absorbed into the step size η and being tuned as a hyper-parameter in practice. Therefore, a scale dependent metric such as Eq. (7) does not fully capture what orthogonalization achieves in practice and could be misleading in certain cases.

²We assume matrices have no fewer rows than columns; MUON’s implementation transposes wide matrices before orthogonalization anyway.

3.3 Directional Alignment

In contrast, if we were to measure directional alignment instead of exact orthogonality, cosine similarity becomes a strong candidate as it cancels the scaling effect on each singular value, i.e.,

$$\frac{\langle \mathbf{Q}, \mathbf{Q}_\star \rangle_F}{\|\mathbf{Q}\|_F \|\mathbf{Q}_\star\|_F} = \frac{c \|\mathbf{Q}_\star\|_F^2}{|c| \|\mathbf{Q}_\star\|_F^2} = 1. \quad (12)$$

Thus, cosine similarity³ is invariant to the global scaling factor c and reflects the fact that the update direction is unchanged.

More generally, within the Newton–Schulz (NS) iteration that MUON (Jordan et al., 2024) applies, running one step yields

$$\begin{aligned} \mathbf{Q}_{\text{NS}}^{(1)} &= a\mathbf{G} + b(\mathbf{G}\mathbf{G}^T)\mathbf{G} + c(\mathbf{G}\mathbf{G}^T)^2\mathbf{G} \\ &= \mathbf{U}(a\mathbf{\Sigma} + b\mathbf{\Sigma}^3 + c\mathbf{\Sigma}^5)\mathbf{V}^T \\ &= \mathbf{U} \text{diag}(\phi(\sigma_1), \dots, \phi(\sigma_n)) \mathbf{V}^T \end{aligned} \quad (13)$$

where $\mathbf{G} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is the SVD of the momentum matrix, $\phi(x) = ax + bx^3 + cx^5$ with coefficients $(a, b, c) = (3.4445, -4.7750, 2.0315)$ transforms each singular value σ_i . MUON repeats Eq. (13) by five times, yielding

$$\mathbf{Q}_{\text{NS}}^{(5)} = \mathbf{U} \text{diag}(\phi^5(\sigma_1), \dots, \phi^5(\sigma_n)) \mathbf{V}^T \quad (14)$$

In this case, the cosine similarity between the NS output and the true polar factor becomes

$$\cos(\mathbf{Q}_{\text{NS}^{(5)}}, \mathbf{Q}_\star) = \frac{\sum_i \phi^5(\sigma_i)}{\sqrt{n} \sqrt{\sum_i \phi^5(\sigma_i)^2}}, \quad (15)$$

which depends only on the relative singular value distribution of the NS output matrix and is invariant to scale-dependent statistics.

We remark that the cosine similarity we promote [Eq. (15)] does not necessarily contradict with the exact orthogonalization error [Eq. (7)] in practice, but cosine similarity does have superior robustness, interpretability and their practical applicability. We refer detailed discussions to Appendix B.

3.4 Spectral Effects of MUON²

We now investigate why MUON² can jointly improve the efficiency and effectiveness of MUON. We center around the Newton–Schulz (NS) iteration for our analysis as it is the only major complexity that MUON introduces over SGD/Adam-family

³The matrix inner product is $\langle \mathbf{A}, \mathbf{B} \rangle_F = \text{Tr}(\mathbf{A}^T \mathbf{B})$.

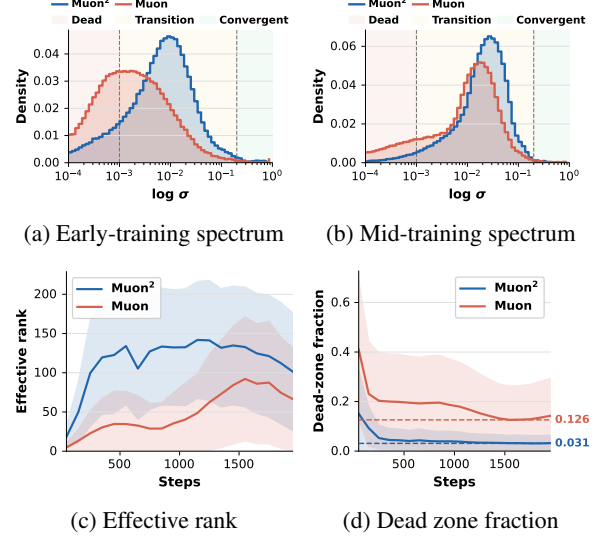


Figure 1: Spectral effect of MUON² on the input matrix of the Newton–Schulz iteration.

optimizers. As shown by Eq. (13), each NS step introduces multiple matrix multiplications per parameter that require accessing the full-matrix on each device. This introduces not just computational overhead, but also communicational overhead that’s often non-trivial and latency-bound in large-scale distributed setting. Therefore, it is essential to reduce the necessary NS steps for developing efficient MUON, yet the convergence of NS iteration largely depends on the steps being performed. To understand how MUON² breaks free from this limitation, we analyze it in two-fold, focusing on how MUON² changes: (1) The momentum matrix prior to the NS iteration. (2) The output of each NS iteration step.

For consistency, all quantitative studies in this section are conducted on training data collected from a LLaMA-60M model with MUON and MUON² using the polar approximation defined in (Jordan et al., 2024). We argue that the claims and observations we make also generalize to other polar methods such as PolarExpress (Amsel et al., 2026), with minor differences in certain numerical values, see full details in Appendix C.

3.4.1 MUON² Improves NS Input Spectrum

We start with characterizing the singular value distribution of the input momentum matrix of NS, that is Eq. (2) for MUON and Eq. (4) for MUON². We normalize it by its Frobenius norm (Jordan et al., 2024) to reflect the actual input of the NS iteration. Given an ϵ , which defines the practical orthogonalization target of mapping any normalized singular values σ_i to $[1 - \epsilon, 1 + \epsilon]$, and given a practical

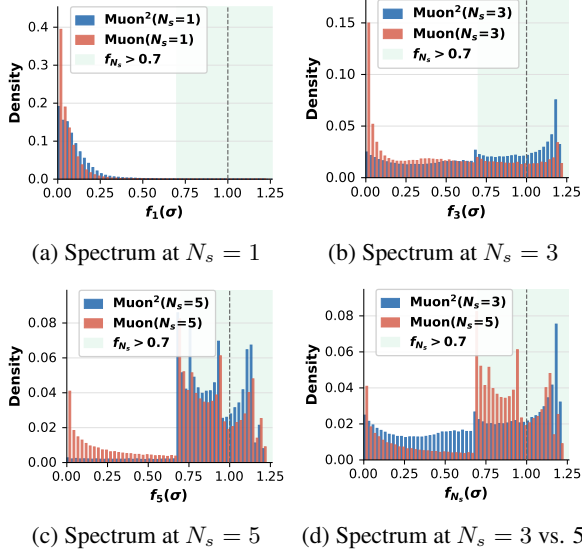


Figure 2: Spectral effect of MUON² on the Newton-Schultz (NS) output (i.e., $f_{N_s}(\sigma)$) at each step (i.e., N_s).

choice of Newton-Schultz steps N_s , we define the following convergence zones for a polar approximation based on its destination values $\phi_\epsilon^{N_s}(\sigma_i)$:

- *Dead zone*: $\phi_\epsilon^{N_s}(\sigma_i) < 1 - \epsilon$, reflects the range of singular values that fail to converge after N_s steps. Larger the dead zone is, more deviated the results are from the true orthogonalized update.
- *Transition zone*: $\phi_\epsilon^{N_s}(\sigma_i) \geq 1 - \epsilon$ and $\phi_\epsilon^1(\sigma_i) < 1 - \epsilon$. This is the region where non-trivial amount of N_s steps are needed to achieve the practical orthogonalization target.
- *Convergent zone*: $\phi_\epsilon^1(\sigma_i) \geq 1 - \epsilon$, where σ_i is large enough that only one NS iteration is needed.

Concretely, under MUON’s setting where $\epsilon \approx 0.3$, $N_s = 5$, we can calculate each zone roughly as $[0, 0.001]$, $[0.001, 0.2]$, $[0.2, 1]$, and is colored differently in Figures 1a and 1b.

The biggest practical challenge for NS is to project a wide range of singular values that spans multiple orders of magnitude to close one as fast as possible. As shown in Figure 1a, MUON’s early-training stage singular value distribution for NS input spans from 10^{-4} to 1, and is centered around 10^{-3} , where almost half singular values fall into the dead zone. Comparatively, MUON² shows a significantly right-shifted distribution, where it centers around 10^{-2} , $10\times$ larger than MUON, and the majority falls into the transition zone. As the training continues, both methods have their distribution shifting right, while MUON² shows consistently lower fraction in the dead zone, demonstrated by Figures 1b and 1d. In addition to lower dead-zone

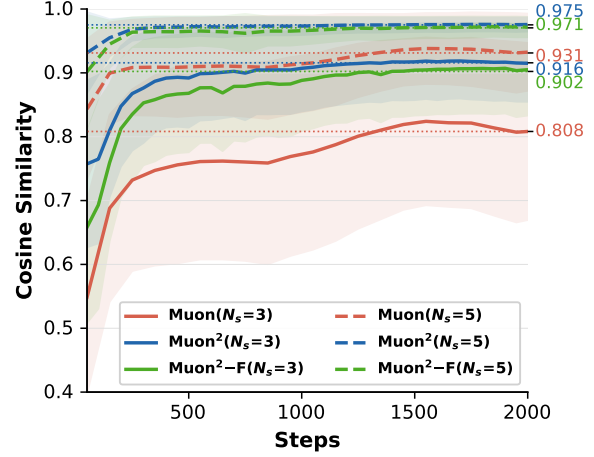


Figure 3: Cosine similarity of the output matrix vs. true orthogonalized update for MUON, MUON² and MUON²-F at different Newton-Schultz steps.

fraction, Figure 1c shows that MUON² has consistently higher effective rank throughout training, highlighting the fact that its singular values are closer together, i.e., a tighter spectrum, which coincides with the properties that cosine similarity [Eq. (15)] promotes. Therefore, we anticipate the spectrum after polar transformation will also be tighter, resulting a higher cosine similarity, thus more aligned with the true orthogonalized update.

These findings conclude our first perspective: the preconditioning effect of MUON² significantly improves the input spectrum of the NS iteration, providing a stronger baseline that is easier to achieve practically sufficient polar approximation.

3.4.2 MUON² Improves Polar Quality

Now we compare the polar approximation behavior of MUON² and MUON by visualizing their singular value distributions at NS steps (N_s) 1, 3, and 5 in Figures 2a to 2c. Across all cases, MUON² exhibits a tighter spectrum with substantially fewer near-zero singular values. By $N_s = 5$, its singular values fall almost entirely within the target range. We further compare MUON² at $N_s = 3$ against MUON at $N_s = 5$ in Figure 2d. Despite using substantially fewer iterations, MUON² produces a highly similar distribution while reducing the density of extremely small singular values by roughly half. These findings suggest that MUON² at $N_s = 3$ already provides a strong polar approximation.

Quantitatively, we calculate the cosine similarity [Eq. (15)] of MUON² and MUON at each N_s , reflecting how closely the approximate polar factors align with the true orthogonalized gradients. As shown in Figure 3, MUON² at $N_s = 3$ already

achieves cosine similarity comparable to MUON at $N_s = 5$ (0.916 vs. 0.931), and further improves to 0.975 at $N_s = 5$, substantially surpassing MUON. In contrast, reducing MUON from $N_s = 5$ to $N_s = 3$ causes a significant drop in cosine similarity (0.931 $\rightarrow$ 0.808), which we later show also leads to a dramatic degradation in model performance.

To better connect the qualitative and quantitative analyses, we compare MUON² at $N_s = 3$ with MUON at $N_s = 5$ in Figure 2d. The visualization shows that MUON has monotonically decreasing densities between near-zero and the target range, whereas MUON² exhibits a more uniform spread with higher density in the transition and convergent zones. This suggests a more practically favored spectrum of MUON² that accounts for its further improvement at $N_s = 5$.

These findings conclude our second perspective: benefited from the spectral effect of the second moment preconditioning, MUON² consistently produces updates that are both better aligned with and easier to progress towards the true orthogonalized gradient at every NS iteration. This effectively reduces the necessary steps of the NS iteration and substantially improves the polar approximation quality.

3.4.3 A Random-Matrix Perspective

The two perspectives above are established empirically. In this section, we further provide a theoretical account for the spectral effect of MUON². We model the momentum matrix as $M_{ij} = s_{ij} \xi_{ij}$, where the deterministic scale $s_{ij} > 0$ captures the persistent fluctuation magnitude of each entry, and ξ_{ij} are independent zero-mean, unit-variance variables. Under this model, the spectrum of M is determined by the scale profile (s_{ij}) : when the scales are highly uneven, a large fraction of singular values can be pushed towards zero (we give a concrete construction in Appendix D), which is exactly the ill-conditioned NS input that Section 3.4.1 identifies as the practical challenge. Meanwhile, the preconditioning of MUON² [Eq. (4)] divides each entry by the second-moment estimate of precisely this scale. The following proposition characterizes the spectrum after the division.

Proposition 1. *Let $(\xi_{ij})_{i,j \geq 1}$ be a fixed infinite array of i.i.d. variables with zero mean, unit variance, and $\mathbb{E}\xi^4 < \infty$. For a sequence $m = m_n \leq n$ with $m/n \rightarrow \lambda \in (0, 1)$, let $M = (s_{ij} \xi_{ij}) \in \mathbb{R}^{m \times n}$, where the scales $s_{ij} > 0$ are deterministic, and let $\widetilde{M} := (M_{ij}/s_{ij})$ denote the exactly preconditioned*

matrix. Then the empirical singular-value distribution of $\widetilde{M}/\sqrt{n}$ converges almost surely to the singular-value form of the Marchenko–Pastur law with ratio λ (Marčenko and Pastur, 1967) — the pushforward of the Marchenko–Pastur law under $x \mapsto \sqrt{x}$, supported on $[1 - \sqrt{\lambda}, 1 + \sqrt{\lambda}]$ — independently of (s_{ij}) . Moreover, the empirical distribution of $\{\sqrt{m} \sigma_i(\widetilde{M}/\|\widetilde{M}\|_F)\}_{i=1}^m$ converges almost surely to the same law; consequently, for every $0 \leq t < 1 - \sqrt{\lambda}$, the fraction of singular values of the Frobenius-normalized NS input below $t/\sqrt{m}$ vanishes almost surely.

Proposition 1 states that once the scales are divided out, the singular values concentrate on the Marchenko–Pastur support $[1 - \sqrt{\lambda}, 1 + \sqrt{\lambda}]$, independently of (s_{ij}) . In other words, the preconditioning removes the dependence of the spectrum on the scale profile: no matter how uneven the scales are, the fraction of singular values below the lower edge $1 - \sqrt{\lambda}$ vanishes. This edge depends only on the aspect ratio of the layer: it is positive for rectangular layers such as the MLP projections, which is why the proposition assumes $\lambda < 1$, while for square layers it sits at zero and a weaker form of the same conclusion holds (see Appendix D). Consequently, the dead-zone fraction of the preconditioned input vanishes, matching the right-shifted, tighter spectrum observed in Section 3.4.1.

In practice, the momentum is not pure noise but also accumulates a signal component, which corresponds to a nonzero mean μ_{ij} in our model. The following corollary shows that the conclusion of Proposition 1 is robust to such a signal in two regimes: a low-rank signal of any bounded strength, and a weak signal of any rank.

Corollary 1. *In the setting of Proposition 1, let $M_{ij} = \mu_{ij} + s_{ij}\xi_{ij}$ with deterministic μ_{ij} , so that $\widetilde{M} = D + \Xi$ with $D := (\mu_{ij}/s_{ij})$ and $\Xi := (\xi_{ij})$. (i) If $\text{rank}(D) = o(m)$ and $\limsup_n \|D\|_F^2/(mn) \leq d^2 < \infty$ for some $d \geq 0$, then almost surely the fraction of NS-input singular values below $t/\sqrt{m}$ vanishes for every $0 \leq t < (1 - \sqrt{\lambda})/(1 + d)$. (ii) If $\|D\|_F^2 = o(mn)$, then all conclusions of Proposition 1 hold unchanged.*

In the corollary, D is the matrix of per-entry signal-to-noise ratios. Case (i) corresponds to the regime suggested by our measurements: the low effective ranks in Section 3.4.1 indicate a signal concentrated in few directions, and for such a signal the guaranteed margin only degrades from $1 - \sqrt{\lambda}$ to $(1 - \sqrt{\lambda})/(1 + d)$. Case (ii) shows that

a signal spread over many directions is tolerated as long as it is weak. We note the limitations of these results: they are asymptotic, they control the fraction of singular values in the dead zone rather than the smallest one, and they idealize the second-moment estimate as the exact scale. We quantify these gaps, together with the connection to the factorized second moment of MUON²-F, in Appendix D, where the proofs are also given.

3.5 Overall Benefits of MUON²

MUON² delivers a stronger orthogonalization. With the same number of Newton-Schultz (NS) iterations, MUON² significantly improves the alignment of MUON’s update direction with the true orthogonalized gradient.

MUON² reduces NS iterations for practically sufficient orthogonalization. To achieve a similar level of update-direction alignment, MUON² reduces the required NS iterations by 40%. We will further show in the experiments that 3-step MUON² already achieves better model performance than 5-step MUON. We speculate that this is due to the richer historical information carried by our second-moment preconditioning.

3.6 MUON²-F: A Memory Efficient MUON²

From Eq. (3) and (4), MUON² introduces additional memory overhead by maintaining a second moment throughout training. However, since the second moment only conditions the input to orthogonalization rather than directly determining the update, we expect MUON² to be relatively insensitive to approximation error. Motivated by this, we adopt Adafactor-style factorized second moments:

$$\begin{aligned} \mathbf{r}_t &\leftarrow \beta_2 \mathbf{r}_{t-1} + (1 - \beta_2) \text{sum}_{\text{col}}(\mathbf{R}_t), \\ \mathbf{c}_t &\leftarrow \beta_2 \mathbf{c}_{t-1} + (1 - \beta_2) \text{sum}_{\text{row}}(\mathbf{R}_t), \\ \hat{\mathbf{V}}_t &\leftarrow \frac{\mathbf{r}_t \mathbf{c}_t^\top}{\text{sum}(\mathbf{r}_t)}. \end{aligned} \quad (16)$$

Instead of storing the full second-moment matrix, this approximation keeps only row- and column-wise statistics and reconstructs the matrix via their outer product. We denote this variant as MUON²-Factorized (MUON²-F). As shown later, MUON²-F remains close to the exact MUON² in both reduced NS iteration requirements and model performance, while keeping memory overhead nearly unchanged from MUON. Full details are in Algorithm 2.

Model	NS Steps	MUON	MUON ²	MUON ² -F
GPT-Small	$N_s = 3$	32.70	28.12 (-4.58)	28.30 (-4.40)
	$N_s = 5$	29.51	27.95 (-1.56)	27.93 (-1.58)
GPT-Base	$N_s = 3$	24.69	20.39 (-4.30)	21.17 (-3.52)
	$N_s = 5$	21.47	19.96 (-1.51)	20.58 (-0.89)
GPT-Large	$N_s = 3$	21.13	16.99 (-4.14)	17.69 (-3.44)
	$N_s = 5$	17.56	16.52 (-1.04)	16.55 (-1.01)

Table 1: Validation perplexity ($\downarrow$) of MUON and MUON² on GPT models across Newton–Schulz iterations.

4 Experiments

In this section, we evaluate MUON² on extensive pre-training experiments that cover GPT, LLaMA and Mixture-of-Experts architectures on various scales. These experiments strongly and consistently demonstrate that MUON² achieves two benefits: (1) Significantly better model performance; (2) Substantially fewer Newton–Schulz (NS) iterations. We also compare MUON² against other MUON variants and show that none of them can achieve both benefits as MUON² does.

4.1 Pre-Training GPT

We pre-train GPT models at three scales: small, base and large, with respectively 3.0B, 7.2B, and 15.5B tokens, following the compute-optimal training regime (Hoffmann et al., 2022). We use the FineWeb dataset (Penedo et al., 2024) tokenized by the GPT tokenizer. Trainings are run on H100/A100 GPUs, with the pipeline adapted from NanoGPT (Karpathy, 2022). We use the polar method from the original MUON work (Jordan et al., 2024). Detailed configurations and hyperparameters are provided in Appendix E.1.

As shown in Table 1, MUON² consistently outperforms MUON not just when they use the same NS steps, but also when MUON² uses substantially fewer NS steps. With only $N_s = 3$, MUON² outperforms MUON with $N_s = 5$ at all three scales. When both at $N_s = 3$, the gap between MUON² and MUON is even more dramatic. Meanwhile, the memory efficient version MUON²-F also achieves comparatively good performance with practically negligible memory cost. To ensure fair comparison, we sweep learning rates for MUON and MUON² and show results at GPT-Large scale in Figure 4. We can observe that the benefits MUON² provides are irrelevant to each individual choice of learning rate, suggesting its broad practical applicability.

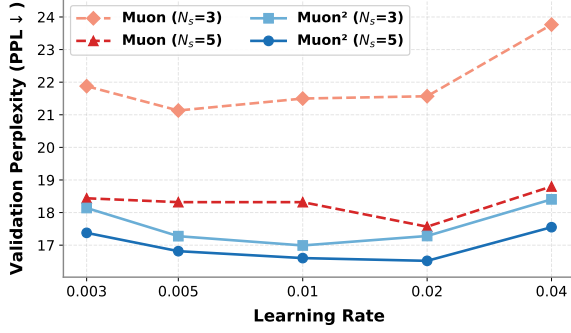


Figure 4: Learning rate sweep on GPT-Large comparing MUON² and MUON across Newton-Schulz iterations.

Model	NS Steps	MUON	MUON ²	MUON ² -F
LLaMA-60M	$N_s = 3$	26.37	24.59 (-1.78)	24.68 (-1.69)
	$N_s = 5$	24.98	24.60 (-0.38)	24.66 (-0.32)
LLaMA-350M	$N_s = 3$	14.91	13.44 (-1.47)	13.55 (-1.36)
	$N_s = 5$	14.03	13.46 (-0.57)	13.44 (-0.59)
LLaMA-1B	$N_s = 3$	11.63	10.42 (-1.21)	10.49 (-1.14)
	$N_s = 5$	10.62	10.21 (-0.41)	10.21 (-0.41)

Table 2: Validation perplexity ($\downarrow$) of MUON and MUON² on LLaMA models across Newton-Schulz iterations.

4.2 Pre-Training LLaMA

We pre-train LLaMA-style models at four scales: 60M, 350M, 1B and 13B, with respectively 1.0B, 7.3B, 20.1B, and 26.8B. We use C4 (Raffel et al., 2020) dataset tokenized by LLaMA-2 tokenizer. We adopt the training pipeline from Nanotron (Tazi et al., 2025). Detailed configurations and hyper-parameters are provided in Appendix E.2.

As shown in Tables 2 and 3, the results are consistent with GPT models. MUON² continues to outperform MUON even with fewer NS iterations. At all four scales, reducing NS steps will cause a dramatic degradation for MUON but only minimum drops for MUON², and the gap between MUON² and MUON²-F is also less pronounced.

4.3 Pre-Training Mixture-of-Experts

We further evaluate on Mixture-of-Experts (MoE), a family of model architectures that is widely adopted in industry (Achiam et al., 2023; Zeng et al., 2025; Liu et al., 2024). We pre-train a 7B-A1B MoE (1B active parameters, 2 out of total 16 experts) for 20.1B tokens on 4 variants: MUON and MUON² each with $N_s = 3$ and $N_s = 5$ (Table 3). MUON² continues to outperform MUON significantly on both NS settings, and reduces NS steps by 40% while improving model performance.

Model	NS Steps	MUON	MUON ²
LLaMA-13B	$N_s = 3$	10.54	9.11 (-1.43)
	$N_s = 5$	9.24	8.71 (-0.53)
MoE-7B-A1B	$N_s = 3$	11.93	9.85 (-2.08)
	$N_s = 5$	10.61	9.66 (-0.95)

Table 3: Validation perplexity ($\downarrow$) of MUON and MUON² on LLaMA-13B and MoE models across NS iterations.

4.4 Downstream Performance Evaluation

Beyond validation perplexity, we also verify the effectiveness of MUON² by performing downstream evaluations on the trained LLaMA-1B in Table 4. Results are well aligned with validation perplexity that MUON² consistently outperforms MUON. Full details and more evaluations are in Appendix E.4.

4.5 Comparisons against MUON Variants

We have demonstrated that MUON² consistently outperforms MUON while reducing 40% NS iterations across various scales and architectures. Now we verify that existing MUON variants cannot achieve these benefits at the same extent.

4.5.1 Variants of Polar Method

Methods that solely modify the polar approximation are closer to MUON² in design principles, such as PolarExpress (Amsel et al., 2026) and Turbo-Muon (Boissin et al., 2025). We compare MUON² with them on GPT-Small and GPT-Base. To ensure fairness, we also sweep hyper-parameters for baselines and report their best results in Table 5. Full details are provided in Appendix E.5. By setting MUON² as the baseline in Table 5, we clearly observe that both PolarExpress and Turbo-Muon underperform MUON² even with more NS iterations. They do suffer less degradation than plain MUON with reduced NS steps, but not comparable to MUON², and their improvement (if any) over MUON at $N_s = 5$ is much less pronounced.

4.5.2 Variants of Update Rules

For other variants of MUON, we focus on ones that are closer to MUON² in form, such as NorMuon (Li et al., 2025) and AdaMuon (Si et al., 2025). Despite of both using second moment in certain capacity, these approaches are fundamentally different from ours: they adapt the step size of each update direction similar to Adam, while MUON² preserves the orthogonalized directions as in MUON.

Similarly, we compare MUON² with NorMuon and AdaMuon on GPT-Small and GPT-Base with hyper-parameter sweeping, and report their best

	NS Steps	Avg	ARC-c [†]	ARC-e	OBQA	HellaS.	PIQA	WinoG. [†]	LAMBADA	MMLU [†]
MUON	$N_s = 3$	42.56	27.08	42.80	32.00	44.37	71.22	56.75	36.52	29.75
	$N_s = 5$	45.36	30.01	46.09	33.20	49.75	72.25	59.91	40.99	30.71
MUON ²	$N_s = 3$	45.89	31.91	47.81	31.80	50.23	72.20	59.88	42.50	30.79
	$N_s = 5$	47.83	33.48	49.03	35.20	52.61	74.05	61.90	45.08	31.31

Table 4: Downstream evaluation on LLaMA-1B across NS iterations. Tasks with [†] are fine-tuned first (Appendix E.4); the rest are zero-shot.

	GPT-Small		GPT-Base	
NS Steps	$N_s = 3$	$N_s = 5$	$N_s = 3$	$N_s = 5$
MUON²	28.12	27.95	20.39	19.96
MUON (Jordan et al., 2024)	32.70 (+4.58)	29.51 (+1.56)	24.69 (+4.30)	21.47 (+1.51)
PolarExpress (Amsel et al., 2026)	30.01 (+1.89)	29.42 (+1.47)	22.74 (+2.35)	21.16 (+1.20)
Turbo-Muon (Boissin et al., 2025)	29.70 (+1.58)	29.66 (+1.71)	23.46 (+3.07)	21.93 (+1.97)
NorMuon (Li et al., 2025)	30.35 (+2.23)	28.40 (+0.45)	23.33 (+2.94)	21.27 (+1.31)
AdaMuon (Si et al., 2025)	31.20 (+3.08)	29.30 (+1.35)	26.07 (+5.68)	22.42 (+2.46)

Table 5: Validation perplexity ($\downarrow$) of MUON² comparing against MUON variants on GPT-Small and Base.

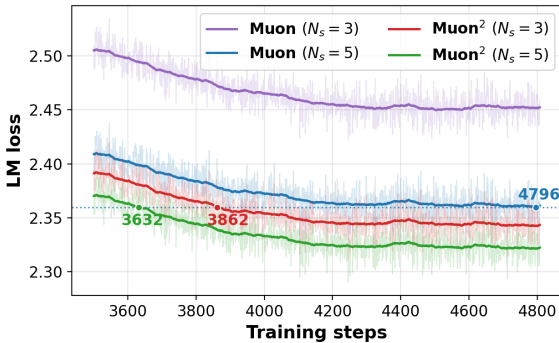


Figure 5: On LLaMA-1B, MUON² achieves same loss with 25% fewer training steps than MUON.

results in Table 5 (full results in Appendix E.5). Despite their improvement (if any) over MUON, both methods fail to reduce the necessary NS iterations, and they both underperform MUON² in all settings, even with more NS steps than MUON².

4.6 Training Speedup

We show in Figure 5 that on LLaMA-1B, MUON² achieves the same training loss as MUON with up to 25% fewer steps. Meanwhile, the second-moment scaling of MUON² is computationally minimal, so that the per-training-step time is almost unchanged from MUON. As a result, the reduced steps directly translate to 25% fewer training time (1042 vs. 796 GPU-hours) when achieving the same loss (2.36).

Model	Memory (GB)			
	AdamW	MUON	MUON ²	MUON ² -F
LLaMA-1B	20.39	18.56	20.36	18.57
LLaMA-7B	59.31	47.29	60.66	47.53

Table 6: Memory comparison across Adam, MUON, MUON² and MUON²-F.

For the same training steps, MUON² with $N_s = 3$ achieves a lower loss (2.34 vs 2.36) while reducing GPU-hours from 1044 to 1016. See Appendix E.9.

4.7 Memory Efficiency of MUON²-F

Table 6 shows the memory usage across numerous approaches. MUON²-F uses practically same memory as MUON, verifying its superior memory efficiency while preserving most of MUON²'s performance gains. See details in Appendix E.10.

5 Conclusion

We have presented MUON², a simple yet effective extension of MUON that applies adaptive second-moment preconditioning before orthogonalization, and shown that MUON² achieves two desirable outcomes simultaneously: improved optimization behavior and reduced orthogonalization cost. We have analyzed the effect of MUON² by first identifying the core challenge of the polar approximation lies in its ill-conditioned input, then showing how the proposed preconditioning fundamentally addresses this issue and results in significantly higher orthogonalization quality. As a result, MUON² achieves practically sufficient orthogonalization with substantially fewer polar iterations. We have shown through comprehensive experiments that MUON² (-F) consistently improves model performance while reducing polar iterations by 40%. When achieving the same training loss, MUON² saves up to 25% GPU-hours over MUON, significantly lowers the massive pre-training cost, pushing forward the frontier of efficient and powerful optimizers for LLM pre-training.

6 Limitation

MUON² is developed under academic budgets, following training settings in state-of-the-art literature (Li et al., 2025; Si et al., 2025; Zhang et al., 2026; Ahn et al., 2025; Amsel et al., 2026; Boissin et al., 2025), where training models on 13B parameters already exceeds the scales adopted in most of these papers. As a result, we have not yet evaluated MUON² at frontier-scale settings that are only viable under industrial budgets. In addition to MUON²-F, there may still exist other memory-efficient designs that better balance optimization quality and optimizer-state cost. We leave this as a future research.

7 Ethical Considerations

This work studies optimization methods for large-scale language model pre-training and does not introduce new datasets or involve the collection of personally identifiable information. All experiments are conducted on publicly available corpora (Penedo et al., 2024; Raffel et al., 2020) that follow their respective licenses and terms of use. As our method focuses on improving training efficiency and optimization quality, we do not foresee ethical concerns beyond those already associated with large language model pre-training. In particular, models trained with MUON² may still inherit biases, toxic content, hallucinations, or other harmful behaviors present in the underlying training data. Therefore, models trained using our optimizer should still undergo standard safety evaluation and responsible deployment practices.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, and 1 others. 2024. Nemotron-4 340b technical report. *arXiv preprint arXiv:2406.11704*.
- Kwangjun Ahn, Byron Xu, Natalie Abreu, Ying Fan, Gagik Magakyan, Pratyusha Sharma, Zheng Zhan, and John Langford. 2025. Dion: Distributed orthonormalized updates. *arXiv preprint arXiv:2504.05295*.
- Noah Amsel, David Persson, Christopher Musco, and Robert M Gower. 2026. The polar express: Optimal matrix sign methods and their application to the muon algorithm. In *International Conference on Learning Representations*, volume 2026, pages 138323–138360.
- Zhidong Bai and Jack W Silverstein. 2010. *Spectral analysis of large dimensional random matrices*, volume 20. Springer.
- Jeremy Bernstein. 2025. [Deriving muon](#).
- Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, and 1 others. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.
- Thibaut Boissin, Thomas Massena, Franck Mamalet, and Mathieu Serrurier. 2025. Turbo-muon: Accelerating orthogonality-based optimization with preconditioning. *arXiv preprint arXiv:2512.04632*.
- Jie Chen and Edmond Chow. 2014. A stable scaling of newton-schulz for improving the sign function computation of a hermitian matrix. *Preprint*. *ANL/MCS-P5059-0114*.
- Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and 1 others. 2023. Symbolic discovery of optimization algorithms. *Advances in neural information processing systems*, 36:49205–49233.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.
- Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2021. 8-bit optimizers via block-wise quantization. *arXiv preprint arXiv:2110.02861*.
- John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).
- Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, and 1 others. 2021. A framework for few-shot language model evaluation. *Zenodo*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Ekaterina Grishina, Matvey Smirnov, and Maxim Rakhuba. 2025. Accelerating newton-schulz iteration for orthogonalization via chebyshev-type polynomials. *arXiv preprint arXiv:2506.10935*.

- Vineet Gupta, Tomer Koren, and Yoram Singer. 2018. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pages 1842–1850. PMLR.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, and 1 others. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.
- Roger A Horn and Charles R Johnson. 2012. *Matrix analysis*. Cambridge university press.
- Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. 2024. [Muon: An optimizer for hidden layers in neural networks](#).
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.
- Andrej Karpathy. 2022. [nanogpt](#).
- Ahmed Khaled, Kaan Ozkara, Tao Yu, Mingyi Hong, and Youngsuk Park. 2025. Muonbp: Faster muon via block-periodic orthogonalization. *arXiv preprint arXiv:2510.16981*.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Zichong Li, Liming Liu, Chen Liang, Weizhu Chen, and Tuo Zhao. 2025. Normuon: Making muon more efficient and scalable. *arXiv preprint arXiv:2510.05491*.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.
- Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, and 1 others. 2025a. Muon is scalable for llm training. *arXiv preprint arXiv:2502.16982*.
- Ziyue Liu, Ruijie Zhang, Zhengyang Wang, Mingsong Yan, Zi Yang, Paul D. Hovland, Bogdan Nicolae, Franck Cappello, Sui Tang, and Zheng Zhang. 2025b. [CoLA: Compute-efficient pre-training of LLMs via low-rank activation](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4627–4645, Suzhou, China. Association for Computational Linguistics.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Vladimir A Marčenko and Leonid Andreevich Pastur. 1967. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457–483.
- James Martens and Roger Grosse. 2015. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pages 2408–2417. PMLR.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2381–2391.
- Leon Mirsky. 1960. Symmetric gauge functions and unitarily invariant norms. *The quarterly journal of mathematics*, 11(1):50–59.
- Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc-Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The lambda dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 1525–1534.
- Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, and 1 others. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.
- Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: international conference for high performance computing, networking, storage and analysis*, pages 1–16. IEEE.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.
- Ishaan Shah, Anthony M Polloreno, Karl Stratos, Philip Monk, Adarsh Chaluvvaraju, Andrew Hojel, Andrew Ma, Anil Thomas, Ashish Tanwer, Darsh J Shah, and 1 others. 2025. Practical efficiency of muon for pretraining. *arXiv preprint arXiv:2505.02222*.

- Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International conference on machine learning*, pages 4596–4604. PMLR.
- Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.
- Chongjie Si, Debing Zhang, and Wei Shen. 2025. Adamuon: Adaptive muon optimizer. *arXiv preprint arXiv:2507.11005*.
- Nouamane Tazi, Ferdinand Mom, Haojun Zhao, Phuc Nguyen, Mohamed Mekouri, Leandro Werra, and Thomas Wolf. 2025. The ultra-scale playbook: Training llms on gpu clusters. 2025. URL: <https://huggingface.co/spaces/nanotron/ultrascaleplaybook>.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, and 1 others. 2025. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*.
- Nikhil Vyas, Depen Morwani, Rosie Zhao, Mujin Kwun, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. 2024. Soap: Improving and stabilizing shampoo using adam. *arXiv preprint arXiv:2409.11321*.
- Zhengyang Wang, Ziyue Liu, Ruijie Zhang, Avinash Maurya, Paul Hovland, Bogdan Nicolae, Franck Cappello, and Zheng Zhang. 2025. Boost: Bottleneck-optimized scalable training framework for low-rank large language models. *arXiv preprint arXiv:2512.12131*.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4791–4800.
- Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, and 1 others. 2025. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*.
- Ruijie Zhang, Ziyue Liu, Zhengyang Wang, and Zheng Zhang. 2025. Lax: Boosting low-rank training of foundation models via latent crossing. *arXiv preprint arXiv:2505.21732*.
- Ruijie Zhang, Yequan Zhao, Ziyue Liu, Zhengyang Wang, Dongyang Li, Yupeng Su, Sijia Liu, and Zheng Zhang. 2026. Teon: Tensorized orthonormalization beyond layer-wise muon for large language model pre-training. *arXiv preprint arXiv:2601.23261*.
- Yushun Zhang, Congliang Chen, Ziniu Li, Tian Ding, Chenwei Wu, Diederik P Kingma, Yinyu Ye, Zhi-Quan Luo, and Ruoyu Sun. 2024. Adam-mini: Use fewer learning rates to gain more. *arXiv preprint arXiv:2406.16793*.
- Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. 2024. Galore: Memory-efficient llm training by gradient low-rank projection. *arXiv preprint arXiv:2403.03507*.

Algorithm 2 The MUON²-F Optimizer

Require: 2D weights $\mathbf{W}_t \in \mathbb{R}^{n \times m}$, objective $\mathcal{L}$, learning rate η , momentum coefficients β_1, β_2 , Newton–Schulz steps K , numerical constant ϵ

Ensure: Updated weights $\mathbf{W}_{t+1}$

```

1:  $\mathbf{M}_0 \leftarrow \mathbf{0}, \mathbf{r}_0 \leftarrow \mathbf{0} \in \mathbb{R}^n, \mathbf{c}_0 \leftarrow \mathbf{0} \in \mathbb{R}^m$ 
2: for  $t \leftarrow 1, 2, \dots$  do
3:    $\mathbf{G}_t \leftarrow \nabla_{\mathbf{W}_t} \mathcal{L}(\mathbf{W}_t)$ 
4:    $\mathbf{M}_t \leftarrow \beta_1 \mathbf{M}_{t-1} + (1 - \beta_1) \mathbf{G}_t$ 
5:    $\mathbf{R}_t \leftarrow \mathbf{G}_t \odot \mathbf{G}_t$ 
6:    $\mathbf{r}_t \leftarrow \beta_2 \mathbf{r}_{t-1} + (1 - \beta_2) \text{sum}_{\text{col}}(\mathbf{R}_t)$ 
7:    $\mathbf{c}_t \leftarrow \beta_2 \mathbf{c}_{t-1} + (1 - \beta_2) \text{sum}_{\text{row}}(\mathbf{R}_t)$ 
8:    $\hat{\mathbf{V}}_t \leftarrow \frac{\mathbf{r}_t \mathbf{c}_t^\top}{\text{sum}(\mathbf{r}_t)}$ 
9:    $\tilde{\mathbf{M}}_t \leftarrow \mathbf{M}_t \odot (\sqrt{\hat{\mathbf{V}}_t} + \epsilon \mathbf{1})$ 
10:   $\mathbf{O}_t \leftarrow \text{Newton-Schulz}(\tilde{\mathbf{M}}_t, K)$ 
11:   $\mathbf{W}_{t+1} \leftarrow \mathbf{W}_t - \eta \sqrt{m/n} \mathbf{O}_t$ 
12: end for
```

A MUON²-F Algorithm

Note that many approaches have been proposed to reduce the memory overhead introduced by the second-moment matrix (Shazeer and Stern, 2018; Zhang et al., 2024; Zhao et al., 2024; Chen et al., 2023; Dettmers et al., 2021), we leave it as future work for comprehensively studying and comparing among them. In this paper, we adopt Adafactor (Shazeer and Stern, 2018) due to its simplicity and widely demonstrated effectiveness. The full algorithm of MUON²-F is provided in Appendix A.

B Discussion on Cosine Similarity

To elaborate, cosine similarity [Eq. (15)] is robust against adversarial settings, such as in Eq. (8) where a global scaling exists. Furthermore, it delivers an interpretable measurement that lies in $[0, 1]$ which reflects how much the polar approximate aligns with the ground truth direction, where 1 represents a perfect alignment and 0 represents an almost orthogonal direction. In contrast, the exact orthogonality error [Eq. (7)] gives a numerical value that is incident to the actual size of the matrix, and can only be used as a comparative metric for ranking purposes. Last but not least, cosine similarity presents a more practically meaningful measurement. To highlight how different messages these metrics convey, we consider a rather synthetic setting: we evaluate the singular values mapped by NS from a uniform grid of $[0, 1]$, using the follow-

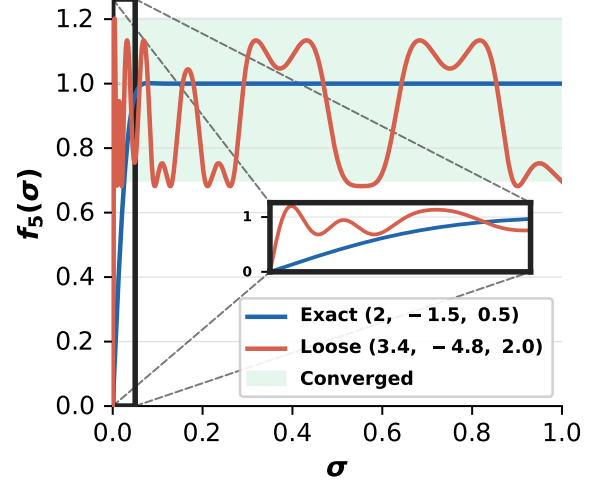


Figure 6: How the NS iteration maps singular values using different coefficients.

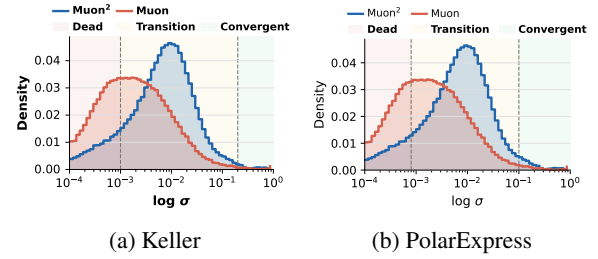


Figure 7: The comparison of convergence zones between Keller (MUON’s (Jordan et al., 2024) polar method) and PolarExpress (Amsel et al., 2026).

ing two sets of coefficients:

- *Loose target*: $(3.4445, -4.7750, 2.0315)$, which is adopted by (Jordan et al., 2024) that roughly maps singular values from $[0, 1]$ to $[0.7, 1.3]$.
- *Exact target*: $(2, -1.5, 0.5)$, an earlier attempt of (Jordan et al., 2024) that vast majority of singular values are mapped to exact one except small values near zero.

MUON favors the *loose target* as it rapidly reaches a practically sufficient orthogonalization with only 5 NS iterations (Figure 6). However, Eq. (7) yields 0.03 for the *exact target* and 0.31 for the *loose target*, a 10x higher error for the latter. This reflects the fact that Eq. (7) measures the exact orthogonality, while in practice, the *loose target* provides sufficient orthogonalization that is well aligned with the ground truth, suggested by the 0.98 cosine similarity given by Eq. (15).

C Convergence Zones for PolarExpress

As per discussion in Section 3.4.1, we divide the singular values of the NS input matrix from $[0, 1]$ into Dead, Transition, and Convergent zones. This partition is determined given particular choices of polar approximation method and the practical tolerance ϵ , where the orthogonalization objective is mapping singular values to $[1 - \epsilon, 1 + \epsilon]$, as suggested by (Jordan et al., 2024). That said, the boundary of each convergence zone can change when different polar methods are used, such as PolarExpress (Amsel et al., 2026). When using PolarExpress, the boundary of dead zone reduces from 0.001 to 0.0008, and the boundary that separates transition and convergent zone reduces from 0.2 to 0.1. However, such changes are insufficient to solve the underlying challenge of the ill-conditioned input matrix. As demonstrate by Figure 7, a substantial part of singular values for MUON with PolarExpress still fall in dead zone, and the advantages of MUON² (i.e., significantly lower dead zone fraction, right-shifted distribution) persist.

D Proofs of Proposition 1 and Corollary 1

Throughout, singular values are ordered decreasingly, $\sigma_1(A) \geq \dots \geq \sigma_m(A) \geq 0$, and for an $m \times n$ matrix A we write $F_{n,A}^\sigma(s) := \frac{1}{m} \# \{i : \sigma_i(A/\sqrt{n}) \leq s\}$ for the empirical CDF of the singular values of $A/\sqrt{n}$. $G(s) := F^{\text{MP}}(s^2)$ for $s \geq 0$ (and 0 for $s < 0$) denotes the singular-value form of the Marchenko–Pastur CDF F^{MP} with ratio λ . Since $\lambda < 1$, the Marchenko–Pastur density is bounded (its support $[(1 - \sqrt{\lambda})^2, (1 + \sqrt{\lambda})^2]$ is bounded away from zero), so F^{MP} and hence G are continuous everywhere, with $G \equiv 0$ on $(-\infty, 1 - \sqrt{\lambda}]$ and $G \equiv 1$ on $[1 + \sqrt{\lambda}, \infty)$.

D.1 Proof of Proposition 1

Step 1 (spectrum under the $\sqrt{n}$ normalization). By construction $\widetilde{M} = (\xi_{ij})$ has i.i.d. zero-mean unit-variance entries — the scales cancel exactly, which is why every conclusion is independent of (s_{ij}) . The almost-sure Marchenko–Pastur theorem (Bai and Silverstein, 2010, Thm. 3.6) applies to $S_n := \frac{1}{n} \widetilde{M} \widetilde{M}^\top$ and yields, on an event Ω_1 of probability one, $F_m^{\text{eig}}(x) \rightarrow F^{\text{MP}}(x)$ for every x . Since $\sigma_i(\widetilde{M}/\sqrt{n}) = \sqrt{\lambda_i(S_n)}$ and squaring is strictly increasing on $[0, \infty)$, for $s \geq 0$ counting gives $F_{n,\widetilde{M}}^\sigma(s) = F_m^{\text{eig}}(s^2) \rightarrow G(s)$; for $s < 0$, both $F_{n,\widetilde{M}}^\sigma(s)$ and $G(s)$ are zero. This proves the first

claim.

Step 2 (Frobenius normalization). Let $T_n := \|\widetilde{M}\|_F^2 = \sum_{ij} \xi_{ij}^2$, a sum of $N := mn$ i.i.d. terms with mean 1 and finite variance $\mathbb{E}\xi^4 - 1$. Chebyshev’s inequality gives $\mathbb{P}(|T_n/N - 1| \geq a) \leq (\mathbb{E}\xi^4 - 1)/(a^2 mn)$, and since $m/n \rightarrow \lambda > 0$ implies $\sum_n \frac{1}{mn} < \infty$, the Borel–Cantelli lemma (applied for $a = 1/k$, $k \in \mathbb{N}$, then intersecting) yields $T_n/N \rightarrow 1$ almost surely; hence $R_n := \|\widetilde{M}\|_F/\sqrt{mn} \rightarrow 1$ on an event Ω_2 of probability one. (If $\|\widetilde{M}\|_F = 0$, define the Frobenius-normalized matrix arbitrarily; since $R_n \rightarrow 1$ almost surely, this convention is eventually irrelevant.) The scaling rule $\sigma_i(cA) = c\sigma_i(A)$ applied twice gives the exact identity

$$\sqrt{m} \sigma_i(\widetilde{M}/\|\widetilde{M}\|_F) = \sigma_i(\widetilde{M}/\sqrt{n}) / R_n.$$

On $\Omega_1 \cap \Omega_2$, fix $\delta \in (0, \frac{1}{2})$; for n large, $1 - \delta < R_n < 1 + \delta$, so the empirical CDF of $\{\sigma_i/R_n\}$ at s is sandwiched between $F_{n,\widetilde{M}}^\sigma(s(1 - \delta))$ and $F_{n,\widetilde{M}}^\sigma(s(1 + \delta))$; letting $n \rightarrow \infty$ and then $\delta \downarrow 0$, continuity of G squeezes the limit to $G(s)$. Finally, for $t < 1 - \sqrt{\lambda}$, the dead-zone fraction $\frac{1}{m} \# \{i : \sigma_i(\widetilde{M}/\|\widetilde{M}\|_F) \leq t/\sqrt{m}\}$ equals this empirical CDF at t , which converges to $G(t) = 0$. $\square$

D.2 Proof of Corollary 1

Case (i). The rank inequality (Bai and Silverstein, 2010, Thm. A.44) bounds the Kolmogorov distance between the eigenvalue ESDs of $\frac{1}{n}(D + \Xi)(D + \Xi)^\top$ and $\frac{1}{n}\Xi\Xi^\top$ by $\text{rank}(D)/m$; since the eigenvalues are the squared singular values and $s \mapsto s^2$ is increasing on $[0, \infty)$, the same bound holds between the singular-value CDFs: $\sup_s |F_{n,D+\Xi}^\sigma(s) - F_{n,\Xi}^\sigma(s)| \leq \text{rank}(D)/m \rightarrow 0$, so the ESD of $(D + \Xi)/\sqrt{n}$ has the same limit G . For the normalization, the triangle inequality and Step 2 give $\limsup_n \|D + \Xi\|_F/\sqrt{mn} \leq d + 1$ almost surely. Fix $t \geq 0$ with $t(1 + d) < 1 - \sqrt{\lambda}$ and $\varepsilon > 0$ small enough that $t(1 + d + \varepsilon) < 1 - \sqrt{\lambda}$; for n large, the dead-zone fraction at threshold $t/\sqrt{m}$ is at most $F_{n,D+\Xi}^\sigma(t(1 + d + \varepsilon)) \leq F_{n,\Xi}^\sigma(t(1 + d + \varepsilon)) + \text{rank}(D)/m \rightarrow G(t(1 + d + \varepsilon)) = 0$. $\square$

Case (ii). By Mirsky’s inequality (Mirsky, 1960), applied with both singular-value sequences in decreasing order (for rectangular matrices, zero-pad both to square ones; see also Horn and Johnson, 2012, Cor. 7.3.5), $\sum_i (\sigma_i((D + \Xi)/\sqrt{n}) - \sigma_i(\Xi/\sqrt{n}))^2 \leq \|D\|_F^2/n = m\delta_n^2$ with $\delta_n^2 := \|D\|_F^2/(mn) \rightarrow 0$. Hence at most a fraction δ_n^2/ε^2

of the indices satisfy $|\sigma_i^{D+\Xi} - \sigma_i^\Xi| \geq \varepsilon$, giving $F_{n,\Xi}^\sigma(s-\varepsilon) - \delta_n^2/\varepsilon^2 \leq F_{n,D+\Xi}^\sigma(s) \leq F_{n,\Xi}^\sigma(s+\varepsilon) + \delta_n^2/\varepsilon^2$; letting $n \rightarrow \infty$ and then $\varepsilon \downarrow 0$, continuity of G gives $F_{n,D+\Xi}^\sigma(s) \rightarrow G(s)$ for every s . Moreover $\|D + \Xi\|_F - \|\Xi\|_F \leq \|D\|_F = o(\sqrt{mn})$, so $\|D + \Xi\|_F/\sqrt{mn} \rightarrow 1$ almost surely, and the argument of Step 2 applies verbatim, yielding every conclusion of Proposition 1. $\square$

Remark on the assumptions. The i.i.d., zero-mean, unit-variance hypotheses enter through the Marchenko–Pastur theorem; the aspect ratio $m/n \rightarrow \lambda > 0$ enters both there and in Step 2, where it gives $\sum_n \frac{1}{mn} < \infty$; $\mathbb{E}\xi^4 < \infty$ enters only through the variance bound in Step 2 (without it, $R_n \rightarrow 1$ holds in probability and all distributional conclusions survive, with “almost surely” weakened accordingly); and $\lambda < 1$ buys the strictly positive lower edge $1 - \sqrt{\lambda}$ — at $\lambda = 1$ the limiting CDF is still continuous, but the edge sits at zero.

Remarks on the scope of the results. (1) The guarantees are asymptotic and are stated at the threshold scale $t/\sqrt{m}$. To connect them with the empirical dead-zone boundary, consider the 512×1344 MLP projections of the LLaMA-60M measured in Section 3.4.1, where $\lambda \approx 0.38$: the limiting lower edge corresponds to $(1 - \sqrt{\lambda})/\sqrt{m} \approx 1.7 \times 10^{-2}$ after the Frobenius normalization, an order of magnitude above the dead-zone boundary 10^{-3} . (2) The claim in Section 3.4.3 that an uneven scale profile can push a large fraction of singular values towards zero is verified by the following construction: if a fraction ρ of rows of M carries a negligible share of $\|M\|_F^2$, then zeroing these rows is a small Frobenius perturbation that reduces the rank to at most $(1 - \rho)m$. Therefore, by Weyl’s inequality for singular values, at least ρm singular values of the normalized raw matrix are correspondingly small. (3) Proposition 1 divides by the exact scale s_{ij} , while MUON² divides by its second-moment estimate. In the weak-signal regime of case (ii) in Corollary 1, the average signal-to-noise ratio $\frac{1}{mn} \sum_{ij} \mu_{ij}^2/s_{ij}^2$ vanishes, so $\mathbb{E}[M_{ij}^2] \approx s_{ij}^2$ for all but a vanishing fraction of entries and the estimate is consistent with the exact scale. Moreover, since the NS input is Frobenius-normalized, only the relative profile of the scales matters, and estimating it up to a common factor suffices. The additive ϵ in the estimate is orders of magnitude smaller than the entries of $\sqrt{V_t}$ (see Appendix E.6) and therefore does not affect the profile. (4) When the scales factorize

as $s_{ij} = r_i c_j$, which is the premise behind the factored second moment of MUON²-F, we have $D = \text{diag}(1/r_i) (\mu_{ij}) \text{diag}(1/c_j)$, and multiplying by invertible diagonal matrices preserves rank. Therefore $\text{rank}(D) = \text{rank}((\mu_{ij}))$: the rank condition of case (i) in Corollary 1 can be checked on the raw signal directly, while the bounded-strength condition remains required. (5) For square layers ($m = n$, $\lambda = 1$), the Marchenko–Pastur limit still applies, but its lower edge sits at zero and the positive margin of Proposition 1 is lost. The same proof still shows that the fraction of singular values below $t/\sqrt{m}$ converges to the limiting CDF at t , which is at most $\frac{2}{\pi}t(1 + o(1))$ under the singular-value law at $\lambda = 1$. Therefore, the preconditioned spectrum places no mass accumulation at zero, whereas an uneven scale profile can place a constant fraction of singular values near zero by the construction in (2).

E Detailed Evaluations

E.1 GPT Models

Model configurations of each scale of the GPT models we considered are listed in Table 7. The learning rate sweeping results and their visualization are in Table 8 and Figure 8 for GPT-Small, Table 9 and Figure 9 for GPT-Base, and Table 10 and Figure 4 for GPT-Large.

Model	n_{embd}	n_{layer}	n_{head}	Param(M)
GPT-Small	768	12	12	124
GPT-Base	1024	24	16	362
GPT-Large	1280	36	20	774

Table 7: Architecture configurations of **GPT** models.

LR	Baseline		Muon ² (ours)	
	$N_s=3$	$N_s=5$	$N_s=3$	$N_s=5$
0.003	34.90	31.18	31.00	30.04
0.005	32.70	29.51	28.83	28.41
0.010	33.49	29.85	28.36	28.01
0.020	33.36	29.86	28.12	27.95
0.040	33.40	30.13	29.30	28.99

Table 8: Learning rate sweep on **GPT-Small**. The best validation perplexity ($\downarrow$) of each method is bolded.

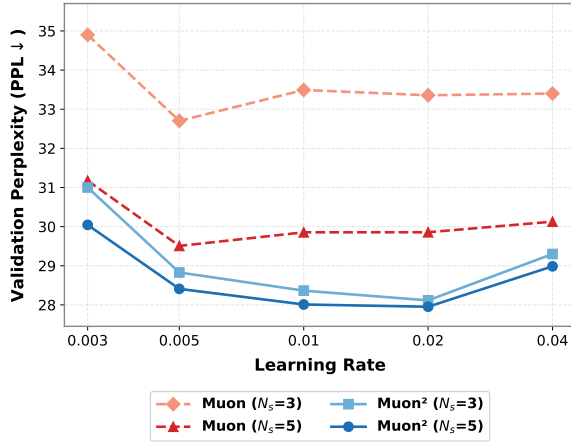


Figure 8: Learning Rates Sweep on **GPT-Small**

LR	Baseline		Muon ² (ours)	
	$N_s=3$	$N_s=5$	$N_s=3$	$N_s=5$
0.003	25.43	22.11	21.73	21.01
0.005	24.69	21.75	20.64	20.33
0.010	25.42	21.99	20.39	19.96
0.020	24.78	21.47	20.50	20.40
0.040	25.75	22.22	21.97	21.24

Table 9: Learning rate sweep on **GPT-Base**. The best validation perplexity ($\downarrow$) of each method is bolded.

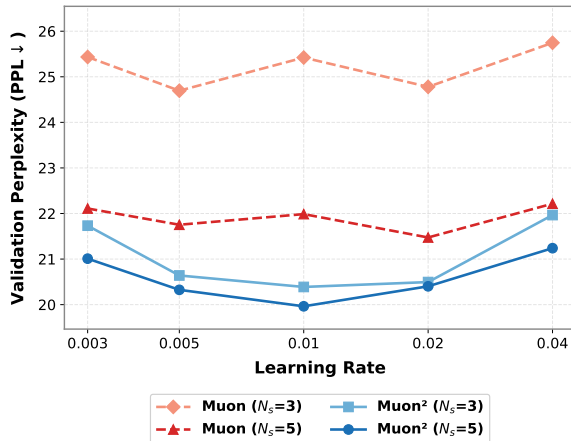


Figure 9: Learning Rates Sweep on **GPT-Base**

LR	Baseline		Muon ² (ours)	
	$N_s=3$	$N_s=5$	$N_s=3$	$N_s=5$
0.003	21.88	18.44	18.14	17.38
0.005	21.13	18.32	17.28	16.82
0.010	21.50	18.32	16.99	16.60
0.020	21.57	17.57	17.28	16.52
0.040	23.77	18.80	18.41	17.55

Table 10: Learning rate sweep on **GPT-Large**. The best validation perplexity ($\downarrow$) of each method is bolded.

E.2 LLaMA Models

Model configurations of each scale of the LLaMA models we considered are listed in Table 11. The learning rate sweeping results and their visualization are in Table 12 and Figure 10 for LLaMA-60M, Table 13 and Figure 11 and for LLaMA-350M, and Table 14 and Figure 12 for LLaMA-1B.

Model	n_{embd}	n_{inter}	n_{layer}	n_{head}	Param(M)
LLaMA-60M	512	1344	8	8	58
LLaMA-350M	1024	2736	24	16	368
LLaMA-1B	2048	5472	24	32	1280
LLaMA-13B	5120	13827	40	40	13018

Table 11: Architecture configurations of **LLaMA** models.

LR	MUON		MUON ² (ours)	
	$N_s=3$	$N_s=5$	$N_s=3$	$N_s=5$
0.02	26.44	25.91	25.90	25.87
0.04	26.37	24.91	24.77	24.68
0.05	27.06	24.94	24.78	24.68
0.06	27.26	24.98	24.59	24.60
0.08	27.35	25.15	24.71	24.71

Table 12: Learning rate sweep on **LLaMA-60M**. The best validation perplexity ($\downarrow$) of each method is bolded.

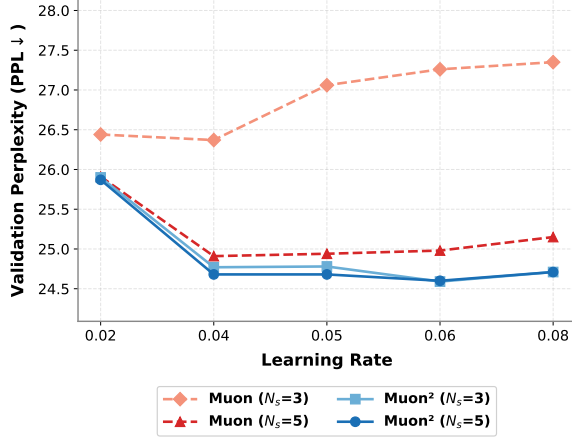


Figure 10: Learning Rates Sweep on LLaMA-60M

LR	MUON		MUON ² (ours)	
	$N_s=3$	$N_s=5$	$N_s=3$	$N_s=5$
0.04	14.91	14.03	13.44	13.46
0.05	15.18	14.12	13.50	13.51
0.06	15.33	14.18	13.58	13.59
0.07	15.43	14.30	13.67	13.63
0.08	15.76	14.49	13.71	13.73

Table 13: Learning rate sweep on LLaMA-350M. The best validation perplexity ($\downarrow$) of each method is bolded.

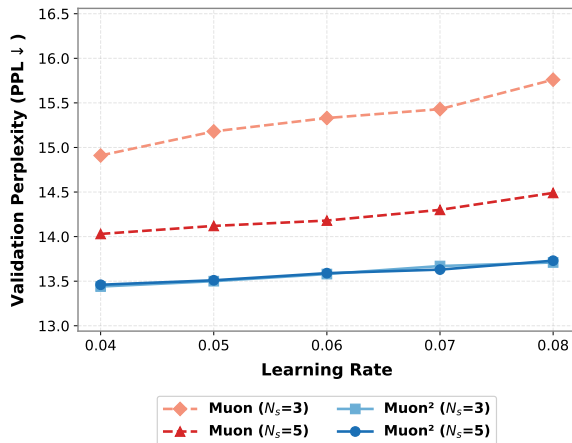


Figure 11: Learning Rates Sweep on LLaMA-350M

LR	MUON		MUON ² (ours)	
	$N_s=3$	$N_s=5$	$N_s=3$	$N_s=5$
0.04	11.63	10.62	10.43	10.21
0.05	11.86	10.70	10.50	10.26
0.06	11.98	10.77	10.55	10.31

Table 14: Learning rate sweep on LLaMA-1B. The best validation perplexity ($\downarrow$) of each method is bolded.

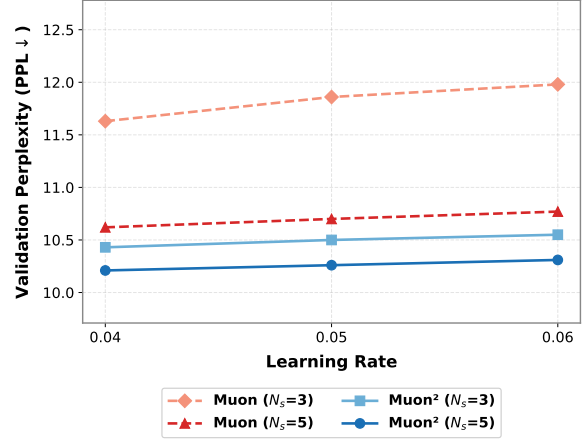


Figure 12: Learning Rates Sweep on LLaMA-1B

E.3 Mixture-of-Experts

Model	n_{embd}	n_{inter}	n_{layer}	n_{head}	n_{exp}	n_{act}	Param(M)
MoE-7B-A1B	2048	4096	24	32	16	2	6980

Table 15: Architecture configurations of Mixture-of-Experts model.

Model configuration of the Mixture-of-Experts model of Section 4.3 is in Table 15.

E.4 Downstream

We evaluate downstream performance on a standard suite of mostly zero-shot benchmarks covering commonsense reasoning, reading comprehension, and knowledge recall. Specifically, we report results on **ARC-Challenge** and **ARC-Easy** (Clark et al., 2018), which test grade-school science questions at two difficulty levels; **OpenBookQA (OBQA)** (Mihaylov et al., 2018), a multi-step reasoning benchmark requiring elementary science facts; **HellaSwag** (Zellers et al., 2019), which evaluates commonsense sentence completion; **PIQA** (Bisk et al., 2020), focused on physical commonsense reasoning; **WinoGrande** (Sakaguchi et al., 2021), a large-scale

Winograd-style coreference resolution benchmark; **LAMBADA** (Paperno et al., 2016), which measures long-range contextual word prediction; and **MMLU** (Hendrycks et al., 2020), covering multitask language understanding across 57 subjects. We report accuracy for all tasks and the unweighted average (Avg) across the suite. See additional downstream evaluations on GPT-Large in Table 17.

Fine-tuned evaluations on LLaMA-1B. At this model scale, zero-shot accuracy on ARC-Challenge, WinoGrande, and MMLU remains close to the random-guessing baseline (25%, 50%, and 25%, respectively), so zero-shot comparisons on these tasks carry little discriminative signal. For the three tasks marked with [†] in Table 4, we therefore fine-tune each pretrained LLaMA-1B model on the corresponding training data before evaluation, applying the identical protocol to every optimizer. Fine-tuning minimizes the language-modeling loss on answer tokens only, using AdamW (Loshchilov and Hutter, 2017) with weight decay 0.01, $\beta = (0.9, 0.95)$, a cosine schedule with 3% warmup, gradient clipping at 1.0, and bf16 mixed precision. For WinoGrande we use the official training set (40k examples; 2 epochs, batch size 32); for ARC-Challenge, the combined ARC-Easy and ARC-Challenge training splits (3.4k examples; 3 epochs, batch size 16); for MMLU, the auxiliary training set (77k examples truncated to 512 tokens; 1 epoch, batch size 32). For each model and task, the fine-tuning learning rate is swept over $\{0.5, 1, 2, 3, 5, 10\} \times 10^{-5}$ (up to 3×10^{-5} for MMLU), and we report the best configuration averaged over three random seeds. Evaluation follows the same harness as the zero-shot tasks (Gao et al., 2021, v0.4.12): accuracy at 0-shot for ARC-Challenge and WinoGrande, and at 5-shot for MMLU following standard practice. Table 16 reports means and standard deviations across seeds for the sweep around the optimum: the sweep is first run with a single seed, and the two best learning rates per task are then repeated with three seeds. Comparing each model at its best learning rate, MUON² consistently outperforms MUON at matched N_s on all three tasks.

Fine-tuned evaluations on GPT-Large. We apply the same protocol to the GPT-Large models and mark the fine-tuned tasks with [†] in Table 17, with two adaptations to the architecture: evaluation uses a context length of 1024, the maximum position of GPT-2, and MMLU fine-tuning enables activation

checkpointing to keep batch size 32. For each task, the two learning rates that were optimal on LLaMA-1B are repeated with three seeds; Table 18 reports the sweep. The conclusions from LLaMA-1B carry over: MUON² outperforms MUON at matched N_s on ARC-Challenge and WinoGrande, and MUON² at $N_s = 3$ matches MUON at $N_s = 5$ on ARC-Challenge while exceeding it on WinoGrande. The fine-tuning also resolves an artifact of the zero-shot table: the zero-shot WinoGrande column ranked MUON ($N_s = 3$) highest (52.09), yet after fine-tuning the same model trails every other model by more than 4 points, confirming that rankings near the random baseline carry no signal. MMLU is the exception at this scale: after fine-tuning, all four models remain at the chance level (25–27%), so we report the numbers for completeness but draw no comparison from MMLU on GPT-Large. We attribute this primarily to the model and data scale, as the larger LLaMA-1B with a 4096 context reaches only $\sim 31\%$, and secondarily to the 1024 context, which truncates about 19% of the 5-shot prompts.

E.5 MUON Variants

We compare with MUON variants including PolarExpress (Amsel et al., 2026), Turbo-Muon (Boissin et al., 2025), NorMuon (Li et al., 2025) and AdaMuon (Si et al., 2025). We focus on comparing them with MUON² on GPT-Small and GPT-Base. We integrated their methods into our training framework using exactly what have been provided in their official repositories. For fairness, all methods are being swept on learning rate to make sure each method performs at their best capabilities, full sweeping results are in Tables 19 to 21. The reason we have AdaMuon separately in Table 21 is because it requires significantly smaller learning rates than other methods.

LR	PolarExpress		Turbo-Muon		NorMuon	
	$N_s=3$	$N_s=5$	$N_s=3$	$N_s=5$	$N_s=3$	$N_s=5$
0.003	31.51	30.45	31.36	30.31	32.50	29.90
0.005	30.01	29.42	29.74	29.66	30.70	28.40
0.010	31.32	29.73	29.70	29.79	31.43	28.72
0.020	30.31	29.66	30.10	29.88	31.05	28.48
0.040	31.25	30.03	30.07	30.82	30.35	29.09

Table 19: Learning rate sweep on **GPT-Small**. The best validation perplexity ($\downarrow$) of each method is bolded.

		ARC-c		WinoG.		MMLU	
NS Steps		1×10^{-5}	2×10^{-5}	2×10^{-5}	3×10^{-5}	2×10^{-5}	3×10^{-5}
MUON	$N_s = 3$	25.20 $\pm$ 0.05	27.08 $\pm$ 0.26	56.70 $\pm$ 0.54	56.75 $\pm$ 0.72	29.21 $\pm$ 0.86	29.75 $\pm$ 0.32
	$N_s = 5$	29.27 $\pm$ 0.15	30.01 $\pm$ 0.05	59.85 $\pm$ 0.32	59.91 $\pm$ 0.83	30.24 $\pm$ 0.13	30.71 $\pm$ 0.41
MUON ²	$N_s = 3$	31.91 $\pm$ 0.39	31.66 $\pm$ 0.39	59.88 $\pm$ 0.93	59.30 $\pm$ 0.16	30.72 $\pm$ 0.51	30.79 $\pm$ 0.82
	$N_s = 5$	32.65 $\pm$ 0.36	33.48 $\pm$ 0.36	61.90 $\pm$ 1.16	61.67 $\pm$ 1.13	31.31 $\pm$ 0.78	30.60 $\pm$ 0.64

Table 16: Fine-tuning learning-rate sweep on LLaMA-1B for the three fine-tuned tasks (mean and standard deviation over three seeds). For each model and task, the best configuration (bold) is the one reported in Table 4. Comparing each optimizer at its best learning rate, MUON² outperforms MUON at matched N_s on every task.

	NS Steps	Avg	ARC-c [†]	ARC-e	OBQA	HellaS.	PIQA	WinoG. [†]	LAMBADA	MMLU [†]
MUON	$N_s = 3$	40.69	27.99	44.23	30.80	39.92	68.82	54.56	32.19	27.02
	$N_s = 5$	42.07	28.21	44.07	30.40	42.50	69.80	59.12	37.07	25.41
MUON ²	$N_s = 3$	42.90	28.30	44.91	29.40	43.72	69.70	60.77	40.03	26.38
	$N_s = 5$	43.70	30.26	46.93	31.60	45.01	69.80	60.38	40.07	25.52

Table 17: Downstream evaluation on GPT-Large trained by MUON and MUON² across different NS iterations. Tasks marked with [†] are fine-tuned before evaluation under the protocol of Appendix E.4; the remaining tasks are zero-shot. MMLU is not bolded, as all four models remain at the chance level at this scale even after fine-tuning.

LR	PolarExpress		Turbo-Muon		NorMuon	
	$N_s=3$	$N_s=5$	$N_s=3$	$N_s=5$	$N_s=3$	$N_s=5$
0.003	23.10	21.80	24.48	22.79	24.70	21.89
0.005	22.75	21.39	23.95	22.51	23.90	21.54
0.010	23.38	21.44	24.14	22.42	24.18	21.42
0.020	22.74	21.16	23.46	21.93	23.33	21.27
0.040	23.70	22.43	26.23	22.72	24.25	21.64

Table 20: Learning rate sweep on **GPT-Base**. The best validation perplexity ($\downarrow$) of each method is bolded.

LR	GPT-Small		GPT-Base	
	$N_s=3$	$N_s=5$	$N_s=3$	$N_s=5$
0.001	36.63	31.49	26.07	22.73
0.003	31.20	29.30	27.91	22.94
0.005	33.45	30.48	26.08	22.42

Table 21: Learning rate sweep of **AdaMuon** on **GPT-Small** and **GPT-Base**. The best validation perplexity ($\downarrow$) of each method is bolded.

E.6 Sensitivity to β_2 and ϵ

MUON² introduces two hyperparameters over MUON: the second-moment decay β_2 and the numerical constant ϵ . We assess sensitivity on LLaMA-60M under the paper’s training setup, varying one at a time around the defaults ($\beta_2 = 0.95$, $\epsilon = 10^{-8}$); as a reference, we apply the identical protocol to MUON’s own momentum coefficient. Results are in Table 22.

MUON² is robust to both hyperparameters: all settings fall within 0.15 PPL of each other, and every setting outperforms base MUON (24.98). The robustness to ϵ is structural rather than tuned: ϵ enters only through the divisor $\sqrt{V_t} + \epsilon 1$, so it can matter only when comparable to the entries of $\sqrt{V_t}$; every conventional choice (10^{-10} to 10^{-6}) sits orders of magnitude below that scale. Unlike in Adam, ϵ also does not couple with the step size, because the NS iteration normalizes away the overall scale of its input. By contrast, MUON is markedly more sensitive to its momentum coefficient than MUON² is to either β_2 or ϵ .

E.7 Ablation: Preconditioning without Orthogonalization

To separate the improvement in NS convergence from a generic benefit of second-moment preconditioning, we ablate the NS iteration entirely: the preconditioned momentum $\tilde{M}_t$ is used directly as the update direction. At its best configuration on LLaMA-60M, this no-NS variant reaches a validation perplexity of 27.14 — far behind both MUON² (24.59) and base MUON (24.98). Second-moment preconditioning used directly as the update therefore does not deliver the gain; the improvement of MUON² comes specifically from feeding the orthogonalization a better-conditioned input.

		ARC-c		WinoG.		MMLU	
NS Steps		1×10^{-5}	2×10^{-5}	2×10^{-5}	3×10^{-5}	2×10^{-5}	3×10^{-5}
MUON	$N_s = 3$	27.99 ± 0.77	26.31 ± 0.68	54.56 ± 0.75	53.85 ± 0.64	27.02 ± 0.55	26.62 ± 1.27
	$N_s = 5$	28.21 ± 0.05	26.96 ± 0.31	58.93 ± 1.11	59.12 ± 1.03	25.41 ± 0.24	25.22 ± 0.62
MUON ²	$N_s = 3$	28.30 ± 0.44	28.01 ± 1.02	60.77 ± 0.60	60.43 ± 1.01	25.62 ± 0.11	26.38 ± 0.85
	$N_s = 5$	30.26 ± 0.65	28.61 ± 0.43	60.27 ± 0.46	60.38 ± 0.34	25.17 ± 0.46	25.52 ± 1.18

Table 18: Fine-tuning learning-rate sweep on GPT-Large (mean and standard deviation over three seeds). For ARC-Challenge and WinoGrande, the best configuration per model (bold) is the one reported in Table 17. MMLU values are not bolded, as all of them remain at the chance level.

Setting	Val. PPL
default ($\beta_2=0.95$, $\epsilon=10^{-8}$)	24.59
MUON ² $\beta_2 = 0.9$	24.72
$\beta_2 = 0.99$	24.69
$\epsilon = 10^{-6}$	24.74
$\epsilon = 10^{-10}$	24.56
MUON momentum = 0.95 (default)	24.98
momentum = 0.9	25.74
momentum = 0.99	26.63

Table 22: Hyperparameter sensitivity on LLaMA-60M. Every MUON² setting stays within 0.15 PPL of the default and outperforms MUON, whereas MUON is substantially more sensitive to its own momentum coefficient.

E.8 Spectral Analysis on GPT-2 Small

The spectral analyses of Section 3.4.1 are conducted on a single model family. We repeat the full analysis on GPT-2 Small with identical definitions; Table 23 summarizes the results. The conclusions carry over: MUON² keeps roughly $4\times$ fewer singular values in the dead zone, about triples the effective rank of the NS input, and with 3 NS steps reaches the directional alignment that MUON needs 5 steps for (0.84 vs. 0.86). The gap is largest early in training: at step 50, 94.8% of MUON’s singular values fall in the dead zone (effective rank 2.1), versus 47.1% for MUON² (effective rank 8.5).

We additionally use this setting to examine whether update-scale fluctuations, which directional alignment deliberately ignores, could affect the effective learning rate: across 48 weight matrices and 38 logged steps, the coefficient of variation of the per-step update RMS is 0.06 for MUON² versus 0.20 for MUON. This is expected: the update magnitude is set by the fixed post-NS rescaling shared by both methods, so any residual fluctuation stems from imperfect orthogonalization, of which

Metric	MUON	MUON ²
Dead-zone ratio	0.26	0.06
Transition-zone ratio	0.73	0.93
Effective rank	48.9	142.5
Directional alignment @ $N_s=1$	0.49	0.64
Directional alignment @ $N_s=3$	0.69	0.84
Directional alignment @ $N_s=5$	0.86	0.96

Table 23: Spectral analysis of Section 3.4.1 repeated on GPT-2 Small (identical definitions).

MUON²’s better-conditioned input leaves less.

E.9 Training Speedup

Here we show the wall-time vs. training loss in Figure 13, and the per-training-step time in Table 24.

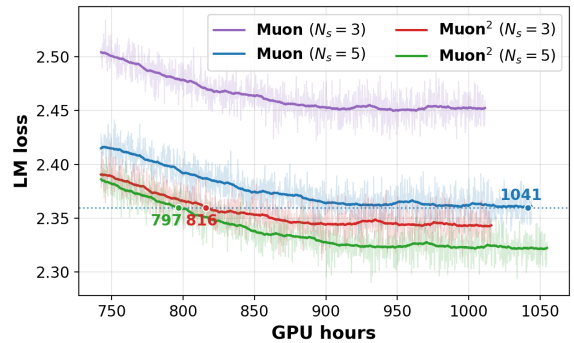


Figure 13: On LLaMA-1B, MUON² achieves same loss with 1/4 fewer GPU-hours than MUON.

Model	NS Steps	Step Time (ms)
MUON	$N_s = 3$	2958
	$N_s = 5$	3053
MUON ²	$N_s = 3$	2971
	$N_s = 5$	3085

Table 24: Per-training-step time for LLaMA-1B on a supercomputer with 64 nodes (4 GPUs each).

E.10 Memory Efficiency of MUON²-F

The memory profiling shown in Table 6 is conducted on a single 94GB H100 GPU using training framework adopted from Nanotron (Karpathy, 2022) with bfloat 16 precision, sequence length 4096, and micro batch size 1.

F LLM Usage

We used large language models solely as writing assistance tools for grammar and wording refinement. They were not involved in the development of ideas, experimental design, analysis, or interpretation of results. All technical content, claims, and conclusions were produced and verified by the authors.