From SGD to Muon: Adaptive Optimization via Schatten-p Norms

Thomas Massena[*]
IRIT & SNCF

Corentin Friedrich
IRT Saint Exupery

Mathieu Serrurier
IRIT

Abstract

Modern optimizers, like Muon, impose matrix-wise geometry constraints on their updates. These matrix-wise constraints can be unified under Linear Minimization Oracle (LMO) theory. However, all current methods impose fixed LMO geometries for the update rules, chosen by-design or empirically, which are not necessarily optimal according to the problem's geometry. We introduce a novel efficient data-driven criterion for dynamically choosing proxy-optimal update LMO geometries on individual Deep Neural Network layers. Derived in closed form from gradient and activation statistics using a single-step random feature regression surrogate model, our criterion navigates a design space interpolating from SGD to Muon updates. Moreover, integrating parameter-wise preconditioning allows our framework to recover SGD, Muon, Adam, and MuAdam as specific extrema. To make this adaptive approach scalable, we pair it with efficient computational strategies, achieving only a $\sim 3\%$ runtime overhead on highly optimized baselines. As a proof of concept, we show that this data-driven optimizer beats or remains competitive with the performance of the best performing optimizer between Muon and AdamW across three different training scenarios. Ultimately, this work provides evidence that LMO geometry can be successfully and efficiently adapted from runtime data, opening a new pathway for optimizer design beyond static geometries.

1 Introduction

The training of deep neural networks has traditionally been dominated by coordinate-wise adaptive optimizers like Adam and its variants [Kingma and Ba, 2015, Loshchilov and Hutter, 2019]. By scaling gradient updates element-by-element, these methods are highly effective at managing heterogeneous gradient scales and stochastic noise. However, by treating parameters as flattened vectors, they inherently ignore the matrix geometries and structural dependencies present in the transformer-dominated neural architecture landscape [Vaswani et al., 2017]. Recently, the Muon optimizer emerged as a powerful alternative that explicitly targets the matrix geometry of hidden layers [Jordan et al., 2024b]. Instead of parameter-wise scaling, Muon updates weight matrices by using the closest orthogonal matrix to the gradient matrix in Frobenius norm (also known as the matrix polar factor). This operation removes anisotropic scaling and enforces gradient orthogonality, leading to strong empirical performance, even at large scales for large language model pretraining [Liu et al., 2025]. Despite these breakthroughs, the purely geometric approach of Muon exhibits notable vulnerabilities. By treating all singular values equally (effectively applying a flattening operation to the spectrum), Muon lacks the variance tracking and coordinate-wise adaptivity that makes Adam robust. Consequently, Muon can magnify noisy gradient directions on ill-conditioned loss landscapes [Gonon et al., 2026, Li et al., 2025, Si et al., 2025, Dragutinović and Ranganath, 2026].

All existing approaches rely on a heuristically fixed update geometry described by a fixed norm Linear Minimization Oracle [Pethick et al., 2025]. In this paper, we propose a generalization that bridges the

[*]Correspondence to thomas.massena@irit.fr

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

gap between coordinate-wise adaptivity and spectral geometry. Building on the insight that standard Euclidean descent (SGD) and spectral descent (Muon) are not competing paradigms, but rather the extreme ends of a broader family of updates defined under Schatten-p norms [Schatten, 1950, Cesista, 2025, Peyré, 2026], we consider a single-step random feature regression model surrogate, previously characterized in Davis and Drusvyatskiy [2025] for estimating optimal update geometries p^* based on first-order accessible quantities, i.e: activations, gradients and momenta.

Our contributions are the following:

- First, we provide a first-order theoretically principled, single-step, random feature regression model that allows us to pick layer-wise optimal Schatten-p norm constraint descent rules according to the geometry of the problem (Section 3).
- Then, we study the stability of this p^* proxy across layers throughout training, and observe stable evolutions that undergo sudden phase changes during training. Leveraging this, we find strategies to update layer-wise p^* values in a stable and efficient manner throughout training (Section 4).
- Additionally, we provide a method for the estimation of the optimal p^*-dependent update; building on state of the art polar factor estimation methods to minimize runtime overhead and allow for competitive geometry-aware optimization (Section 5).
- Finally, we validate our method by running a set of diverse experiments, showing that our optimizer matches or surpasses the best performing optimizer between Muon and AdamW in a variety of experimental settings (Section 6).

Our code is provided on GitHub.

2 Theoretical Background and Related Work

First, let us recall the definition of a Schatten-p norm.

Definition 1 (Schatten-p Norm)**.** *Let $X \in \mathbb{R}^{m\times n}$, with singular values $\sigma_i(X)$. For $p \in [1,\infty)$, the Schatten p-norm of X, denoted $\|X\|_{S_p}$, is defined as the ℓ_p-norm of its singular values:*

$$\|X\|_{S_p} := \left(\sum_{i=1}^{\min(m,n)} \sigma_i^p(X) \right)^{\frac{1}{p}} \tag{1}$$

In the following section, we provide the theoretical background necessary to our method.

2.1 Generalizing Muon and SGD as Schatten-p Norm LMOs

Following insights from Peyré [2026], we provide a generalization of both SGD and Muon's update rules under the framework of Schatten-p norm constrained gradient descent [Bernstein and Newhouse, 2024, Pethick et al., 2025]. First, let us introduce the framework of Linear Minimization Oracles (LMOs). Given a gradient $G = \nabla_W \mathcal{L}$ with singular value decomposition $U\Sigma V^T$, the LMO aims to solve:

$$\mathrm{LMO}_{\mathcal{D}}(G) := \mathrm{argmin}_{\delta W \in \mathcal{D}} \langle G, \delta W \rangle_F \tag{2}$$

where $\mathcal{D}$ is the norm ball defined as $\mathcal{D} := \{x \mid \|x\| \leq \rho\}$, with $\rho > 0$ and $\|\cdot\|$ some valid norm [Pethick et al., 2025]. Denoting the Schatten-p norm unit ball as $\mathcal{B}_{S_p} = \{X \in \mathbb{R}^{m\times n} \mid \|X\|_{S_p} \leq 1\}$, we can express the exact LMO update directions. For any gradient $G \in \mathbb{R}^{m\times n}$, there exist strictly positive scaling factors $\alpha_\infty, \alpha_2 \in \mathbb{R}_{>0}$ such that:

$$\begin{aligned} \mathrm{argmin}_{\delta W \in \mathcal{B}_{S_\infty}} \langle G, \delta W \rangle_F &= -\alpha_\infty U V^T \qquad \text{(Muon)} \\ \mathrm{argmin}_{\delta W \in \mathcal{B}_{S_2}} \langle G, \delta W \rangle_F &= -\alpha_2 U \Sigma V^T \qquad \text{(SGD)} \end{aligned} \tag{3}$$

both the analogous Muon and SGD updates are part of a continuous framework of Schatten constrained norm updates. More generally, we find:

Proposition 1. *Let $G \in \mathbb{R}^{m\times n}$ have singular value decomposition $G = U\Sigma V^T$, with sorted, positive and nonzero singular values. For any $p \in [1, \infty)$, the solution to the constrained problem, satisfies:*

$$\exists\, \alpha_p \in \mathbb{R}_{>0},\ s.t. \operatorname{argmin}_{\delta W \in \mathcal{B}_{S_{p+1}}} \langle G, \delta W\rangle_F = -\alpha_p U\Sigma^{1/p}V^T \tag{4}$$

Our proof is given in Appendix A.1 as the general form of Schatten-$(p+1)$ norm updates, admitting SGD ($p = 1$) and Muon ($p = \infty$) as extrema. In this paper, we propose to dynamically adjust the LMO update rule in an adaptive and layer-wise manner to improve the training of deep neural networks.

2.2 Random Feature Regression

In this section, we introduce the random feature regression setting from Davis and Drusvyatskiy [2025], which the authors use to characterize the regimes that advantage spectral updates (Muon) over Euclidean updates (SGD) by comparing their guaranteed loss decrease under optimizer curvature constraints. First, let us define the studied random feature regression setting:

$$\min_{W\in\mathbb{R}^{m\times n}} \mathcal{L}(W) := \frac{1}{2k}\|WA - Y\|_F^2 \tag{5}$$

where $W \in \mathbb{R}^{m\times n}$ is the l-th layer's weight matrix, $A \in \mathbb{R}^{n\times k}$ is the post-activation matrix of the previous layer and $Y \in \mathbb{R}^{m\times k}$ the expected activation matrix of the l-th layer. This model is standard and motivated for analytically tractable simplification of neural network behavior (c.f. [Cho and Saul, 2009, Daniely et al., 2016, Rahimi and Recht, 2007, 2008, Rudi and Rosasco, 2017]). In this setting, for a weight perturbation δW, the loss can be approximated via:

$$\mathcal{L}(W + \delta W) - \mathcal{L}(W) = \langle G, \delta W\rangle_F + \frac{1}{2k}\|\delta W.A\|_F^2 \tag{6}$$

In this setting, Davis and Drusvyatskiy [2025] show that the Euclidean descent update (i.e. SGD) relies on the operator norm inequality $\|\delta W.A\|_F \le \|\delta W\|_F\|A\|_2$, whereas the spectral descent case (i.e. Muon) relies on the Frobenius norm inequality $\|\delta W.A\|_F \le \|\delta W\|_2\|A\|_F$. Ultimately, the authors draw the following conclusion: Spectral updates outperform SGD when the *squared Nuclear-to-Frobenius ratio of the gradient* exceeds the *Stable Rank of the activations* (i.e. $\|G\|_*^2/\|G\|_F^2 > \|A\|_F^2/\|A\|_2^2$, with $\|\cdot\|_*$ denoting the nuclear norm). Importantly, the insights from the random feature regression model are shown to hold during stochastic training of deep neural networks.

2.3 Related Work

First-order diagonal methods (Adam [Kingma and Ba, 2015], AdamW [Loshchilov and Hutter, 2019]) rescale gradients per-coordinate, ignoring matrix structure. A richer class maintains structured curvature approximations: Shampoo [Gupta et al., 2018], SOAP [Vyas et al., 2024], and PSGD [Li, 2017] with its Lie-group extensions [Pooladzandi and Li, 2024] are properly understood as second-order or quasi-second-order methods that exploit Kronecker-factored Hessian or Fisher statistics. Bernstein and Newhouse [2024] show that after disabling accumulation, some of these methods reduce to steepest descent under specific operator norms, reframing them as norm-aware first-order updates and motivating a principled LMO design space further developed in Pethick et al. [2025] and Veprikov et al. [2025]. Muon [Jordan et al., 2024b, Liu et al., 2025] implements this for the spectral norm via polar-factor orthogonalization. Its lack of coordinate-wise adaptivity on ill-conditioned landscapes [Gonon et al., 2026, Dragutinović and Ranganath, 2026] has spurred hybrid variants (MuAdam [Veprikov et al., 2025], AdaMuon [Si et al., 2025], NorMuon [Li et al., 2025], MuonEq [Chang et al., 2026]) and fixed-p alternatives [Qi et al., 2026]. Du and Su [2026] provide the only theoretical justification for a specific LMO choice, by accomodating the update rule of the optimizer step to use a right preconditioning by the inverse of the activations Gram matrix, they show that $p^\star = \infty$ is optimal under an isotropy assumption on the weights. We instead derive a closed-form, layerwise $p^\star$ from quantities any first-order optimizer already tracks without requiring any specific preconditioning, continuously interpolating between SGD and Muon.

3 Finding the Optimal Geometry

Here, we extend the random feature regression analysis of Davis and Drusvyatskiy [2025] to derive a proxy-optimality criterion for the Schatten-$(p+1)$ update geometry.

3.1 Finding an Optimal Update Geometry under First-Order Moments

Rather than analyzing a moment-free single-step update (which would yield a clean theorem about a moment-less optimizer that is not used in practice) we tie the descent guarantee to a first-order momentum buffer and, optionally, an element-wise second-order preconditioner (c.f. Section 3.2). *We consider the case where the weight update is tied to a momentum buffer* that aggregates past gradients using an Exponential Moving Average (EMA) mechanism. We introduce the following momentum $M \leftarrow \beta M + (1-\beta)G$, with singular value decomposition $U_M \Sigma_M V_M^T$. The Schatten-$(p+1)$ update, in this case, becomes: $\delta W = -\eta U_M \Sigma_M^{1/p} V_M^T$. Here, our goal is to find p^* such that the loss decrease, independently of scaling is maximized:

$$\mathcal{L}(W + \delta W) - \mathcal{L}(W) = -\eta \langle G, U_M \Sigma_M^{1/p} V_M^T \rangle_F + \frac{\eta^2}{2k} \| U_M \Sigma_M^{1/p} V_M^T \cdot A \|_F^2, \tag{7}$$

Using the trace cyclic property to reformulate the first and second term of the right hand side, we provide the following result for the optimal Schatten-$(p+1)$ norm geometry according to the spectral statistics of activations, gradients and momenta.

Proposition 2 (Optimal p^* Proxy)**.** *For all $p \in [1, \infty)$, the value of p that maximizes the guaranteed descent of Equation* (7) *under optimal step size the Schatten-$(p+1)$ update can be formulated as:*

$$p^* = \mathrm{argmax}_{p \in [1,\infty]} \frac{\left(\sum_i C_{i,i} \sigma_{M,i}^{1/p} \right)^2}{\sum_i \sigma_{M,i}^{2/p} B_{i,i}} \tag{8}$$

with $C_{i,i} = U_M^T G V_M$ and $B_{i,i} = \|A^T V_M e_i\|^2$ defined previously, G the raw unprojected gradient and A the data flowing inside the layer l that uses weights W.

Our complete proof is given in Appendix A and the unimodality of the underlying argmax objective is discussed in A.2.7, along with the stability of the optimal step size in B.

Remark Proposition 2 relies on the random feature regression surrogate model, which is intentionally simplified and therefore approximate in practical deep learning settings. We aim for a simple layerwise rule that selects p^* using only quantities that are already available to a first-order optimizer (gradient, momentum, and activations) and that applies uniformly to every matrix parameter in a deep network without needing knowledge of the definition of the forward pass tied to a parameter.

3.2 Adding Second-Order Moments

In the previous section, we provided a surrogate method for the choice of optimal layer-wise p^* update geometries, allowing for a principled choice of SGD-like or Muon-like update rules. However, this interpolation does not only impact the geometry of the update but also its stability properties. Indeed, in the high p^* regime, Muon enforces a flattening of the singular value spectra, this provides explicit control of update magnitude and anisotropy at matrix level. As p^* decreases, this spectral constraint is progressively relaxed, and the training process becomes increasingly sensitive to coordinate-wise scale disparities and stochastic gradient noise. This motivates the incorporation of an explicit variance control mechanism that provides stability in lower p^* regimes. We retain this benefit by combining p^* as in Proposition 1, and using an Adam-style accumulator $D \leftarrow \beta_2 D + (1-\beta_2) G \odot G$, where $\odot$ denotes the Hadamard product, to scale the variance of updates. The interface between this preconditioner and the derivations of Section 3 is examined more in depth in Appendix D. Finally, as argued in Qi et al. [2026], optimization with different update geometries is more stable in the presence of second-order moments, further justifying the necessity for second order moments in lower p^* regimes.

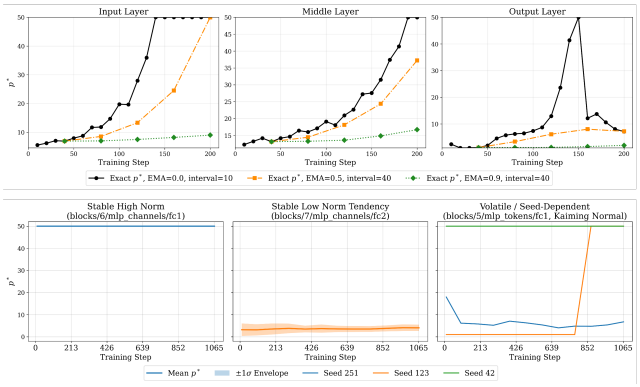


Figure 1: (top) The optimal p^* values on the input, hidden and output layer of an MLP model trained on the MNIST dataset. (bottom) Optimal layerwise p^* values can exhibit stable or volatile behaviour on an MLP-Mixer trained on the ImageNette dataset.

Here, we rely on the unifying work of Veprikov et al. [2025] to add second order moment preconditioning to optimization methods that rely on norm constrained updates [Pethick et al., 2025]. We propose the following p-dependent heuristic to recover Adam updates when $p = 1$ and Muon when $p \to \infty$.

$$\delta W = -\eta \, D^{\circ -1/(2(p+1))} \odot \mathrm{LMO}_{(p+1)}\Big(D^{\circ -1/(2(p+1))} \odot M\Big), \tag{9}$$

where $\mathrm{LMO}_{(p+1)}(X) \propto U\Sigma^{1/p}V^T$ denotes the Schatten-$(p+1)$ norm LMO update. Here, the choice of the exponent $1/(2(p+1))$ is the unique choice that: (i) renders the update invariant to global rescaling of the loss; a property as a key desideratum identified by [Veprikov et al., 2025]; (ii) while allowing the update rule to interpolate between that of Adam and that of Muon. The derivation of this invariance motivated scaling is given in Appendix D. Throughout this paper, we will denote the LMO update with no second order moments (recovering SGD when $p = 1$ and Muon when $p \to \infty$) as SMuon (standing for Schatten-Muon), while the variance controlled LMO update with Adam updates when $p = 1$ that uses p-dependent moments will be denoted as SMuon (Adam).

4 Estimating Optimal p Values in Practice

In this section, we compute optimal p^* values across different settings. Here, we use $p_{\max} = 50$ as the maximum p^* computable value for each layer since $U\Sigma^{1/50}V^T \approx UV^T$ can be considered to be numerically identical to Muon updates for our purposes. These experiments reveal three important phenomena. Firstly, optimal p^* values seem locally stable, with some sudden phase changes. Secondly, layer-wise p^* values with a varying degree of temporal sampling and optimal p^* computations are compatible with the use of different update frequencies depending on the problem. In order to characterize these phase changes in sparser temporal sampling scenarios, we propose the use of an EMA mechanism with strength β_p on the underlying quantities of the p^* computation (i.e., $C_{i,i}$, $\sigma_{M,i}$ and $B_{i,i}$).

Why smooth the spectral quantities rather than $p^\star$? We apply the EMA to $\{C_{i,i}, \sigma_{M,i}, B_{i,i}\}$ rather than directly to $p_t^\star$ for two reasons. First, $p^\star$ is the argmax of a nonlinear ratio of these quantities, and argmax does not commute with averaging. Therefore, smoothing the inputs and re-solving for $p^\star$ provides a more natural plug-in estimator, while smoothing $p_t^\star$ has no such interpretation. Second, the empirically observed trajectories of $p^\star$ (Fig. 1) exhibit abrupt phase transitions between locally stable regimes; smoothing the inputs lets the underlying state evolve continuously while $p^\star$ snaps to whichever regime currently dominates, whereas smoothing $p_t^\star$ could drag the estimate through intermediate values that are themselves suboptimal. The parameter β_p then controls a standard bias–variance tradeoff between minibatch noise and lag behind genuine geometry shifts balancing the two on the workloads we consider.

Low Frequency p^* Updates (fig. 1, top) In a toy experiment, we plot the evolution of the optimal p^* value for an MLP network [Rosenblatt et al., 1962] trained on 200 steps on the MNIST dataset [LeCun et al., 1989] with learning rate 10^{-2} and batch size 4096. We plot three curves of optimal p^* values for all layer parameters inside the network in Figure 1 (top): The exact optimal p^* curve, computed every 10 steps (i.e., the exact, high frequency update), is plotted as a black line. A lower-frequency p^* estimation, computed every 40 steps, that uses $\beta_p = 0.5$ to smooth out slight local p^* variations, is plotted in orange. Finally, a smoother estimate, with $\beta_p = 0.9$, that also is computed every 40 steps is displayed in green. The obtained p^* values are given in fig. 1 (top). In this setting, we notice stable and predictable evolutions of p^* values for each layers, that can undergo sudden phase changes. Also, we recover that using EMA smoothing on the spectral statistics for the computation of p^* can help smooth out the evolution of p^* throughout training; which is rather practical if we are to update the layers with p^* dependent update rules, as we don't want the update rule to fluctuate too randomly during training. Finally, we see that for a rightly chosen EMA strength — here 0.5, in a rapidly decreasing loss landscape where the optimization geometry shifts rather quickly — the optimal p^* value can be approximated stably and with lower-frequency computations, although it can lag behind the optimal p^* value computed with higher frequency and no EMA.

Stability of p^* Values for Deep Neural Networks In a larger scale experiment, we train an MLP-Mixer Tiny model with patch size 16 from the `timm` library [Wightman, 2019] on the ImageNette dataset for 20 epochs. During the training process, we use an adaptive optimizer that computes p^* on the end of every epoch with an EMA of 0.95. We initialize $p^* = p_{\max}$, effectively starting the training with the Muon update, then, we apply $U\Sigma^{1/p_l^*}V^T$ updates to the matrix parameters for every layer l of proxy optimality parameter p_l^*. We record three runs with different random seeds for four different initialization methods, namely `orthogonal`, `kaiming_uniform`, `kaiming_normal`, and `truncated_normal` [Glorot and Bengio, 2010, He et al., 2015, Hu et al., 2020]. Our observations are threefold.

First, our proxy-optimality p^* criterion can be stable and point towards spectral updates as being optimal as demonstrated in Figure 1 (bottom), left panel, where we notice that the first linear layer of the MLP channel parameter for block 5 of the MLP mixer systematically recommends $p^* = p_{\max}$ during the whole training process across all experiments. Secondly, the proxy p^* criterion can also point towards lower p^* values consistently across all runs and initialization methods, as shown on the second layer of the 7th block of the Mixer model (Figure 1, bottom - middle), where $p^* \ll p_{\max}$ consistently across experiments. Finally, while the p^* optimal value can be stable in randomized settings for certain layers, some other layers exhibit run-dependent characteristics, even for the same initialization method, as shown in Figure 1, (bottom - right) where the same Kaiming Normal initialization method can yield different optimal p^* trajectories for the same parameter matrix.

For the rest of the paper, we will use fixed frequency p^* updates with EMA values in $[0.9, 0.99]$ depending on the neural network. Further ablations on the EMA value and update frequency parameters can be found in Appendix I, along p^* visualizations from NanoGPT style runs. Additionnally, for runtime improvements, we provide the theoretical machinery for a top-k SVD based method of the p^* estimation, using the foundational work of Halko et al. [2011], presented in Appendix J.

5 Efficiently Approximating the Fractional Polar Factor

Practically, Schatten-p norm updates are built around what we term the "fractional map direction", defined by the functional: $\mathcal{F}_p : G \longmapsto U\Sigma^{1/p}V^T$, when $G = U\Sigma V^T$.

This map smoothly interpolates between vanilla SGD at $p = 1$ and the Muon orthogonalization step as $p \to \infty$. Computing $\mathcal{F}_p$ exactly via the Singular Value Decomposition (SVD) is prohibitive at scale. Although its asymptotic complexity matches matrix multiplication, SVD is intrinsically sequential, lacks tensor-core acceleration, and severely underutilizes modern GPU and TPU architectures. To resolve this, we require a matrix-multiplication-centric approximation. Below, we detail our primary algorithmic contribution: a highly efficient Taylor approximation method built atop any polar factor estimation method. We also briefly discuss alternative polynomial approaches and concurrent methods to contextualize the accuracy-throughput trade-off.

Taylor Approximation via Polar Factor Estimation. Our approach leverages the polar factor matrix along with the original update matrix to construct an accurate, hardware-efficient approxima-

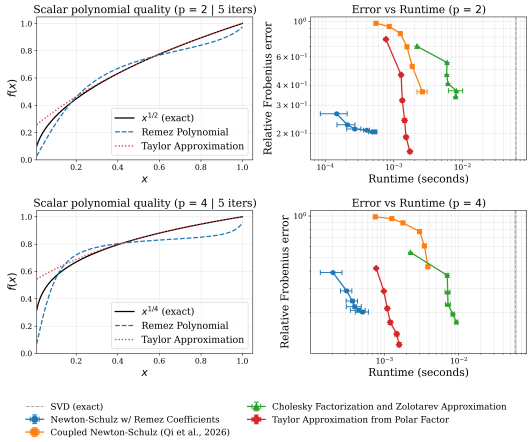


Figure 2: Scalar approximation quality (*left*) and wall-clock error-vs-runtime Pareto frontiers (*right*) for $p \in \{2, 4\}$ on 1024×1024 matrices drawn from a random heavy tailed distribution with condition numbers $[10, 10^3]$ (25 trials, single RTX 4090 GPU). Iterative methods sweep matmul iterative budgets $T \in \{1, \ldots, 6\}$; the dashed vertical line marks the exact SVD baseline.

tion of $\mathcal{F}_p$. This method directly benefits from recent advancements in Newton-Schulz iterations and Gram Newton-Schulz methods [Amsel et al., 2026, Ahn et al., 2025, Zhang et al., 2026, Grishina et al., 2025].

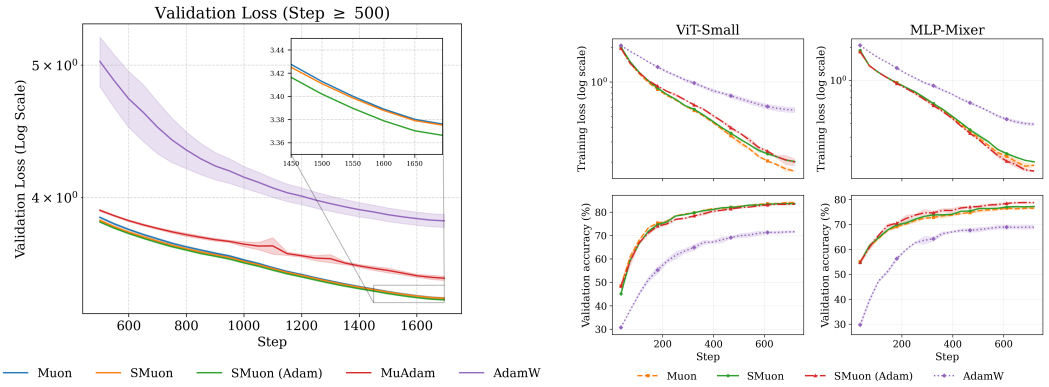
Proposition 3 (Fractional Update Estimate)**.** *For any matrix $G \in \mathbb{R}^{m \times n}$ of singular value decomposition $U\Sigma V^T$ and polar factor $P = UV^T$, we can write:*

$$U\Sigma^{1/p}V^T = \alpha^{1/p} \sum_{i=0}^{\infty} \binom{1/p}{i} \left(\frac{1}{\alpha} GP^T - I\right)^i P \tag{10}$$

for all finite $\alpha \geq \|G\|_2$.

Our proof is given in Appendix C. From any chosen Newton-Schulz implementation, we reliably extract two quantities without resorting to SVD: an upper bound of the spectral norm $\alpha = \|G\|_{S_i}$ (with $i \in \{2, 4\}$ depending on the specific kernel variant [Grishina et al., 2025]), and the polar factor estimate $P \approx UV^T$. Importantly, this method only necessitates matrix multiplications applied after the Newton-Schulz routine, which can be made into a nested fused multiply add sequence via Horner's rule. In practice, the truncation order of the binomial expansion of Proposition 3 can be treated adaptively, chosen as the minimal degree required for the scalar worst-case error on a logarithmic grid within $[\varepsilon, 1]$ to fall below a predefined user tolerance. Experimentally, our method places on the dominant, lower-left Pareto frontier (see Figure 2), providing one to two orders of magnitude higher precision than pure polynomial methods at comparable wall-clock time. Furthermore, its precision degrades gracefully for large values of p as shown in Appendix H.

Concurrent and alternative methods. Bypassing the polar factor, one can target $f(x) = x^{1/p}$ directly with Newton-Schulz-style iterations using Remez-fitted odd-degree polynomials. This achieves the fastest absolute runtime (Figure 2, blue), matching the default Newton-Schulz orthogonalization routine. However, the curvature of $x^{1/p}$ near the origin causes the polynomial approximation to degrade and the L_∞ error degrades sharply for $p \geq 10$ (Appendix G). Qi et al. [2026] couple Newton-Schulz recursions to extract $A^{1/2}$ and $A^{-1/2}$ jointly, achieving high accuracy at the cost of doubled working memory and a restriction to dyadic exponents $p = 2^k$, while requiring k nested passes through the coupled Newton-Schulz routine. Zolotarev-based reductions of the fractional factor to k shifted resolvents $(G^TG + s_jI)^{-1}$ accommodate arbitrary p via batched Cholesky, but forfeit the matmul throughput of tensor-core-optimized polynomials, rendering them impractical at training scale (Figure 2, green).



(a) We plot the validation loss value across training steps on the NanoGPT speedrun across 6 different random seeds. The validation loss is computed on context windows of 262144 tokens, which explains the difference with the numbers from [Du and Su, 2026] which uses half that size.

(b) We plot the training loss and validation accuracy for each optimizer across 20 epochs on the ImageNette dataset, for both a ViT - Small and an MLP-Mixer - Small models.

Figure 3: Training results across language modeling and vision tasks.

About Optimization Quality It is worth noting that perfectly resolving $x^{1/p}$ is not strictly necessary for optimization efficacy. On a standard iteration budget, our Taylor expansion effectively bridges the approximation gap, reaching errors well below the $L_\infty \approx 3.2\times10^{-1}$ error of stock Muon coefficients[2] on $[0.02, 1.0]$. Also, importantly, as demonstrated by Gonon et al. [2026], moderate spectral error in this regime frequently acts as an implicit regularizer, surprisingly improving performance. Thus, we select the Taylor-Horner method which provides the optimal balance of accelerator-friendly throughput and sufficient theoretical precision.

6 Experimental Observations

We evaluate SMuon across three settings selected to stress different parts of the framework: language model pre-training where Muon excels, a mixed-regime vision task, and low-rank fine-tuning where it appears that a purely spectral geometry update is ill-posed for optimization. SMuon will be compared by evaluating the training loss and the validation accuracy, which are common metrics used for optimizer assessment. Each setting reports mean $\pm$ std across seeds, and hyperparameter configurations can be found in Appendix K.

Language model pre-training. We use the Modded-NanoGPT speedrun [Jordan et al., 2024a] on $2\times$ NVIDIA A100s GPUs, training a GPT-2-scale model on FineWeb [Penedo et al., 2024]. To avoid an architecture co-tuned with Muon, we adopt an older checkpoint close to record $\#28$[3], as argued in Du and Su [2026], modified to support our optimizer. All optimizers run with Muon's tuned hyperparameters except MuAdam, for which we sweep the learning rate over six multipliers of Muon's value; AdamW uses the base learning rate found optimal for Muon's auxiliary parameters. SMuon recomputes $p^\star$ every 100 steps for $\sim 3\%$ overhead. We report mean $\pm$ std across 6 seeds (Figure 3a).

The five optimizers separate along the two axes of our framework, LMO geometry and elementwise preconditioning. AdamW lags substantially, as it only treats weight coordinate-wise and ignores the matrix geometry that the transformer architecture rewards. MuAdam recovers part of this gap by composing an Adam-style preconditioner with the Schatten-∞ LMO [Veprikov et al., 2025], but performs worse than Muon. Muon is our reference, as it is the optimizer of the underlying speedrun record. SMuon ties Muon with a marginal numerical advantage but no clear statistical trend; this is not surprising as we find that layerwise p^* values are generally close to $p_{\max}$. On the six seeds run with both optimizers, SMuon (Adam) improves over Muon on every seed ($3.366 \pm 3.1 \cdot 10^{-3}$ vs $3.376 \pm 4.3 \cdot 10^{-3}$), with a mean per-seed reduction of $9.7 \cdot 10^{-3}$; a paired t-test on these matched

[2] As defined in the Keller Jordan implementation.

[3] NanoGPT checkpoint URL

runs rejects the null of equal means at $p < 10^{-3}$ ($t = 6.89$, 5 degrees of freedom). The advantage is largest early in training, when layerwise $p^\star$ on the attention matrices drops sharply within the first $\sim$100 steps, and SMuon (Adam) is alone in supplying both the requested low-p LMO geometry and the elementwise variance control that low-p updates demand. As training progresses and attention $p^\star$ climbs back toward $p_{\max}$ (Figure 4), SMuon (Adam) becomes increasingly Muon-like and the gap narrows monotonically, likely reflecting the natural flattening of the loss curve as training approaches its target. At equal wall-clock time, accounting for the $\sim$3% overhead, SMuon (Adam) reaches $3.370 \pm 5 \cdot 10^{-3}$ versus Muon's $3.376 \pm 4 \cdot 10^{-3}$, retaining the advantage.

Vision. We train a ViT-Small and an MLP-Mixer - Small model with a patch size of 16 from `timm` [Wightman, 2019] on ImageNette [Howard, 2019] for 20 epochs (batch 256, 3 seeds, $\eta \in \{0.1, 0.3, 1, 3, 5, 10, 20\} \cdot 10^{-3}$). With $p^\star$ recomputed once per epoch ($\beta_p = 0.95$, $\sim 0.6\%$ overhead on a single RTX 4090 GPU), we plot the average train loss and validation accuracy across random seeds for the best performing learning rate configuration in Figure 3b. On the MLP-Mixer - Small model, we observe that SMuon matches the performance of Muon in terms of train loss, while SMuon (Adam) exceeds both, with AdamW showing worse performance. This result consolidates our case for adaptive updates with p-dependent preconditioning for variance controlled updates. The ViT-Small model's layers overwhelmingly recommends high p^* values, with 88.9% of p^* values being over 49.9, the adaptive optimizer applies Muon updates when $p^* > p_{\max} - 0.1$. Given the high similarity between optimizer updates, the SMuon variants yield similar validation performance to Muon while allowing for faster training than Adam.

Low-rank fine-tuning. To probe a regime where Muon's full-spectrum spectral geometry is *a priori* suboptimal, we fine-tune Qwen-2.5-0.5B-Instruct [Yang et al., 2024] on GSM8K [Cobbe et al., 2021] with rank-32 LoRA adapters [Hu et al., 2021] over 4-bit quantized weights [Dettmers and Zettlemoyer, 2023] for 3 epochs. The rank-32 structure of the LoRA factors means the polar factor of the momentum matrix has at most 32 nonzero singular values, and orthogonalization across this degenerate spectrum amplifies the directions corresponding to small singular values, which are a priori noisy. The selector should therefore prefer intermediate $p^\star$ on these layers, and indeed we observe an average $p^\star$ across layers and training of 2.75 (SMuon) and 3.10 (SMuon (Adam)), well below $p_{\max} = 50$. Results across 3 seeds are reported in Table 1. Here, we use a p^* computation every epoch with $\beta_p = 0.95$. The picture is relatively clear, with variance controlled optimizers (via orthogonalization or second-order moments), outperforming their counterparts. SGD trails all configurations, while SMuon does slightly better than SGD but still substantially worse than other configurations; justifying the variance controlled adaptive LMO scheme of section 3.2. Finally, we note that Muon performs worse than AdamW and SMuon (Adam) due to its rigid geometry constraints. At last, AdamW outperforms all configurations, closely followed by SMuon (Adam), that permits an adaptive choice of LMO in the correct regime (low p^*) along with variance controlled updates even when p^* is low.

Table 1: Optimizer comparison (mean $\pm$ std over 3 seeds).

	SGD	**AdamW**	**Muon**	**SMuon**	**SMuon (Adam)**
Train loss	$0.736 \pm 7 \cdot 10^{-3}$	$\mathbf{0.337 \pm 2 \cdot 10^{-3}}$	$0.387 \pm 2 \cdot 10^{-3}$	$0.464 \pm 3 \cdot 10^{-3}$	$0.340 \pm 1 \cdot 10^{-3}$
Val acc.	$22.38 \pm 6 \cdot 10^{-3}$	$\mathbf{29.67 \pm 1 \cdot 10^{-3}}$	$28.74 \pm 7 \cdot 10^{-3}$	$28.18 \pm 2 \cdot 10^{-3}$	$29.37 \pm 7 \cdot 10^{-3}$

7 Conclusions, Limitations and Future Work

We introduced a principled framework for adaptive optimization based on Schatten-p norm constrained descent, recovering SGD and Muon as the endpoints $p{=}1$ and $p{\to}\infty$ of a continuous family of update geometries. Using a single-step random feature regression proxy adapted to first-order momentum, we derived a closed-form, layerwise selector $p^\star$ from quantities a standard optimizer already maintains, and paired it with an efficient matmul-centric Taylor approximator for the fractional polar factor $U\Sigma^{1/p}V^T$, built atop of state-of-the-art Newton–Schulz kernels. The resulting optimizer, SMuon (Adam), outperforms a heavily-tuned Muon baseline on Modded-NanoGPT and shows competitiveness on vision workloads at sub-percent overhead. Also, in voluntarily adversarial settings to Muon, we show that our proxy imposes Adam like updates thus staying competitive with AdamW on low-rank fine-tuning and significantly outperforming Muon in terms of training loss.

Three directions follow naturally from our framework as future optimizations. Firstly, the random feature regression model is, by design, the simplest second-order surrogate that captures the gradient-activation interaction; richer surrogates that account for downstream Jacobians (e.g. for attention projections or LoRA factors) could yield sharper selectors at the cost of locality [Li, 2017], and we view the trade-off between proxy fidelity and per-layer plug-and-play deployment as an open empirical question. Secondly, our derivation computes the descent maximization criterion on a single optimizer step. Extending the criterion to multi-step horizons and accounting for the trajectory of $p^\star$ under the induced dynamics with more granularity than our E.M.A mechanism allows would tighten the connection between the proxy and the long-run training loss (see Appendix L). Lastly, and perhaps most importantly, revealing repeatable and robust p^* behaviors during training, and showing that these p^* value dynamics can be repeated or predicted on larger models [Kaplan et al., 2020] could allow practitioners to choose predicted-optimal update geometries p^* on larger networks layers, without paying the cost of p^* approximation.

Additional Ablations

We provide additional experiments in the appendices to study the effect of our geometry adaptation framework. Appendix F compares our p^* selection proxy to a grid search among fixed p candidate values, and shows that our selection process allows similar or better performance to the best p value from the gridsearch out of the box. Appendix E studies the scaling behavior of the optimization method across batch size, model width and model depth, showing that our method achieves the best performance across the test configurations and that changing the batch size, model width and model depth result in very different p^* trends thus confirming the need for a layer-wise selection model.

Acknowledgements

The authors want to thank Louis Béthune and Thibaut Boissin for their insightful comments throughout the writing of this paper. Our work has benefited from the AI Cluster ANITI and the research program DEEL[4]. ANITI is funded by the France 2030 program under the Grant agreement n°ANR-23-IACL-0002. DEEL is an integrative program of the AI Cluster ANITI, designed and operated jointly with IRT Saint Exupéry, with the financial support from its industrial and academic partners and the France 2030 program under the Grant agreement n°ANR-10-AIRT-01. This project was provided with computing and storage resources by GENCI at IDRIS thanks to the grant 2025-AD011016850 on the supercomputer Jean Zay's A100 and H100 partition.

The authors declare no competing interests.

References

Kwangjun Ahn, Byron Xu, Natalie Abreu, Ying Fan, Gagik Magakyan, Pratyusha Sharma, Zheng Zhan, and John Langford. Dion: Distributed orthonormalized updates. *arXiv preprint arXiv:2504.05295*, 2025.

Noah Amsel, David Persson, Christopher Musco, and Robert M. Gower. The polar express: Optimal matrix sign methods and their application to the muon algorithm. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=yRtgZ1K8hO`.

Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology. In *OPT 2024: Optimization for Machine Learning*, 2024.

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

Richard P Brent. *Algorithms for minimization without derivatives*. Courier Corporation, 2013.

Franz Louis Cesista. Steepest Descent under Schatten-p Norms, February 2025. URL `https://leloykun.github.io/ponder/steepest-descent-schatten-p/`.

[4]https://www.irt-saintexupery.com/deel/

Da Chang, Qiankun Shi, Lvgang Zhang, Yu Li, Ruijie Zhang, Yao Lu, Yongxiang Liu, and Ganzhao Yuan. Muoneq: Balancing before orthogonalization with lightweight equilibration. *arXiv preprint arXiv:2603.28254*, 2026.

Youngmin Cho and Lawrence Saul. Kernel methods for deep learning. *Advances in neural information processing systems*, 22, 2009.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. *Advances in neural information processing systems*, 29, 2016.

Damek Davis and Dmitriy Drusvyatskiy. When do spectral gradient updates help in deep learning? *arXiv e-prints*, pages arXiv–2512, 2025.

Blaise Delattre, Quentin Barthélemy, Alexandre Araujo, and Alexandre Allauzen. Efficient bound of lipschitz constant for convolutional layers by gram iteration. In *International conference on machine learning*, pages 7513–7532. PMLR, 2023.

Tim Dettmers and Luke Zettlemoyer. The case for 4-bit precision: k-bit inference scaling laws. In *International Conference on Machine Learning*, pages 7750–7774. PMLR, 2023.

Sara Dragutinović and Rajesh Ranganath. To use or not to use muon: How simplicity bias in optimizers matters. In *Workshop on Scientific Methods for Understanding Deep Learning*, 2026. URL https://openreview.net/forum?id=GsZtgQf3IM.

Zhehang Du and Weijie Su. The newton-muon optimizer, 2026. URL https://arxiv.org/abs/2604.01472.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/glorot10a.html.

Antoine Gonon, Andreea-Alexandra Muşat, and Nicolas Boumal. Insights on muon from simple quadratics. *arXiv preprint arXiv:2602.11948*, 2026.

Ekaterina Grishina, Matvey Smirnov, and Maxim Rakhuba. Accelerating newton-schulz iteration for orthogonalization via chebyshev-type polynomials. *arXiv preprint arXiv:2506.10935*, 2025.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pages 1842–1850. PMLR, 2018.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2): 217–288, 2011.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

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

Jeremy Howard. Imagenette: A smaller subset of 10 easily classified classes from imagenet, March 2019. URL https://github.com/fastai/imagenette.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

Wei Hu, Lechao Xiao, and Jeffrey Pennington. Provable benefit of orthogonal initialization in optimizing deep linear networks. In *International Conference on Learning Representations*, 2020.

Keller Jordan, Jeremy Bernstein, Brendan Rappazzo, @fernbear.bsky.social, Boza Vlado, You Jiacheng, Franz Cesista, Braden Koszarsky, and @Grad62304977. modded-nanogpt: Speedrunning the nanogpt baseline, 2024a. URL https://github.com/KellerJordan/modded-nanogpt.

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

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2, 1989.

Xi-Lin Li. Preconditioned stochastic gradient descent. *IEEE transactions on neural networks and learning systems*, 29(5):1454–1466, 2017.

Zichong Li, Liming Liu, Chen Liang, Weizhu Chen, and Tuo Zhao. Normuon: Making muon more efficient and scalable. *arXiv preprint arXiv:2510.05491*, 2025.

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

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024.

Thomas Pethick, Wanyun Xie, Kimon Antonakopoulos, Zhenyu Zhu, Antonio Silveti-Falls, and Volkan Cevher. Training deep learning models with norm-constrained LMOs. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=2Oqm2IzTy9.

Gabriel Peyré. Muon dynamics as a spectral wasserstein flow, 2026. URL https://arxiv.org/abs/2604.04891.

Omead Pooladzandi and Xi-Lin Li. Curvature-informed sgd via general purpose lie-group preconditioners, 2024. URL https://arxiv.org/abs/2402.04553.

Xianbiao Qi, Marco Chen, Jiaquan Ye, Yelin He, and Rong Xiao. Delving into muon and beyond: Deep analysis and extensions. *arXiv preprint arXiv:2602.04669*, 2026.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. *Advances in neural information processing systems*, 21, 2008.

Frank Rosenblatt et al. *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*, volume 55. Spartan books Washington, DC, 1962.

Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. *Advances in neural information processing systems*, 30, 2017.

R Schatten. A theory of cross spaces. *Ann. of Math. Studies*, 26, 1950.

Chongjie Si, Debing Zhang, and Wei Shen. Adamuon: Adaptive muon optimizer. *arXiv preprint arXiv:2507.11005*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Andrey Veprikov, Arman Bolatov, Samuel Horváth, Aleksandr Beznosikov, Martin Takáč, and Slavomir Hanzely. Preconditioned norms: A unified framework for steepest descent, quasi-newton and adaptive methods, 2025. URL https://arxiv.org/abs/2510.10777.

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 *The Thirteenth International Conference on Learning Representations*, 2024.

Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Jack Zhang, Noah Amsel, Berlin Chen, and Tri Dao. Gram newton-schulz, 2026. URL https://dao-ailab.github.io/blog/2026/gram-newton-schulz/.

A Proofs

A.1 The Schatten-p Norm Constrained Update Direction

Proposition 4. *Let $G \in \mathbb{R}^{m\times n}$ have SVD $G = U\Sigma V^T$, with $\sigma_1 \geq \cdots \geq \sigma_r > 0$ the nonzero singular values. For any $p \in [1, \infty)$, the solution to the constrained problem*

$$\min_{\delta W \in \mathbb{R}^{m\times n}} \langle G,\, \delta W\rangle_F \quad \textit{s.t.} \quad \|\delta W\|_{S_{p+1}} \leq 1 \tag{$\mathcal{P}$}$$

satisfies $\delta W^\star \propto U\Sigma^{1/p}V^T$. In the limit $p \to \infty$, the constraint becomes a spectral norm constraint and the solution recovers $\delta W^\star \propto \mathrm{PolarFactor}(G) = UV^T$.

Proof. **Step 1: Decoupling via unitary invariance.**

By von Neumann's trace inequality, for any $A, B \in \mathbb{R}^{m\times n}$:

$$\langle A, B\rangle_F \geq -\sum_{i=1}^{r} \sigma_i(A)\,\sigma_i(B), \tag{11}$$

with equality if and only if A and B share the same left and right singular vectors, with $B = -U_A\,\mathrm{diag}(\sigma_i(B))V_A^T$. Applying this to $A = G$ and $B = \delta W$, the objective $\langle G, \delta W\rangle_F$ is minimised, for any fixed singular values $d_i = \sigma_i(\delta W)$, by taking $\delta W = -UDV^T$ with $D = \mathrm{diag}(d_i)$, yielding:

$$\langle G, \delta W\rangle_F = -\sum_{i=1}^{r} \sigma_i\, d_i. \tag{12}$$

Since $\|\delta W\|_{S_{p+1}}^{p+1} = \sum_i d_i^{p+1}$ is invariant to the choice of singular vectors, the problem $(\mathcal{P})$ reduces without loss of generality to:

$$\max_{d_i \geq 0} \sum_{i=1}^{r} \sigma_i\, d_i \quad \text{s.t.} \quad \sum_{i=1}^{r} d_i^{p+1} \leq 1. \tag{$\mathcal{P}'$}$$

The constraint is active at any optimum since all $\sigma_i > 0$, so we work with $\sum_i d_i^{p+1} = 1$.

Step 2: Solving the scalar problem via Hölder's inequality.

Apply Hölder's inequality with conjugate exponents $a = \frac{p+1}{p}$ and $b = p + 1$, satisfying $\frac{1}{a} + \frac{1}{b} = \frac{p}{p+1} + \frac{1}{p+1} = 1$, to the sequences $x_i = \sigma_i$ and $y_i = d_i$:

$$\sum_{i=1}^{r} \sigma_i\, d_i \leq \left(\sum_{i=1}^{r} \sigma_i^{(p+1)/p}\right)^{p/(p+1)} \left(\sum_{i=1}^{r} d_i^{p+1}\right)^{1/(p+1)}. \tag{13}$$

Since $\sum_i d_i^{p+1} = 1$, this simplifies to:

$$\sum_{i=1}^{r} \sigma_i\, d_i \leq \left(\sum_{i=1}^{r} \sigma_i^{(p+1)/p}\right)^{p/(p+1)}. \tag{14}$$

Equality in Hölder's inequality holds if and only if $x_i^a \propto y_i^b$, i.e. $\sigma_i^{(p+1)/p} \propto d_i^{p+1}$, which gives:

$$d_i \propto \sigma_i^{1/p}. \tag{15}$$

Imposing the normalisation $\sum_i d_i^{p+1} = 1$ with $d_i = c\,\sigma_i^{1/p}$ and solving for c:

$$c^{p+1}\sum_{i=1}^{r} \sigma_i^{(p+1)/p} = 1 \implies c = \left(\sum_{i=1}^{r} \sigma_i^{(p+1)/p}\right)^{-1/(p+1)}, \tag{16}$$

yielding the unique maximiser of $(\mathcal{P}')$:

$$d_i^\star = \frac{\sigma_i^{1/p}}{\left(\sum_{j=1}^{r} \sigma_j^{(p+1)/p}\right)^{1/(p+1)}}. \tag{17}$$

Step 3: Reconstruction and absorption into the learning rate.

The solution to $(\mathcal{P})$ is:

$$\delta W^\star = -UD^\star V^T = -\frac{U\Sigma^{1/p}V^T}{\left(\sum_j \sigma_j^{(p+1)/p}\right)^{1/(p+1)}}, \tag{18}$$

where the denominator is a strictly positive scalar depending only on G and p. Absorbing it into the learning rate $\eta > 0$, the descent direction is:

$$\delta W^\star \;\propto\; -U\Sigma^{1/p}V^T. \tag{19}$$

Step 4: Special cases and the limit $p \to \infty$.

- $p = 1$: $d_i^\star \propto \sigma_i$, so $\delta W^\star \propto U\Sigma V^T = G$, recovering the **SGD** update direction.
- $p \to \infty$: Since $\lim_{p\to\infty} \|\cdot\|_{S_{p+1}} = \|\cdot\|_{S_\infty}$ [Delattre et al., 2023], the constraint in $(\mathcal{P})$ converges to the spectral norm constraint $\|\delta W\|_{S_\infty} \leq 1$. Correspondingly, $\sigma_i^{1/p} \to 1$ for all i, so $\delta W^\star \propto UV^T = \mathrm{PolarFactor}(G)$, recovering the **Muon** update direction.

□

A.2 Closed Form Optimal p Value From the Random Feature Regression Model

We prove the closed-form expression for the optimal Schatten exponent $p^\star$ given in Proposition 2 of the main text, restricted to the unpreconditioned case $D = \mathbf{1}$. The general case with $D \neq \mathbf{1}$ is treated in Appendix D. We additionally show that the resulting one-dimensional optimization problem in p is well-suited to bounded scalar line search.

A.2.1 Setting and Notation

We work in the random feature regression setting of Section 2.2, with loss

$$\mathcal{L}(W) = \frac{1}{2k}\|WA - Y\|_F^2, \tag{20}$$

where $W \in \mathbb{R}^{m\times n}$ is a layer's weight matrix, $A \in \mathbb{R}^{n\times k}$ is the post-activation of the previous layer at step t, and $Y \in \mathbb{R}^{m\times k}$ is the target. The gradient is $G = \nabla\mathcal{L}(W) = \frac{1}{k}(WA - Y)A^T$, and the momentum buffer follows the standard EMA recursion $M \leftarrow \beta M + (1-\beta)G$.

Let $M = U_M\Sigma_M V_M^T$ be the (thin) singular value decomposition of M, with $\Sigma_M = \mathrm{diag}(\sigma_{M,1}, \ldots, \sigma_{M,k})$ and $k = \min(m, n_{\text{out}})$. The Schatten-$p$ momentum descent update considered in this section is

$$\delta W = -\eta\, U_M\Sigma_M^{1/p}V_M^T, \qquad p \in [1, \infty], \tag{21}$$

which recovers SGD with momentum at $p = 1$ and Muon as $p \to \infty$.

A.2.2 Exact Loss Decomposition

The squared loss (20) is a convex quadratic in W, and the loss change from a perturbation δW admits the exact decomposition

$$\mathcal{L}(W + \delta W) - \mathcal{L}(W) = \langle G, \delta W\rangle_F + \frac{1}{2k}\|\delta W \cdot A\|_F^2. \tag{22}$$

This is an equality, not an approximation: the cross-term $\langle G, \delta W\rangle_F$ captures the first-order descent and the quadratic term $\frac{1}{2k}\|\delta WA\|_F^2$ captures the curvature of $\mathcal{L}$ along the update direction, with no higher-order remainder.

A.2.3 Simplification of the First-Order Term

Substituting (21) into the first-order term and expanding the inner product in the singular basis,

$$\begin{aligned}\langle G, \delta W\rangle_F &= -\eta \,\langle G, U_M \Sigma_M^{1/p} V_M^T\rangle_F \\ &= -\eta \,\mathrm{Tr}\big(\Sigma_M^{1/p}\, V_M^T G^T U_M\big) \\ &= -\eta \,\mathrm{Tr}\big(\Sigma_M^{1/p}\, C^T\big) \\ &= -\eta \sum_{i=1}^{k} \sigma_{M,i}^{1/p}\, C_{i,i}, \end{aligned} \tag{23}$$

where we have introduced the *alignment matrix*

$$C := U_M^T G V_M \in \mathbb{R}^{k\times k}, \tag{24}$$

and $C_{i,i}$ are its diagonal entries. The off-diagonal entries of C do not appear in (23) because the trace selects only the diagonal of the product $\Sigma_M^{1/p} C^T$.

Interpretation. $C_{i,i}$ measures the alignment between the current gradient G and the i-th singular component of the momentum-accumulated gradient M. When G is well aligned with M (e.g., during stable phases), the leading $C_{i,i}$ are large and positive; when G deviates from the momentum direction, the $C_{i,i}$ shrink or change sign. The sign of $C_{i,i}$ may flip across the spectrum, reflecting partial misalignment between gradient and momentum.

A.2.4 Simplification of the Curvature Term

Substituting (21) into the curvature term and using the orthonormality of U_M,

$$\begin{aligned}\|\delta W \cdot A\|_F^2 &= \eta^2 \,\|U_M \Sigma_M^{1/p} V_M^T A\|_F^2 \\ &= \eta^2 \,\mathrm{Tr}\big(A^T V_M \Sigma_M^{2/p} V_M^T A\big) \\ &= \eta^2 \,\mathrm{Tr}\big(\Sigma_M^{2/p}\, V_M^T A A^T V_M\big) \\ &= \eta^2 \sum_{i=1}^{k} \sigma_{M,i}^{2/p}\, B_{i,i}, \end{aligned} \tag{25}$$

where

$$B_{i,i} := (V_M^T A A^T V_M)_{i,i} = \|A^T V_M e_i\|^2. \tag{26}$$

Each $B_{i,i} \geq 0$ measures the energy of the i-th right-singular direction of M when projected onto the activation A. The orthonormality of U_M is what eliminates cross-singular-value couplings: $\|U_M X\|_F^2 = \|X\|_F^2$ for any X, so the U_M factors cancel cleanly. This is a special property of $D = \mathbf{1}$ and is the structural reason why the closed-form $p^\star$ exists in this case.

A.2.5 Closed-Form Optimal Step Size

Combining (22), (23), and (25), the loss decrease is

$$\mathcal{L}(W) - \mathcal{L}(W + \delta W) = \eta\, N(p) - \frac{\eta^2}{2k}\, D(p), \tag{27}$$

where we have defined

$$N(p) := \sum_{i=1}^{k} \sigma_{M,i}^{1/p}\, C_{i,i}, \qquad D(p) := \sum_{i=1}^{k} \sigma_{M,i}^{2/p}\, B_{i,i}. \tag{28}$$

The right-hand side of (27) is concave quadratic in η, with maximum at the co-optimal step size

$$\eta^\star(p) = \frac{k\, N(p)}{D(p)}, \tag{29}$$

attained whenever $N(p) > 0$ (otherwise the maximum is at $\eta = 0$, i.e., the update direction is not a descent direction and the quadratic model recommends not moving). Substituting $\eta^\star(p)$ back into (27) yields the maximum loss decrease at exponent p:

$$\big[\mathcal{L}(W) - \mathcal{L}(W + \delta W)\big]_{\eta=\eta^\star(p)} = \frac{k}{2}\,\frac{N(p)^2}{D(p)}. \tag{30}$$

Remark In practice, all $\eta^*(p)$ values we computed on deep learning tasks turned out to be positive *across all vision experiments where they where monitored*. However, in the case where $\eta^*(p) \leq 0$, a fallback heuristic could be to leave the p^* value unchanged for that particular step.

A.2.6 Closed-Form Optimal p

The factor $k/2$ in (30) is independent of p, so the value of p that maximizes the loss decrease under co-optimal step size is

$$\boxed{p^\star = \operatorname{argmax}_{p\in[1,\infty]} \frac{N(p)^2}{D(p)} = \operatorname{argmax}_{p\in[1,\infty]} \frac{\left(\sum_i \sigma_{M,i}^{1/p} C_{i,i}\right)^2}{\sum_i \sigma_{M,i}^{2/p} B_{i,i}}} \tag{31}$$

which is the expression stated in Proposition 2. Note that the optimal step size $\eta^\star(p)$ is itself p-dependent, but its value does not enter the criterion for $p^\star$: the η-degree-of-freedom has been integrated out by the closed-form maximization in (29), leaving $p^\star$ as a function of the spectrum and the alignment/curvature coefficients only. The criterion can equivalently be read as a *correlation-to-curvature ratio*: $N(p)$ measures the p-weighted correlation between gradient and momentum-singular structure, $D(p)$ measures the corresponding curvature, and $p^\star$ chooses the spectral geometry that maximizes the squared ratio. □

A.2.7 Suitability of Bounded Scalar Line Search

Equation (31) is a one-dimensional optimization over $p \in [1, \infty]$. We argue that it is well-suited to a bounded scalar line search algorithm such as Brent's method [Brent, 2013], with three structural properties supporting this claim.

Bounded effective domain. Although the criterion is defined formally on $[1, \infty]$, the dependence of $N(p)$ and $D(p)$ on p is through $\sigma_{M,i}^{1/p}$ and $\sigma_{M,i}^{2/p}$. As $p \to \infty$, both $\sigma_{M,i}^{1/p} \to 1$ and $\sigma_{M,i}^{2/p} \to 1$, so the criterion saturates to a finite limit, assuming $\sigma_{M,i} > 0$:

$$\lim_{p\to\infty} \frac{N(p)^2}{D(p)} = \frac{\left(\sum_i C_{i,i}\right)^2}{\sum_i B_{i,i}}. \tag{32}$$

The criterion becomes essentially flat once p is large enough that all $\sigma_{M,i}^{1/p}$ are close to unity. We therefore restrict the search to a bounded interval $[p_{\min}, p_{\max}]$ chosen above this saturation regime; in our experiments $p_{\min} = 1.02$ (slightly above 1 to avoid the degenerate $\sigma_{M,i}^1$ summation) and $p_{\max} = 50$ are sufficient.

Smoothness. Both $N(p)$ and $D(p)$ are infinitely differentiable functions of p on $(1, \infty)$, since each summand $\sigma_{M,i}^{q/p}$ is smooth in p for $\sigma_{M,i} > 0$ and $q \in \{1, 2\}$. The criterion $J(p) := N(p)^2/D(p)$ is therefore C^∞ wherever $D(p) > 0$, which holds whenever there exists at least one index i with both $\sigma_{M,i} > 0$ and $B_{i,i} > 0$ — a generic condition for non-degenerate momentum and activation matrices.

Mild non-convexity, well-behaved in practice. $J(p)$ is not unimodal in full generality: the squared correlation $N(p)^2$ can have local minima where $C_{i,i}$ entries cancel, producing transient zeros of $N(p)$ separating regions of opposite sign. In stable training phases, however, the dominant $C_{i,i}$ entries are positive and consistently signed the gradient and momentum agree on their leading components and we observe smooth, single-maximum landscapes throughout these phases. The rare cases in which $J(p)$ becomes non-unimodal coincide with phase transitions where the criterion's underlying assumptions are themselves momentarily strained; the temporal EMA on the alignment and curvature coefficients (Section 4) absorbs these transients without affecting the surrounding $p^\star$ trajectory. Brent's method, which combines bracketing with golden-section search and parabolic interpolation, converges rapidly on the smooth landscapes that dominate training and degrades gracefully, returning a local maximum within the bracket, on the rare non-unimodal steps. We have not observed pathological behavior of the line search in any of our experiments.

Cost per evaluation. Once the SVD of M and the alignment/curvature coefficients $\{\sigma_{M,i}, C_{i,i}, B_{i,i}\}_{i=1}^{k}$ are precomputed, each evaluation of $J(p)$ costs $O(k)$ floating-point operations: two k-fold sums and a division. The full line search therefore costs $O(k \cdot n_{\text{eval}})$ on top of the one-time SVD cost, which is negligible compared to the SVD itself. This is the regime in which the bounded scalar minimization is essentially free, and is one of the reasons why the unpreconditioned criterion of (31) is computationally attractive — a property that is partially lost when general second-order moments are introduced (see Appendix D).

B Considering the Scale of Updates

Throughout this study, we choose the optimal update geometry according to a random feature regression surrogate that assumes optimal step sizes. Importantly, this step size can be computed in practice. However, we found this factor to be variable in the $[10^{-2}, 10^{4}]$ range across model layers. Importantly, we found that using optimal step size multipliers according to the random feature regression model could degrade training as some η_t^* values could become higher than η_{t-1}^* after a $p_{t-1}^* \to p_t^*$ update, thus creating loss spikes. Furthermore, scaling each update with its optimal step size according to its LMO formulation requires the computation of the Schatten-$(p+1)$ norm of the update, which is costly in practice. Finally, as previously shown in Qi et al. [2026] for fixed p variants, second order preconditioning seems to importantly mitigate this phenomenon, thus explaining the poor performance of SMuon with no second order preconditioning in some scenarios.

C Proof of Taylor Approximation of $U\Sigma^{1/p}V^T$ in Prop. 3

Proposition 5 (Fractional Update Estimate)**.** *For any matrix $G \in \mathbb{R}^{m\times n}$ of singular value decomposition $U\Sigma V^T$ and polar factor $P = UV^T$, we can write:*

$$U\Sigma^{1/p}V^T = \alpha^{1/p} \sum_{i=0}^{\infty} \binom{1/p}{i} \left(\frac{1}{\alpha} GP^T - I\right)^i P \tag{33}$$

for all finite $\alpha \geq \|G\|_2$.

Proof. Let $G = U\Sigma V^T$ be the singular value decomposition of G. The Schatten-$(p+1)$ fractional map is defined by

$$\mathcal{F}_p(G) = U\Sigma^{1/p}V^T.$$

Let $P = UV^T$ denote the polar factor of G, and let $\alpha > 0$ be a normalization factor (usually $\alpha \geq \|G\|_2$ with $\|\cdot\|_2$ the spectral norm). Define

$$Z = \frac{1}{\alpha} U\Sigma U^T.$$

Note that the eigenvalues of Z are between 0 and 1, due to the normalization factor α. Using $G = U\Sigma V^T$ and $P^T = VU^T$, we have

$$Z = \frac{1}{\alpha} U\Sigma U^T = \frac{1}{\alpha} GP^T.$$

Now observe that, since Z is a positive semi-definite matrix,

$$Z^{1/p} = \left(\frac{1}{\alpha} U\Sigma U^T\right)^{1/p} = \frac{1}{\alpha^{1/p}} U\Sigma^{1/p}U^T.$$

Therefore,

$$\alpha^{1/p} Z^{1/p} P = \alpha^{1/p} \left(\frac{1}{\alpha^{1/p}} U\Sigma^{1/p}U^T\right) UV^T = U\Sigma^{1/p}V^T.$$

Hence,

$$\mathcal{F}_p(G) = \alpha^{1/p} Z^{1/p} P.$$

Next, with I the identity matrix and E the complementary matrix to Z, write

$$Z = I + E, \qquad E = Z - I.$$

Then, using the binomial Taylor expansion,

$$Z^{1/p} = (I + E)^{1/p} = \sum_{i=0}^{\infty} \binom{1/p}{i} E^i.$$

Since

$$E = Z - I = \frac{1}{\alpha} G P^T - I,$$

we obtain

$$Z^{1/p} = \sum_{i=0}^{\infty} \binom{1/p}{i} \left(\frac{1}{\alpha} G P^T - I\right)^i.$$

Finally we obtain the result in Prop. 3,

$$\mathcal{F}_p(G) = \alpha^{1/p} \left[\sum_{i=0}^{\infty} \binom{1/p}{i} \left(\frac{1}{\alpha} G P^T - I\right)^i\right] P.$$

In practice, the infinite series is truncated at degree K, yielding the approximation

$$\widehat{\mathcal{F}}_p(G) = \alpha^{1/p} \left[\sum_{i=0}^{K} \binom{1/p}{i} \left(\frac{1}{\alpha} G P^T - I\right)^i\right] P.$$

Moreover, computing exactly the polar factor P is prohibitive. The Newton-Schulz approximation is used to efficiently estimate P. □

D Second-Order Moments in the Schatten-p Framework

The main text develops the optimal Schatten exponent $p^\star$ criterion for $D = \mathbf{1}$ (SGD, Muon, and Schatten-p momentum descent). This appendix shows how an elementwise second-order preconditioner D can be fitted into the framework, characterizes the resulting modification to the $p^\star$ criterion, and discusses the interpolation between Euclidean and spectral families that arises naturally from this construction.

D.1 Plugging Adam-Style Coordinate Rescaling into the Update Rule

Recall the unified update rule from Veprikov et al. [2025]:

$$\delta W = -\eta\, D^{\circ -1} \odot \mathrm{LMO}_{(p+1)}\big(D^{\circ -1} \odot M\big), \tag{34}$$

where $D \in \mathbb{R}^{m \times n}_{>0}$ encodes second-order moment information. For Adam-style preconditioning, $D^{\circ -1}$ is constructed from a bias-corrected exponential moving average of squared gradients $\hat{V}_t = (1 - \beta_2^t)^{-1} \sum_{s \le t} \beta_2^{t-s} (1 - \beta_2)\, G_s \odot G_s$, with the multiplier scaling

$$D^{\circ -1} = (\hat{V} + \varepsilon)^{\circ -\alpha}, \tag{35}$$

where $\alpha = 1/2$ recovers Adam-style RMS rescaling, $\alpha = 1/4$ recovers MuAdam [Veprikov et al., 2025], and $\alpha = 1/(2(p+1))$ defines a *p-dependent* family that smoothly transitions from MuAdam at $p = 1$ to Muon at $p \to \infty$ (see Section D.3).

The recipe is mechanical: pre-multiply G and M elementwise by $D^{\circ -1}$ before invoking the LMO, and post-multiply the LMO output again by $D^{\circ -1}$ before applying the update. This is what Veprikov et al. [2025] call the *symmetric D-norm preconditioned LMO* for the Schatten-$(p+1)$ base norm.

D.2 Modification of the Optimal p Proxy

We now derive how the $p^\star$ criterion of Proposition 2 changes when $D \neq \mathbf{1}$. Write $\tilde{G} := D^{\circ -1} \odot G$, $\tilde{M} := D^{\circ -1} \odot M$, and let $\tilde{M} = \tilde{U}_M \tilde{\Sigma}_M \tilde{V}_M^T$ be its SVD. The actual physical update is

$$\delta W = -\eta\, D^{\circ -1} \odot \tilde{Y}_p, \qquad \tilde{Y}_p := \tilde{U}_M \tilde{\Sigma}_M^{1/p} \tilde{V}_M^T. \tag{36}$$

Alignment term. Using the identity $\langle G, D^{\circ -1} \odot Y \rangle_F = \langle \tilde{G}, Y \rangle_F$, the first-order term is

$$\langle G, \delta W \rangle_F = -\eta \sum_i \tilde{\sigma}_{M,i}^{1/p} \tilde{C}_{i,i}, \qquad \tilde{C}_{i,i} := (\tilde{U}_M^T \tilde{G} \tilde{V}_M)_{i,i}. \tag{37}$$

This expression is exact for arbitrary D and is independent of p except through the $\tilde{\sigma}_{M,i}^{1/p}$ factor.

Curvature term. The curvature is

$$\frac{1}{2n} \|\delta W \cdot A\|_F^2 = \frac{\eta^2}{2n} \left\| (D^{\circ -1} \odot \tilde{Y}_p) A \right\|_F^2. \tag{38}$$

Crucially, the Hadamard product does not commute with matrix multiplication: $(D^{\circ -1} \odot \tilde{Y}_p)A \neq D^{\circ -1} \odot (\tilde{Y}_p A)$. Expanding in the singular basis,

$$\left\| (D^{\circ -1} \odot \tilde{Y}_p) A \right\|_F^2 = \sum_{r,s} \tilde{\sigma}_{M,r}^{1/p} \, \tilde{\sigma}_{M,s}^{1/p} \, K_{rs}(D), \tag{39}$$

where the coupling tensor is

$$K_{rs}(D) := \sum_{i,j,k} (D^{\circ -1})_{ij} (D^{\circ -1})_{ik} \, \tilde{U}_{M,ir} \tilde{U}_{M,is} \, \tilde{V}_{M,jr} \tilde{V}_{M,ks} (AA^T)_{jk}. \tag{40}$$

The diagonal entries K_{rr} recover the $D = \mathbf{1}$ result $\tilde{B}_{i,i} := \|A^T \tilde{V}_M e_i\|^2$ when $D \propto \mathbf{1}$. The off-diagonal entries K_{rs} for $r \neq s$ vanish identically in that case (by orthonormality of $\tilde{U}_M$) but are nonzero in general — they are the *cross-singular-value couplings* between distinct components of $\tilde{Y}_p$ induced by the elementwise preconditioner.

Computational consequence. For $D = \mathbf{1}$, the curvature is a sum $\sum_i \tilde{\sigma}_{M,i}^{2/p} \tilde{B}_{i,i}$, allowing the bounded scalar minimization over p to be evaluated from precomputed quantities $\{\tilde{\sigma}_{M,i}, \tilde{C}_{i,i}, \tilde{B}_{i,i}\}_i$ at $O(k)$ cost per call. For $D \neq \mathbf{1}$, the curvature (38) requires reforming $\tilde{Y}_p$ for each candidate p — specifically:

$$D_{\text{val}}(p) = \left\| (D^{\circ -1} \odot \tilde{U}_M \tilde{\Sigma}_M^{1/p} \tilde{V}_M^T) A \right\|_F^2, \tag{41}$$

which costs one $m \times n$ matrix product to form $\tilde{Y}_p$, one Hadamard product, one $m \times s$ matrix product with A, and one Frobenius norm. Compared to the $D = \mathbf{1}$ closed form, this is an order-of-magnitude increase in per-call cost during the bounded scalar search, but remains tractable: the bounded minimizer typically issues ~ 20 objective evaluations, costing roughly the same as a single forward pass on the layer being optimized.

Proposition 6 (Layerwise optimal Schatten exponent under preconditioning)**.** *Let $\tilde{M} = D^{\circ -1} \odot M$ have SVD $\tilde{U}_M \tilde{\Sigma}_M \tilde{V}_M^T$, and let $\tilde{C}_{i,i}$ be defined as in* (37)*. The value of p that maximizes the exact one-step descent guarantee on the random feature regression loss under co-optimal step size and the update rule* (36) *is*

$$p^\star(D) = \operatorname{argmax}_{p \in [1,\infty]} \; \frac{\left(\sum_i \tilde{\sigma}_{M,i}^{1/p} \, \tilde{C}_{i,i} \right)^2}{\left\| (D^{\circ -1} \odot \tilde{U}_M \tilde{\Sigma}_M^{1/p} \tilde{V}_M^T) A \right\|_F^2}. \tag{42}$$

For $D \propto \mathbf{1}$, this reduces to Eq. (8) *of the main text, with the curvature decomposing into a sum over singular values. For general D, the curvature is irreducibly a quadratic form in $\tilde{\sigma}_{M,r}^{1/p} \tilde{\sigma}_{M,s}^{1/p}$, and the bounded scalar search requires reforming $\tilde{Y}_p$ at each candidate p.*

In practice however, we consider that the optimality of the p^* proxy value is sufficient by only taking first order moments into account, and we leave the choice of p-dependent second order moment as a consolidating heuristic for the optimization. Importantly, this heuristic simplification is not invalid, it simply assumes that the first-order moment random feature regression setting is *sufficient* to model the behavior of the adaptive LMO in *preconditioned* settings.

Moment configuration	$p = 1$ **update**	$p \to \infty$ **update**
$D = \mathbf{1}$ (no moment)	SGD	Muon
$D^{\circ -1} = \hat{V}^{\circ -1/4}$ (fixed)	Adam	MuAdam
$D^{\circ -1} = \hat{V}^{\circ -1/(2(p+1))}$ (p-dep.)	Adam	Muon

Table 2: Optimizer family recovered by the unified update rule (34) for various choices of moment D and Schatten exponent p. Fixed-α moments interpolate the *magnitude* of the update toward Adam-style at $p = 1$ but preserve the Muon-like spectral structure at $p \to \infty$ *with persistent variance rescaling*. p-dependent moments additionally have $D \to \mathbf{1}$ as $p \to \infty$, recovering pure Muon at the spectral endpoint and pure MuAdam at the Euclidean endpoint.

D.3 Interpolation Between Euclidean and Spectral Families

A pleasant consequence of the unified update rule (34) is that varying p and the choice of D jointly traces out a 2D family of optimizers, recovering established methods at its corners. Table 2 summarizes this structure.

The p-dependent family is particularly attractive: a single optimizer traces out the entire interpolation between Adam and Muon as p varies. The adaptive $p^\star$ selection of Section 2.2 then dynamically chooses the right point in this interpolation per layer per training phase, without manual specification. Additionally, this framework can be combined more efficiently with different structures of D scaling, where, if D is a diagonal matrix multiplication, the efficient computation of p^* can be reinstated from trace operator properties.

In practice, to reconcile the different update scales of Muon and Adam for example, in the SMuon (Adam) optimizer: we interpolate the learning rate logarithmically between that of Muon using the standard default implementation multiplicative factor (when $p^* = p_{\max}$), and that of the auxiliary AdamW optimizer that is used on non-matrix parameters (when $p^* = 1$), ensuring that the empirical RMS norm of updates remains rather stable across p^* values. This eliminates the need for an additional hyperparameter as most practical training scenarios using Muon use auxiliary Adam updates on parameters such as biases. Moreover, while analytical scaling factors could be found for the scale of the updates, they would require costly Schatten-p^* norm estimations at each step.

D.4 Recovering the Right Exponent

We show that $\alpha = 1/(2(p+1))$ is the *unique* exponent such that the symmetric preconditioned update

$$\delta W = -\eta\, D^{\circ -\alpha} \odot \mathrm{LMO}_{(p+1)}\big(D^{\circ -\alpha} \odot M\big), \tag{43}$$

simultaneously (i) recovers the Adam update at $p = 1$ and (ii) is invariant to global rescaling of the loss.

Proof. **Uniqueness via loss-rescaling invariance.** Under a global rescaling $L \to \lambda L$ the gradient, momentum, and second-moment accumulator transform as

$$G \;\to\; \lambda G, \qquad M \;\to\; \lambda M, \qquad D \;\to\; \lambda^2 D,$$

since D is an EMA of $G \odot G$. Substituting into (43),

$$\delta W \;\to\; -\eta\, \lambda^{-2\alpha} D^{\circ -\alpha} \odot \mathrm{LMO}_{(p+1)}\big(\lambda^{1-2\alpha}\, D^{\circ -\alpha} \odot M\big).$$

The Schatten-p LMO is positively homogeneous of degree $1/p$, i.e. $\mathrm{LMO}_{(p+1)}(\mu X) = \mu^{1/p}\mathrm{LMO}_{(p+1)}(X)$ for $\mu > 0$, so the expression reduces to

$$\delta W \;\to\; -\eta\, \lambda^{\,-2\alpha + (1-2\alpha)/p}\; D^{\circ -\alpha} \odot \mathrm{LMO}_{(p+1)}\big(D^{\circ -\alpha} \odot M\big).$$

For the update *direction* to be independent of λ (absorbing any scalar into the step size η), the exponent of λ must vanish:

$$-2\alpha + \frac{1-2\alpha}{p} = 0 \;\;\Longrightarrow\;\; \alpha = \frac{1}{2(p+1)}.$$

This equation has a unique solution for every $p \geq 1$, establishing necessity and sufficiency.

Recovery of Adam at $p = 1$. Setting $p = 1$ gives $\alpha = \frac{1}{4}$. At $p = 1$, the Schatten-2 LMO satisfies $\mathrm{LMO}_2(X) \propto X$ (the update is proportional to the matrix itself), so

$$\delta W \;\propto\; D^{\circ -1/4} \odot \left(D^{\circ -1/4} \odot M\right) \;=\; D^{\circ -1/2} \odot M,$$

which is exactly the Adam preconditioned update with RMS denominator $D^{1/2}$.

Recovery of Muon as $p \to \infty$. As $p \to \infty$, $\alpha = 1/(2(p+1)) \to 0$, so $D^{\circ -\alpha} \to \mathbf{1}$ and the preconditioning vanishes. The update reduces to $\mathrm{LMO}_\infty(M) = UV^T$, recovering the pure Muon orthogonalization step. □

E Scaling Behavior across Adaptive Geometries

E.1 Behavior Across Model Scale and Depth

We examine whether the adaptive geometry remains meaningful as the underlying model changes in scale. We first increase the language-model configuration from 275.7M to 716.6M parameters by changing the width of the blocks in the nanogpt architecture while keeping depth fixed (see Table 3 for validation loss with different model sizes). For plain SMuon, the time-averaged selected exponent increases from 28.58 to 37.43, while the fraction of layerwise selections satisfying $p^* \leq 10$ decreases from 39.5% to 21.8%. Thus, increasing width shifts the optimizer toward more Muon-like proxy-optimal geometries which corroborates common knowledge about Muon's scalability [Liu et al., 2025].

Table 3: Final / time-averaged validation loss as model size increases. Lower is better, averaged across three or two random seeds for the regular and larger configuration respectively.

Parameters	Muon	SMuon	SMuon (Adam)
275.7M	3.3780/3.9376	3.3829/3.9277	**3.3714/3.9128**
716.6M	3.2238/3.9128	3.2173/3.8810	**3.2096/3.8736**

Changing depth produces a different response. At fixed width, increasing the model from 6 to 24 blocks decreases plain SMuon's mean p^* from 38.47 to 24.42. These observations suggest that model parameter count alone is insufficient for geometry adaptation and model-dependent knowledge is needed.

The practical computational overhead also remains small at the larger scale. With the selector refreshed every 100 steps, SMuon (Adam) increases measured training-step time from 1336.0 to 1375.6 ms at 275.7M parameters (2.96% overhead), and from 3381.1 to 3508.0 ms at 716.6M parameters (3.75% overhead).

E.2 Batch-Size Ablation

We additionally test whether the adaptive behavior persists across batch sizes on MLP-Mixer/ImageNette. In Table 4, we compare batch sizes 128 and 512, approximately matching the optimizer-step budget (1460 and 1440 updates, respectively). For each optimizer and batch-size pair, the learning rate is independently selected from six candidates, after which the chosen configuration is evaluated over four matched seeds with identical initialization and data loading. Importantly, these experiments match the number of optimizer steps rather than the number of training examples, so the observed shifts in p^* cannot be attributed to batch size alone.

Plain SMuon improves over Muon by 1.02 percentage points at batch size 128 and 1.07 points at batch size 512. SMuon (Adam) improves over Muon by 1.55 and 0.57 points, respectively. The optimal p^* values also change with batch size: for plain SMuon, the fraction of decisions for all layers and timesteps satisfying $p^* < 8$ increases from 5.0% at batch size 128 to 21.4% at batch size 512. These lower-p^* selections are concentrated in the token-mixing output projections and classification head, while the stem and channel-mixing projections remain predominantly Muon-like.

Table 4: Final ImageNette validation accuracy (%) across batch sizes. Values are mean ± standard deviation over four matched seeds.

Batch size	Muon	SMuon	SMuon (Adam)
128	77.12 ± 0.24	78.14 ± 0.53	$\mathbf{78.67 \pm 0.48}$
512	77.53 ± 0.23	$\mathbf{78.60 \pm 0.48}$	78.10 ± 0.81

E.3 Distributed Training Considerations

In distributed settings, our p^* computation can be efficiently adapted. While we do not discuss practical implementation details, we provide a strategy for the computation of p^* when activation matrices A are sharded across different devices. Instead of communicating all activations to a single rank, our distributed implementation computes the local projected Gram contributions on each rank, which can then be combined via a single all-reduce of size $\mathcal{O}(\min(m, n))$ to correctly capture the global activation geometry.

More precisely, given activations A sharded along the batch dimension as $A = [A_1, \ldots, A_R]$ across R ranks (with $A_r \in \mathbb{R}^{n \times k_r}$ the local shard), the curvature term $B_{i,i} = \|A^T V_M e_i\|^2$ admits the decomposition $\|A^\top V_M e_i\|_F^2 = \sum_r \|A_r^\top V_M e_i\|_F^2$, so each rank can compute its local contribution $V_M^\top A_r A_r^\top V_M$ and a single all-reduce of size $\mathcal{O}(\min(m, n)^2)$ or $\mathcal{O}(\min(m, n))$ if only the diagonal is required recovers the global $B_{i,i}$. The momentum SVD $U_M \Sigma_M V_M^\top$ is computed redundantly on each rank from the (already-replicated) momentum buffer, incurring no additional communication. The alignment coefficients $C_{i,i} = (U_M^\top G_t V_M)_{i,i}$ are similarly local once gradients are reduced, which any data-parallel training loop already does. The total p^*-induced communication overhead is therefore one all-reduce of a vector or small matrix of size $\mathcal{O}(\min(m, n))$ per selector call, which at our default update interval of 100 steps is negligible relative to the per-step gradient all-reduce.

F Adaptive versus Fixed Schatten Geometry

Here, we want to test whether the benefit of SMuon comes from adapting the Schatten geometry itself, or whether similar performance can be obtained by choosing a single fixed exponent and retaining the same surrounding optimizer. We therefore compare adaptive p^* selection against fixed-p controls in both language modeling and vision.

F.1 NanoGPT Training Task

In Table 5, we compare SMuon (Adam) against an otherwise identical variant in which every matrix parameter uses a fixed exponent $p = 50$ throughout training, as using $p = 10$ or $p = 40$ degraded the performance. Since the NanoGPT experiment is a speedrun setting, we evaluate convergence by the number of optimizer steps required to reach common validation-loss thresholds. For six matched seeds, threshold-crossing times are estimated by linear interpolation between logged validation points.

Table 5: Adaptive versus fixed-p SMuon (Adam) on NanoGPT. Threshold crossings are averaged over six matched seeds. "Saved" reports the reduction in optimizer steps obtained by adaptive p^*.

Target loss	Fixed $p = 50$	Adaptive p^*	Steps saved	Relative
4.20	259.5	252.2	7.3	2.83%
4.00	367.3	358.2	9.1	2.49%
3.80	565.1	548.0	17.1	3.02%
3.60	986.7	962.9	23.9	2.42%
3.40	1545.3	1507.8	37.5	2.43%

Adaptive SMuon (Adam) reaches every tested threshold earlier in all six matched seeds. It also reduces the time-averaged validation loss from 3.9173 to 3.9048 (paired t-test $p = 2 \cdot 10^{-3}$), and improves the final validation loss from 3.3754 to 3.3663 ($p = 2 \cdot 10^{-3}$). Because the fixed and adaptive variants share the same p-dependent preconditioning mechanism, this comparison isolates the effect of adapting the geometry of updates.

Table 6: Adaptive versus independently tuned fixed Schatten exponents on MLP-Mixer/ImageNette after 100 epochs. Values are mean ± standard deviation over nine common seeds.

Method	Final train loss	Final validation accuracy (%)
Adaptive p^*	$\mathbf{0.00540 \pm 0.00179}$	82.66 ± 0.30
Fixed $p = 1.02$	0.02575 ± 0.00422	81.64 ± 0.37
Fixed $p = 3$	0.00613 ± 0.00195	$\mathbf{82.74 \pm 0.40}$
Fixed $p = 8$	0.00729 ± 0.00257	82.07 ± 0.41

F.2 Vision Training Task

We next test whether a single manually chosen exponent can recover the performance of adaptive selection in vision. We train an MLP-Mixer on ImageNette for 100 epochs and compare adaptive layerwise p^* against fixed values

$$p \in \{1.02, 3, 8\}.$$

Each fixed-p baseline is tuned independently.

The tuning stage uses a specific seed for 30 epochs. We first perform a coarse search over

$$\eta_{\text{Muon}} \in \{0.005, 0.01, 0.02, 0.04\}, \qquad \eta_{\text{Adam}} \in \{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\},$$

followed by local refinement over

$$\beta_1 \in \{0.9, 0.95\}, \qquad \beta_2 \in \{0.99, 0.999\}.$$

For the SMuon parameters, we use $\beta_{\text{sv}} = 0.9$, and the interval for p^* computations is fixed to 30. The remaining settings are shared across the search: $\epsilon = 10^{-8}$, $\eta_{\min} = 10^{-7}$, $p_{\min} = 1.02$, $p_{\max} = 50$.

Results are given in Table 6. Adaptive p^* obtains the lowest final training loss while reaching validation accuracy comparable to the best independently tuned fixed geometry. In particular, fixed $p = 3$ attains a slightly higher mean validation accuracy, whereas adaptive selection gives lower training loss and does not require choosing p through an external grid search. This shows that the RFR selector can recover performance comparable to the tuned fixed exponent while removing the Schatten exponent itself from the hyperparameter search.

Taken together with the NanoGPT control, these experiments make the case for geometry adaptation for out-of-the-box performance without LMO tuning.

G Alternative Computation Methods for the Schatten-p Update

Layerwise Remez-fitted Newton-Schulz. Here, we present a computational routine that extends the Björck/Newton-Schulz iteration [Björck and Bowie, 1971, Higham, 2008] underlying Muon to approximate the Schatten-$(p+1)$ norm update direction $U\Sigma^{1/p}V^T$. Starting from $X_0 = G/\|G\|_2$, the odd-degree update

$$X_{t+1} = a_t X_t + b_t\, X_t X_t^T X_t + c_t\, (X_t X_t^T)^2 X_t$$

decouples into a purely scalar recursion on the singular values: $\sigma_{t+1} = a_t\sigma_t + b_t\sigma_t^3 + c_t\sigma_t^5$. PolarExpress [Amsel et al., 2026] and Grishina et al. [2025] leverage the attractive fixed point of $f(x) = 1$ to derive Chebyshev-optimal coefficients for the orthogonalization target. No such contractive structure exists for $f(x) = x^{1/p}$. Therefore, we decompose the exponent as $q = p^{-1/T}$, so that each of the T iterations independently targets the mild perturbation $x \mapsto x^q$. Since $q \to 1$ as T grows, a degree-5 polynomial can match this target accurately on each layer. We fit the coefficients by solving the minimax problem

$$\min_{a_t,b_t,c_t} \max_{x\in[\varepsilon_t,1]} \left|a_t x + b_t x^3 + c_t x^5 - x^q\right| \quad \text{s.t.} \quad a_t + b_t + c_t = 1, \tag{44}$$

via SLSQP over a Chebyshev-node discretization. The pinning constraint prevents singular-value blow-up at $x = 1$, and the domain for iteration $t+1$ is updated to be the image of the fitted

polynomial on $[\varepsilon_t, 1]$. This yields the fastest routine we consider in the benchmark of Figure 2, effectively matching the efficiency of Muon's Newton-Schulz algorithm.

Concurrent alternatives. Two recent methods target the same primitive but do not fit our adaptive setting. Qi et al. [2026] run two coupled Newton-Schulz recursions to compute $A^{1/2}$ and $A^{-1/2}$ jointly, then recover $U\Sigma^{1/2}V^T = XA^{-1/2} \cdot A^{1/2}$. The iteration is accurate, but it doubles working memory and only supports dyadic exponents $p = 2^k$. A second line uses the Zolotarev integral $x^{-\alpha} = \frac{\sin(\alpha\pi)}{\pi} \int_0^\infty (t+x)^{-1}\, t^{-\alpha}\, dt$, discretized by Gauss-Legendre to reduce the fractional factor to k shifted resolvents $(G^TG + s_jI)^{-1}$. We solve all k in parallel with a single batched Cholesky on the $m \times m$ normal equations, using power iteration for the spectral-norm rescaling to avoid any SVD. This supports arbitrary p, but the batched factorization costs $\mathcal{O}(km^3)$ time and lacks the tensor-core throughput of matmul-only Newton-Schulz, which places it on the slow end of the Pareto frontier (Figure 2, green).

H Selecting a Maximum p Threshold

In practice, establishing a bounded search interval for the optimal Schatten exponent p^* is necessary to maintain both computational efficiency and numerical stability. We restrict our scalar line search to a maximum value of $p_{max} = 50$ based on two primary structural and algorithmic limitations:

- **Objective Function Saturation:** As p grows large, the singular value terms $\sigma^{1/p}$ and $\sigma^{2/p}$ approach unity. Consequently, the proxy optimality objective function $J(p)$ becomes essentially flat, as the correlation-to-curvature ratio reaches its saturation limit. At this stage, the computed update $U\Sigma^{1/50}V^T$ is practically indistinguishable from the exact Muon update UV^T, making further optimization along p redundant.
- **Estimator Precision and Numerical Degradation:** The precision of our hardware-efficient estimator for the fractional polar factor $U\Sigma^{1/p}V^T$ inherently worsens for highly elevated values of p. As demonstrated in Figure 7, direct polynomial approximations (such as Newton-Schulz with Remez coefficients) experience severe lift-off near the origin and fail to maintain acceptable error bounds when $p \geq 10$. While our primary Taylor approximation built atop the polar factor degrades much more gracefully and remains stable at higher exponents, its relative Frobenius error still increases as $p \to \infty$. Capping p_{max} at 50 ensures that the optimizer relies on the Taylor approximation only where it remains demonstrably accurate.

I On the Stability of Optimal p Values

In this section, we analyze the empirical stability of the optimal Schatten exponent p^* throughout the training process. Observations demonstrate that the trajectory of p^* values remains relatively stable over time. Although the precise p^* values are not strictly identical across independent runs, Figure 4 illustrates that specific layer types consistently exhibit similar behavioral patterns regardless of the random seed.

Particularly, distinct architectural components naturally gravitate toward different optimal update geometries:

- **Attention Weights:** These matrices typically require higher-rank updates, closely mirroring Adam-like update geometries at the beginning of the training process.
- **MLP Layers:** In contrast, MLP layers generally necessitate slightly lower-rank updates throughout training, while still maintaining an update geometry that is very close to strict orthogonality.

This consistent structural divergence between layer types further highlights the practical benefit of utilizing a dynamically adapted, layer-wise p^* selector over fixed global hyperparameter choices.

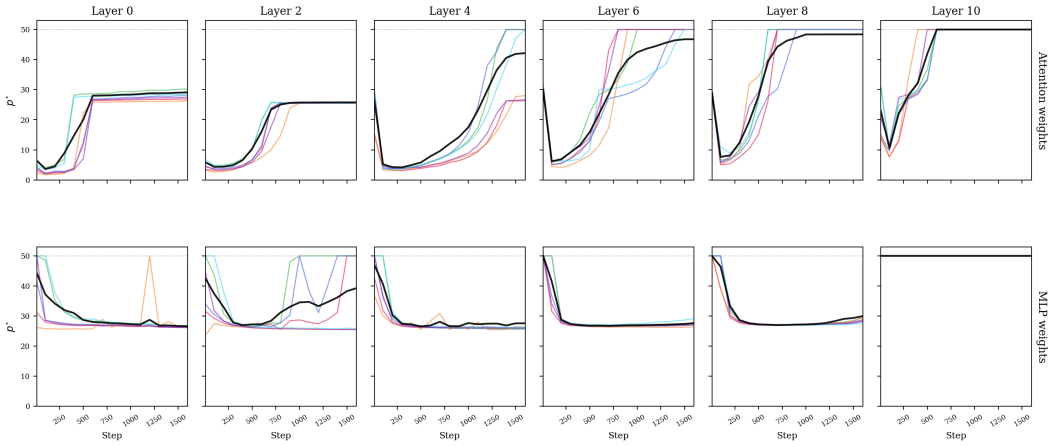


Figure 4: Here we plot the average p^* values of attention and MLP layer weight matrices during training with the SMuon (Adam) optimizer across 6 random seeds, p^* values are computed every 100 steps with an EMA of 0.95.

I.1 Ablation Study: Update Frequency and EMA Regularization

We conducted an ablation study to investigate the interplay between update frequency (measured by the step gap) and the Exponential Moving Average decay rate on training stability and final convergence. In this case, we train an MLP-Mixer - Small model on the ImageNette dataset using a batch size of 62 and a learning rate of 0.02 with the SMuon (Adam) optimizer. The results, averaged across three random seeds, are summarized in Table 7.

The empirical data demonstrates that the optimal EMA decay rate is highly dependent on the update frequency. Models operating with highly frequent updates (Gap = 50) exhibited significant sensitivity to the EMA parameter. Specifically, in these high-noise regimes, a strong EMA ($\beta_p = 0.9$) was critical for stabilizing the training process, achieving a lower final loss (0.0069) compared to lower EMA values (0.0088 for $\beta = 0.0$). Conversely, when updates were less frequent (Gap ≥ 300), the larger accumulated data batches were inherently more stable, and moderate-to-low EMA values proved optimal (e.g., 0.0060 for Gap 600, $\beta = 0.5$).

While moderate EMAs suffice for low-frequency updates in smaller-scale tests, larger models necessitate a stable tracking mechanism for a significantly higher number of parameter matrices. Thus enforcing a stricter constraint on the EMA strength needed for stable training.

Table 7: Final training loss across different update gaps and EMA values, averaged over three random seeds. In high-frequency update regimes (Gap = 50), higher EMA values are strictly required to stabilize the loss.

Update Gap	β_p **= 0.0**	β_p **= 0.5**	β_p **= 0.9**
50	0.0088	0.0086	**0.0069**
150	0.0073	**0.0068**	0.0074
300	**0.0062**	0.0074	0.0069
600	0.0063	**0.0060**	0.0063

J Efficient Estimation of Optimal p via Frobenius-Anchored Randomized SVD

Evaluating the exact objective function $J(p)$ of Proposition 2 requires the full singular value decomposition (SVD) of matrices $A, G \in \mathbb{R}^{N \times D}$, an $\mathcal{O}(ND^2)$ operation that is prohibitively expensive for large layers. To maintain computational tractability, we adapt existing randomized SVD techniques to estimate the optimal parameter $p^* = \arg\max_p J(p)$. Because the function relies on continuous

deformations of Schatten norms, a naive truncation of the spectrum could introduce severe asymptotic errors. We resolve this by anchoring a low-rank randomized SVD with an exact computation of the Frobenius energy, yielding a strictly bounded surrogate objective. The following analysis is run for the simplified random feature regression proxy that does not account for momenta.

J.1 Asymptotic Behavior and Structural Dependence

To understand the necessity of modeling the uncomputed spectral tail instead of only the topk singular values, we first analyze the behavior of $J(p)$ at the boundaries of its domain $p \in (1, \infty)$. Let $q_G(p) = 1 + \frac{1}{p}$ and $q_A(p) = \frac{2(p+1)}{p-1}$ denote the exponents of the inner sums for G and A, respectively.

- **The Lower Bound ($p \to 1^+$):** As p approaches 1, $q_A(p) \to \infty$. The denominator sum becomes heavily dominated by the largest singular value, effectively collapsing to the spectral norm $\sigma_1(A)$. If matrix A exhibits a dominant spectral norm with a sharp drop-off, the objective function is heavily penalized, pulling the optimal p^* toward 1.
- **The Upper Bound ($p \to \infty$):** As p approaches infinity, $q_G(p) \to 1$. The numerator sum converges to the nuclear norm $\|G\|_* = \sum \sigma_i(G)$. If matrix G possesses a heavy power-law tail (e.g., typical of unnormalized gradients), the nuclear norm diverges. This infinite mass acts as a mathematical attractor, forcing $p^* \to \infty$.

Consequently, the optimal p^* is entirely dictated by the structural dichotomy between A and G. If the matrices are forced toward isometry (e.g., via orthogonal initialization) or their variance is strictly constrained (e.g., via RMSNorm), their spectra flatten or decay exponentially.

J.2 Bounding the Spectral Tail

To approximate $J(p)$ without computing the full spectrum, we utilize the randomized subspace iteration framework [Halko et al., 2011] to efficiently extract the top k singular values, $\hat{\Sigma}_k$, and estimate the residual spectral norm $R \approx \sigma_{k+1}$ using Gaussian test vectors.

While R bounds the maximum magnitude of the uncomputed tail, it does not constrain its volume. We constrain the volume by algebraically computing the exact uncomputed Frobenius energy: $E_M = \|M\|_F^2 - \sum_{i=1}^{k} \hat{\sigma}_i^2$. This leaves exactly $d_M = \min(N, D) - k$ singular values constrained by a known total energy E_M and a strict individual maximum R_M.

The inner sums of $J(p)$ require optimizing $\sum x_i^{q/2}$ subject to $\sum x_i = E$ and $x_i \in [0, R^2]$, where $x_i = \sigma_i^2$. For $p \in (1, \infty)$, the exponent $q_G/2 \in (0.5, 1)$ making the transformation strictly concave, whereas $q_A/2 > 1$ making it strictly convex. By majorization, the extreme values of these sums occur when the remaining energy E is packed into singular values of maximum allowable size R^2. We define the worst-case tail mass operator $\mathcal{T}$ as:

$$\mathcal{T}(E, R, d, q) = \begin{cases} d \left(\frac{E}{d}\right)^{q/2} & \text{if } \frac{E}{R^2} > d \\ E \cdot R^{q-2} & \text{otherwise} \end{cases} \tag{45}$$

This yields a strict, worst-case lower bound for the numerator (concave minimization) and upper bound for the denominator (convex maximization). The final surrogate objective, which can be evaluated in $\mathcal{O}(k)$ time, is defined as:

$$\tilde{J}_{lower}(p) = \frac{\left(\sum_{i=1}^{k} \hat{\sigma}_i(G)^{q_G} + \mathcal{T}(E_G, R_G, d_G, q_G)\right)^{\frac{p}{p+1}}}{\left(\sum_{j=1}^{k} \hat{\sigma}_j(A)^{q_A} + \mathcal{T}(E_A, R_A, d_A, q_A)\right)^{\frac{p-1}{2(p+1)}}} \tag{46}$$

J.3 Algorithm

The complete procedure for dynamically estimating p^* is detailed in Algorithm 1. The bounded 1D optimization is performed in log-space ($x = \ln(p-1)$) to preserve numerical precision near the asymptotic boundary $p \to 1$.

Algorithm 1 Frobenius-Anchored Estimation of Optimal Parameter p^*

Input: Matrices $A, G \in \mathbb{R}^{N \times D}$, truncation rank k, power iterations q, max bound p_{max}
Output: Optimal parameter estimate p^*
$d_A \leftarrow \min(N, D) - k, \quad d_G \leftarrow \min(N, D) - k$
Compute exact energies: $F_A^2 \leftarrow \|A\|_F^2, \quad F_G^2 \leftarrow \|G\|_F^2$
for $M \in \{A, G\}$ **do**
    Draw Gaussian matrix $\Omega \in \mathbb{R}^{D \times (k+5)}$
    $Y \leftarrow M\Omega$
    for $i = 1$ **to** q **do** ▷ Subspace Iteration
        $Y \leftarrow M(M^T Y)$ with intermediate orthonormalization
    end for
    Extract orthonormal basis Q from Y
    Compute SVD of projected matrix: $U, \hat{\Sigma}_M, V^T \leftarrow \text{SVD}(Q^T M)$
    Draw Gaussian vectors $\omega \in \mathbb{R}^{D \times 5}$
    Estimate residual norm: $R_M \leftarrow \frac{1}{\sqrt{D}} \max \|(I - QQ^T)M\omega\|_2$
    Compute uncomputed energy: $E_M \leftarrow \max\left(0, F_M^2 - \sum_{i=1}^{k} \hat{\sigma}_{M,i}^2\right)$
end for
Define bounded surrogate objective $\tilde{J}_{lower}(p)$ using $\hat{\Sigma}_A, \hat{\Sigma}_G, R_A, R_G, E_A, E_G, d_A, d_G$
$x^* \leftarrow \arg\max_{x \in [\ln(0.01), \ln(p_{max}-1)]} \tilde{J}_{lower}(1 + \exp(x))$ ▷ Log-space optimization
return $p^* = 1 + \exp(x^*)$

Variance Reduction

Stochastic subsampling introduces variance that can destabilize the trajectory of p^*. To mitigate this without increasing the sampling cost, we employ *Spectral Momentum*. Instead of maximizing $J(p)$ on the instantaneous subsample Σ_t, we maintain an exponential moving average (EMA) of the spectral statistics over training steps:

$$\bar{\Sigma}_t = \beta \bar{\Sigma}_{t-1} + (1 - \beta)\Sigma_{sub,t} \tag{47}$$

This temporal smoothing dampens the noise from random sampling while allowing p^* to adapt to the shifting geometry of the loss landscape.

K Experiment Hyperparameters

NanoGPT experiments Based on Du and Su [2026], our NanoGPT training setup processes 393,216 tokens per global step, achieved using a global batch size of 8 sequences with a sequence length of 49,152 (48×1024). The models are trained for a total of 1,695 iterations on the FineWeb dataset. We use a 12-layer GPT architecture with 6 attention heads and an embedding dimension of 768. All models utilize a learning rate schedule with 0 warmup iterations and a linear cooldown over the final 762 iterations (45% of total steps), with weight decay set to 0.0.

For the SMuon and SMuon (Adam) optimizers, the hyperparameters remain unchanged from those that were optimal for Muon (which were already tuned). The Muon base learning rate is set to 0.05 with momentum parameters of 0.95. The auxiliary Adam optimizer applied to the embeddings, LM head, and scalars uses a learning rate of 0.008 and $\beta = (0.8, 0.95)$. For the Schatten exponent updates in SMuon, we use an update interval of 100 steps, bounds of $p_{\min} = 1.02$ and $p_{\max} = 50.0$, conduct a p^* evaluation every 100 steps with $\beta_p = 0.95$ to keep the training stable. SMuon (Adam) uses a β_2 value of 0.95.

For the standard Adam baseline, we use the exact same learning rate (0.008) as that of the auxiliary Adam in the Muon setup.

For the MuAdam optimizer, we tuned the learning rate multiplier on the originally tuned learning rate across the grid $\{0.001, 0.01, 0.1, 0.5, 1.0, 1.5\}$ and selected the optimal value. Additionally, we experimented with the momentum parameters (β_1, β_2) for MuAdam, testing both $(0.9, 0.999)$ and $(0.95, 0.95)$. We found the latter to perform more robustly and ended up sticking with $(\beta_1, \beta_2) = (0.95, 0.95)$ for our final configurations.

Vision experiments We evaluate our optimization methods on image classification using the ImageNette dataset. We train two distinct architectures sourced from the `timm` [Wightman, 2019] library: a Vision Transformer (ViT-Small, `vit_small_patch16_224`) and an MLP-Mixer (`mixer_s16_224`). All models are trained for 20 epochs using a total batch size of 256 in `bfloat16` mixed precision. Our data processing pipeline applies standard augmentations including RandAugment, random horizontal flipping, resizing to 256, and center cropping to 224×224. The learning rate is decayed using a cosine annealing schedule without restarts. To ensure robust evaluation, all hyperparameter configurations are repeated across three random seeds $(256, 512, 1024)$.

For the AdamW baseline, we sweep the learning rate over the grid $\{0.001, 0.005, 0.01, 0.02\}$ with a fixed weight decay of 0.01. For the Muon, SMuon and SMuon (Adam) optimizers, we sweep the primary learning rate over $\{0.001, 0.003, 0.005, 0.01, 0.02\}$ for the ViT and $\{0.003, 0.005, 0.01, 0.02\}$ for the MLP-Mixer. The core Muon/SMuon parameters utilize a momentum of 0.95 and a weight decay of 0.05, with $\beta_2 = 0.99$ for SMuon (Adam). The auxiliary Adam optimizer—applied to biases, normalizations, and other non-matrix parameters—uses a fixed learning rate of 0.001 and a weight decay of 0.01. The Schatten exponent is updated every epoch with a singular value momentum of 0.95.

LoRA Fine-Tuning We evaluate our optimization methods on a low-rank adaptation (LoRA) fine-tuning task using the Qwen-2.5-0.5B model on the GSM8K dataset. The base model is loaded in 4-bit quantization using `bfloat16` compute precision. We apply LoRA to both the attention and MLP projections (specifically `q_proj`, `k_proj`, `v_proj`, `up_proj`, `gate_proj`, `query_key_value`, `dense_h_to_4h`, and `dense_4h_to_h`) with a rank of $r = 32$ and $\alpha = 64$.

All models are trained for 3 epochs using a batch size of 32. We utilize a cosine learning rate schedule with a linear warmup over the first 10% of total training steps. To ensure statistical robustness, all hyperparameter configurations are evaluated across 3 random seeds.

For the AdamW baseline, as well as the Muon and SMuon optimizers, we sweep the primary learning rate across the grid $\{0.0003, 0.001, 0.003, 0.005, 0.01, 0.02\}$. For the Muon and SMuon configurations, the core 2D matrix parameters use a momentum of 0.95 (and $\beta_2 = 0.95$ for the preconditioned version), while all 1D parameters (such as biases and normalizations) are handled by an auxiliary Adam optimizer.

For the SMuon variants, the Schatten exponent is initialized at the minimum bound, $p = 1.02$ to avoid computational instabilities and updated using the method of Proposition 1. We apply a Schatten update interval of 50 steps and a singular value momentum (`sv_momentum`) of 0.95.

K.1 Learning Rate Invariance

An important part of evaluating optimizers in realistic deployment scenarios is to measure their sensitivity to hyperparameters. In practice, the Muon optimizer is also very appreciated for its relatively stable behavior across learning rate values. In this setting, we train a ViT - small model and an MLP-Mixer small model with patch size 16 on the ImageNette dataset. In this experiment, the RandAugment augmentation method is unused to reduce variance in the training loss and measure the capability of the optimizer to fit the training set across different learning rates. Here, we use a batch size of 128 with p^* updates every epochs and $\beta_p = 0.95$. Results are displayed in 5.

As seen in Figure 5, the learning rate of SMuon and SMuon (Adam) is not radically different to that of Muon, with SMuon (Adam) only demonstrating slightly higher sensitivity to the learning rate on the MLP-Mixer - small model.

L Investigating the Retroactive Effect of Using Adaptive Updates

In this rather exploratory section, we investigate the use of adaptive geometry updates. Our main endeavor is to investigate whether the geometry of updates at step t can have an influence on the optimal geometry of updates at step $t + 1$. To this end, we train an MLP-Mixer - Small model on the ImageNette dataset, using either Muon of SMuon. In this case, Muon is equipped with a p^* approximator that runs 12 times at regular intervals on each experiment, that is ran on three different initialization strategies using three random seeds for each. Our results are presented in Figure 6.

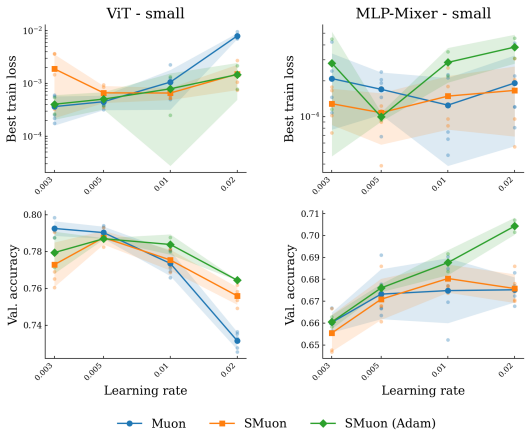


Figure 5: Learning rate invariance curves for two models trained on the ImageNette dataset

Figure 6 reveals three convergence regimes. `kaiming_normal` and `xavier_uniform` both favour Muon over SMuon ($d \approx -0.35$, $p \approx 0.07$ in each case), with divergence localised to `ch_fc2` and `tok_fc1` while `ch_fc1` saturates at $p^* = 50$ for both optimisers. Under `orthogonal` initialisation, both optimizers decline monotonically to $\bar{p}^* \approx 35$ with no detectable difference ($d \approx 0.03$, $p \approx 0.87$). Initialisation explains more variance in p^* than optimizer choice, and no pairwise difference survives Bonferroni correction ($\alpha_{\text{adj}} = 0.025$).

Therefore, it appears this experiment does not give any conclusive evidence of a retroactive effect between the optimizer update rule and the optimality of the geometry for the update at later training steps. However, this study, beyond single step dynamics, could prove particularly useful in the future to encourage the use of optimizers with strong positive retroactive effects.

Finally, this result can be interpreted positively, as strong retroactive effects between update rules and p^* trajectories would severely limit the applicability of single step surrogate models.

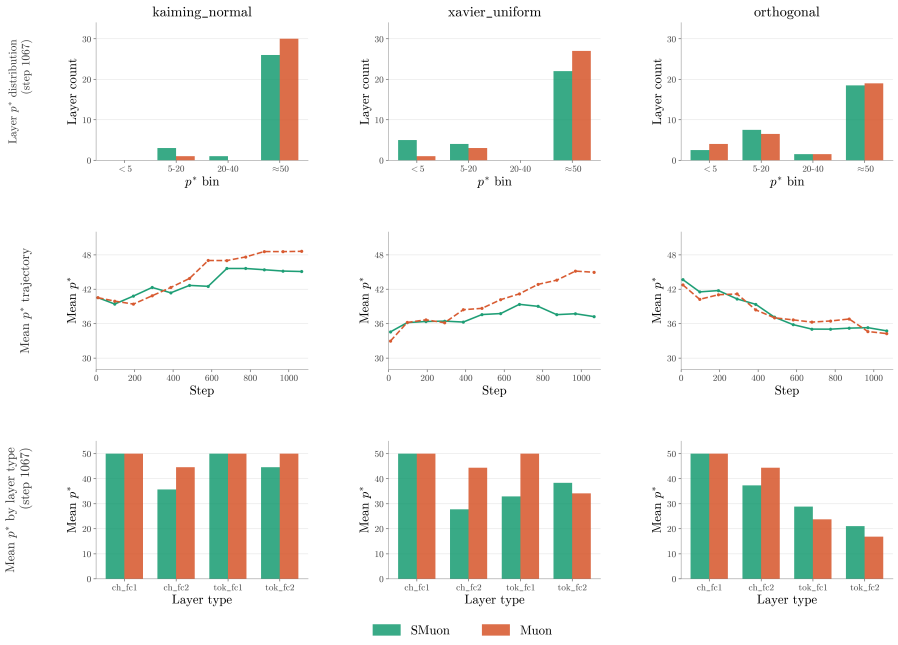


Figure 6: Comparison of p^* dynamics between Muon and SMuon optimizers across three weight initialization strategies (`kaiming_normal`, `xavier_uniform`, `orthogonal`). Each column corresponds to one initialization. Top row: distribution of final p* values across the 31 network layers, binned into four ranges. Middle row: mean p^* trajectory over 12 checkpoints (steps 10–1067). Bottom row: mean final p^* broken down by layer type (channel-mixing and token-mixing branches, first and second fully-connected layers).

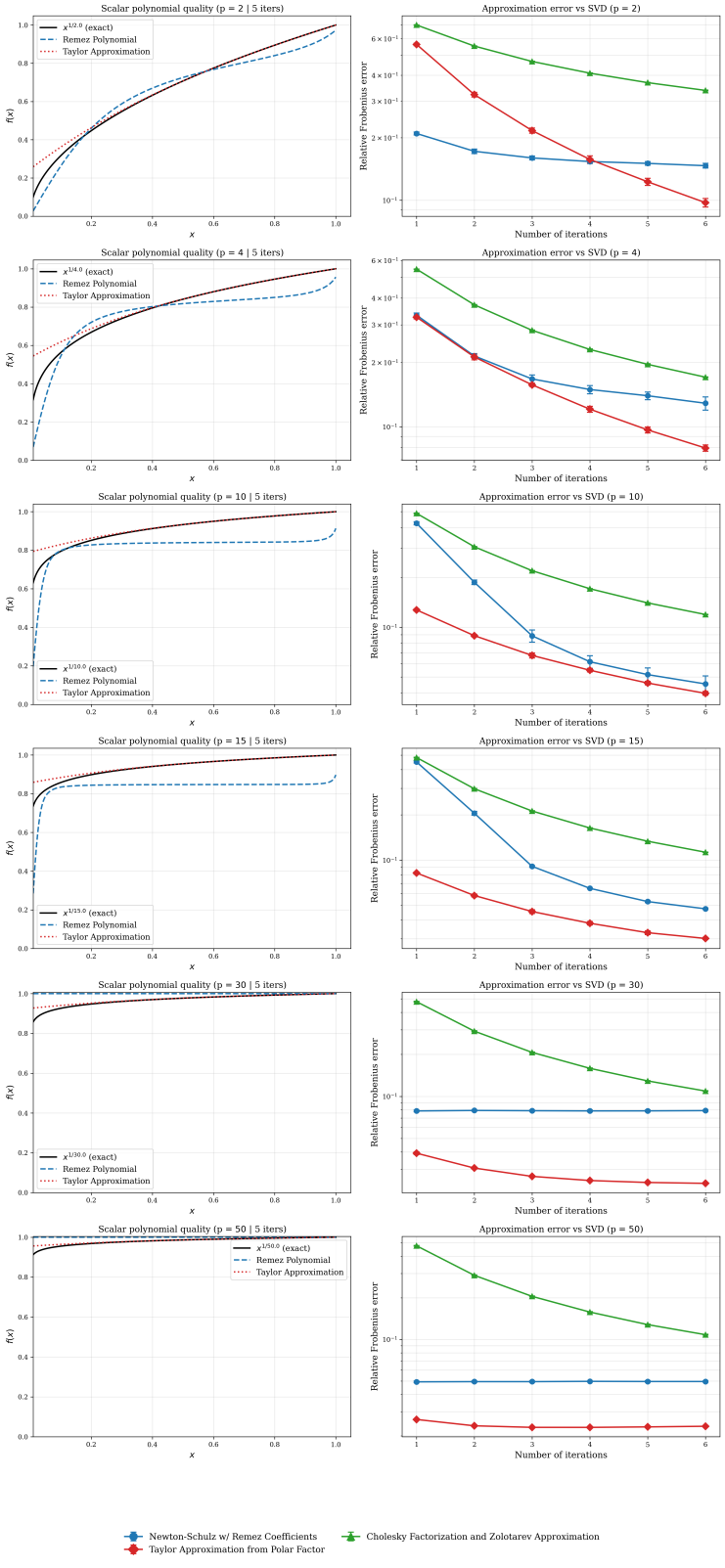


Figure 7: We plot the approximation of quality of $U\Sigma^{1/p}V^T$ for the different methods we propose.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We validate the claims from the end of the introduction theoretically and experimentally.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper is clear about the heuristic nature of the adaptive regime selector model, which can be refined in future work, as stated in Section 2.2 (remark) and Section 7.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: To our knowledge, the proofs are correct and the assumptions stated and discussed.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes the code and appendices give the information to reproduce the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: A recent anonymized version of the code is given as part of the supplementary materials.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, we use the standard train test splits of the evaluated datasets and provide the code for this.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All experiments are ran on minimum 3 random seeds.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper details matrix approximation results on an RTX 4090 GPU. The NanoGPT experiment is ran on 2 $\times$ A100 GPUs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: It does.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: This paper's contribution is not tied to particular applications, let alone deployments.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: No models trained in this paper constitute a risk of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: References to previous publications and sources of code are given.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Code is given as part of the supplementary.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [No]

Justification: LLMs did not impact the core methodology, rigor or originality of the research. They were used for polishing writing and additionally checking validity of theoretical results.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.