MANO: RESTRIKING MANIFOLD OPTIMIZATION FOR LLM PRETRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

While large language models (LLMs) have emerged as a significant advancement in artificial intelligence, optimizer design for LLMs has attracted much attention and costs huge resources. In this line of research, advanced optimizers, such as Adam and Muon, with complex operations are mainstream methods, while conventional manifold optimization methods have been largely overlooked due to their poor performance in large-scale model optimization. In this study, we try to restrike manifold optimization for training LLMs. We design a novel Manifold-Normalized Optimizer (Mano) that trains LLMs on a special rotational Oblique manifold implicitly. Mano's Oblique geometry actually corresponds to a product of neuron-wise hyperspheres that make training LLMs easy even with simple momentum. We also theoretically analyzed Mano in terms of convergence and Fisher information. Experimental results demonstrate that Mano outperforms AdamW and Muon across multi-scale LLaMA and Qwen3 models. Meanwhile, Mano incurs lower computational and memory costs than Muon and Adam.

1 INTRODUCTION

Large language models (LLMs) represent a major milestone in artificial intelligence and computing technology (Kumar, 2024; Gao et al., 2025). Nevertheless, training LLMs with billions of parameters requires specialized hardware accelerators, such as GPUs and NPUs, which consume substantial energy and result in enormous computational costs (Samsi et al., 2023; Chowdhury et al., 2025; Hoffmann et al., 2022). Consequently, advanced optimization methods are critical for reducing the training costs of LLMs.

Adam is widely regarded as one of the most popular optimizers in deep learning, with its variants being extensively used to train deep neural networks across a wide range of tasks (Kingma & Ba, 2015). In particular, AdamW, which decouples weight decay from the original gradient update of Adam, has been the dominant optimizer in training LLMs with broad infrastructure support for large-scale distributed training (Loshchilov & Hutter, 2018; Kunstner et al., 2024). However, by relying on diagonal estimates of per-parameter curvature, Adam-based optimizers ignore spectral information and fail to capture the subspace structures of the parameter matrix. To address this limitation, Muon introduces the Newton–Schulz iteration to approximate matrix orthogonalization, allowing uniform exploration across spectral directions in the loss landscape (Jordan et al., 2024). However, this spectral normalization strategy compresses relative singular-value information encoded in the parameter space and gradient toward the matrix-sign limit, which may lead to suboptimal performance in some settings (Su, 2025).

For optimization problems with a manifold structure, algorithms have been developed to exploit this structure from the perspective of differential geometry and numerical analysis, referred to as manifold optimization methods (Absil et al., 2008; Hu et al., 2020; Fei et al., 2025). However, these traditional manifold optimization methods have been overlooked for LLMs. In this study, we seek to adapt and extend manifold optimization methods to LLM training, where the underlying manifold structure of natural language or optimal parameters is not explicitly known or well understood, relaxing the constraints and assumptions typically required. Instead of retracting and constraining the parameters onto a manifold surface, we project momentum onto tangent vector spaces defined by normalized copies of the parameters, normalize the projected update on the Oblique manifold, and apply it through Euclidean descent. We further study a periodically rotating Oblique geometry

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

that alternates column- and row-wise normalization across optimization steps; its product structure decomposes each high-dimensional matrix update into lower-dimensional neuron-wise blocks, enabling local control of the momentum direction and update scale.

By restriking manifold optimization with these reformed strategies, we propose a new class of optimizer, **MA**nifold-**N**ormalized **O**ptimizer (**Mano**). Throughout this paper, we seek to unravel the hidden potential of manifold optimization methods in today's infrastructure for training neural networks. This work makes four main contributions:

- To our knowledge, we are the first to successfully restrike manifold optimization techniques for practical LLM training with several key insights, while traditional manifold methods perform very poorly on LLMs mainly due to explicit hard constraints on model weights.
- We design a novel and efficient optimizer, **Mano**, that simply updates model weights on a special rotational Oblique manifold implicitly. An Oblique manifold naturally forms a product of neuron-wise hyperspheres, where training LLM becomes much easier.
- We theoretically analyze Mano in terms of convergence and Fisher information. Meanwhile, Mano incurs lower computational and memory costs than Muon and Adam, as Mano only uses linear-cost reductions and element-wise operations.
- Extensive experiments demonstrate that Mano outperforms AdamW and Muon in validation loss at every evaluated LLM architecture and data corpus, while retaining substantially lower per-step overhead than Muon.

2 PRELIMINARIES

A Riemannian manifold $\mathcal{M}$ is a smooth geometric space equipped with a metric that defines a smoothly varying inner product on the tangent space of $\mathcal{M}$. Manifold optimization concerns the problem of minimizing a real-valued loss function f over such a manifold, i.e., $\min_{W\in\mathcal{M}} f(W)$, where $f : \mathcal{M} \to \mathbb{R}$. With this definition, vanilla SGD that optimizes a loss function over the Euclidean vector space $\mathbb{R}^n$ can be interpreted as operating on a Riemannian manifold $(\mathbb{R}^n, g_{ij})$ with metric $g_{ij} = \delta_{ij}$. Bonnabel (2013) generalizes SGD to perform updates on Riemannian manifolds through the following operations:

$$\begin{cases} D_t = \mathrm{proj}_{\mathcal{T}_{W_t}\mathcal{M}}(G_t), \\ W_{t+1} = \mathrm{proj}_{\mathcal{M}}(W_t - \eta_t D_t) \approx \exp_{W_t}(-\eta_t D_t), \end{cases} \tag{1}$$

where $\exp_{W_t}$ is the exponential map that maps a tangent vector at W_t to the point on $\mathcal{M}$ reached by following the corresponding geodesic. First, the gradient G_t is orthogonally projected onto the tangent space $\mathcal{T}_{W_t}\mathcal{M}$ to determine the Riemannian gradient D_t. A gradient step is then performed in the direction $-\eta_t D_t$ and mapped back to the manifold surface. While the exponential map provides the geometrically exact update, its high computational cost is often avoided through numerical retractions. A retraction typically performs an orthogonal projection back onto $\mathcal{M}$, serving as an efficient first-order approximation that maintains the manifold constraint (Bonnabel, 2013).

However, traditional Riemannian manifold optimization strategies often fail to generalize to modern neural networks on general tasks, particularly LLMs. Certain manifolds, such as the Stiefel manifold, require expensive matrix decompositions such as SVD or QR, which impose substantial optimization overhead. Furthermore, hard manifold constraints can restrict the model's ability to explore the loss landscape, especially when the chosen geometric structure does not align with the underlying objective.

Instead of framing natural language modeling or the optimal parameter solution within an arbitrary Riemannian manifold, we hypothesize that the learning trajectory and each constituent update step can be mapped onto smooth surfaces whose geometric structures facilitate convergence. Motivated by this hypothesis, the next section describes how we reform manifold optimization into a momentum optimizer without constraining the model parameters. We discuss additional manifold optimization techniques and recent LLM optimizers in Section 5.

3 METHODOLOGY

In this section, we reformulate traditional manifold optimization strategies into a momentum-driven optimizer and detail the configuration of rotating manifold normalization.

3.1 REFORMED MANIFOLD OPTIMIZATION

To formulate our manifold optimization methodology, we define a tangent-space projector $\mathrm{proj}_{\mathcal{T}_P\mathcal{M}}(Q)$ of a matrix Q onto the first-order approximation of the manifold surface $\mathcal{M}$ around P, and a manifold normalization operation $\mathcal{N}_{\mathcal{M}}(A)$ that constrains a matrix A to the target manifold. For a weight $W_t \in \mathbb{R}^{m\times n}$, gradient $G_t := \nabla f(W_t)$, and learning rate η_t at step t, we formulate the reformed update rule as

$$\begin{cases} \hat{W}_t = \mathcal{N}_{\mathcal{M}}(W_t), \\ D_t = \mathrm{proj}_{\mathcal{T}_{\hat{W}_t}\mathcal{M}}(G_t), \\ \hat{D}_t = \mathcal{N}_{\mathcal{M}}(D_t), \\ W_{t+1} = W_t - \eta_t \hat{D}_t. \end{cases} \tag{2}$$

We emphasize that this reformed update rule differs from the original definition of manifold optimization, $\min_{W\in\mathcal{M}} f(W)$, which defines both the function domain and parameters on the Riemannian manifold. Our update rule imposes a soft manifold constraint: it projects each update onto a tangent surface defined by a normalized copy of the parameters while keeping the objective and parameter solution in Euclidean space.

Among popular matrix manifolds, we select the Oblique manifold (Roberts & Ursell, 1960) because manifold normalization and tangent projection require only dimension-wise reductions and element-wise operations. For $P \in \mathbb{R}^{m\times n}$, the Oblique manifold is

$$\mathcal{OB}(m,n) := \left\{ P \in \mathbb{R}^{m\times n} : \mathrm{diag}(P^\top P) = \mathbf{1}_n \right\}. \tag{3}$$

For matrices $A, B \in \mathbb{R}^{m\times n}$, let $\langle A, B\rangle := \mathrm{tr}(A^\top B)$ and $\|A\|_F := \sqrt{\langle A, A\rangle}$. Define element-wise multiplication and division by $(A \odot B)_{ij} := A_{ij}B_{ij}$ and $(A \oslash B)_{ij} := A_{ij}/B_{ij}$, respectively. Define row- and column-wise inner products and norms by

$$\begin{aligned} \langle A, B\rangle_{\mathrm{row}} &:= \left(\langle A_{i,:}, B_{i,:}\rangle\right)_{i=1}^{m}, & \|A\|_{2,\mathrm{row}} &:= \left(\|A_{i,:}\|_2\right)_{i=1}^{m}, \\ \langle A, B\rangle_{\mathrm{col}} &:= \left(\langle A_{:,j}, B_{:,j}\rangle\right)_{j=1}^{n}, & \|A\|_{2,\mathrm{col}} &:= \left(\|A_{:,j}\|_2\right)_{j=1}^{n}. \end{aligned} \tag{4}$$

Dimension-wise multiplication and division use broadcasting, with a zero block mapped to zero under division. For $k \in \{\mathrm{row}, \mathrm{col}\}$, use $(\|A\|_{2,k}, \langle A, B\rangle_k) = (\|A\|_{2,\mathrm{row}}, \langle A, B\rangle_{\mathrm{row}})$ when $k = \mathrm{row}$ and $(\|A\|_{2,k}, \langle A, B\rangle_k) = (\|A\|_{2,\mathrm{col}}, \langle A, B\rangle_{\mathrm{col}})$ when $k = \mathrm{col}$. For this orientation, define

$$\hat{W}^k := W \oslash \|W\|_{2,k}, \quad V_k(W)[M] := M - \hat{W}^k \odot \langle M, \hat{W}^k\rangle_k. \tag{5}$$

When the corresponding blocks are nonzero, $\hat{W}^{\mathrm{col}} \in \mathcal{OB}(m,n)$ and $(\hat{W}^{\mathrm{row}})^\top \in \mathcal{OB}(n,m)$. This manifold choice divides the extremely high-dimensional parameter space into many lower-dimensional subspaces, where each subspace contributes its own manifold gradient.

The resulting update can be viewed as momentum-based optimization on a product of neuron-wise hyperspheres, where each active row or column block is normalized independently, and its momentum is projected onto the corresponding tangent space.

When integrating the Oblique manifold into Equation 2, a fixed orientation captures only one family of amplitude changes. Row-wise and column-wise projections remove different amplitude-aligned components of the momentum. To avoid relying on a single view of the weight geometry, we introduce a **rotational manifold** scheme that alternates between row-wise and column-wise normalization across successive iterations. Applying this rotation to both the tangent space of the parameters and the update yields an Oblique geometry with oscillating orientation across optimization steps. Integrating this rotational scheme with the reformed update rule yields the **Mano** optimizer, detailed in Algorithm 1.

3.2 THE MANO OPTIMIZER

Given $W_t \in \mathbb{R}^{m\times n}$, let $G_t := \nabla f(W_t)$, μ be the momentum coefficient, λ_{wd} the weight-decay coefficient, and s the update-rescaling factor. The orientation k_t selects row- or column-wise operations, with active block dimensions $d_{\mathrm{row}} := n$ and $d_{\mathrm{col}} := m$. Algorithm 1 provides the complete matrix update.

Algorithm 1: The Mano Optimizer

Require: Weight $W_t \in \mathbb{R}^{m\times n}$, learning rate η_t, momentum coefficient μ, weight decay λ_{wd}, and rescaling factor s.

Initialize $M_{-1} \leftarrow 0$.
for each step $t = 0, 1, \ldots$ **do**
  $G_t \leftarrow \nabla f(W_t)$
  $M_t \leftarrow \mu M_{t-1} + G_t$
  $k_t \leftarrow$ row if $t \bmod 2 = 0$, and $k_t \leftarrow$ col otherwise
  $\hat{W}_t^{k_t} \leftarrow W_t \oslash \|W_t\|_{2,k_t}$
  $V_t \leftarrow V_{k_t}(W_t)[M_t]$
  $\hat{V}_t \leftarrow V_t \oslash \|V_t\|_{2,k_t}$
  $W_{t+1} \leftarrow W_t - \eta_t\big(s\sqrt{d_{k_t}}\,\hat{V}_t + \lambda_{\mathrm{wd}} W_t\big)$
end for

The Mano optimizer can be summarized as **neuron-wise manifold optimization with soft constraints**, with the reformed strategies outlined as follows:

- The parameters W_t are not explicitly constrained to the manifold; the update follows weight decay and Euclidean descent rather than explicit parameter retraction.
- We first compute tangent momentum through parameter-space manifold projection, then apply a momentum-space manifold constraint to keep the update on the manifold.
- We use a rotating Oblique manifold rather than a static Oblique manifold, alternating across the two matrix dimensions (row or column) at every optimization step. This enables a special neuron-wise manifold optimization because each row or column corresponds to a neuron's input or output weights.

In comparison with SGD with momentum, Mano adds two row- or column-wise normalizations and one tangent-space projection, introduces no additional optimizer state, and requires no problem-specific geometric assumptions. Its implementation follows directly from standard SGD with momentum. The memory overhead is comparable to SGD with momentum or Muon, using one momentum state rather than the two moment states of Adam-based optimizers. The manifold normalization is also inexpensive because no `MatMul` operation is involved. We next discuss the rotating scheme and consistent update scale.

Rotational manifold scheme. The iterative procedure of alternating row and column normalization is related to the Sinkhorn–Knopp iteration, which converges under suitable conditions to a doubly stochastic matrix (Knight, 2008). The set of doubly stochastic matrices has also been studied in manifold optimization (Douik & Hassibi, 2019). Mano does not run this normalization to convergence or constrain the parameters to that set. Instead, it applies one orientation at a time and constrains only the tangent update to the active Oblique manifold. The alternating scheme applies neuron-wise manifold optimization to both the input and output weights of neurons.

Consistent update RMS. Each nonzero active block of $\hat{V}_t$ has unit ℓ_2 norm, so its update RMS is $1/\sqrt{d_{k_t}}$. The factor $s\sqrt{d_{k_t}}$ therefore gives update RMS s for either orientation (and at most s when zero blocks occur). The factor controls scale consistently across orientations, but does not require different optimizers to use the same learning rate; Section 4 tunes learning rates separately.

Table 1: Computational cost of Newton–Schulz iteration (NS, T=5) used by Muon versus the manifold normalization enforced by Mano, evaluated on Attention and MLP weight matrices from LLaMA-1B, -7B, and -70B in `BFloat16`. Reported time is averaged over 1000 PyTorch runs and peak GPU memory is measured via TORCH.CUDA. These measurements empirically confirm Mano's theoretically lower computational overhead relative to Muon.

Model	Module	Shape	Newton–Schulz		Mano	
			Time (ms)	Memory (MB)	Time (ms)	Memory (MB)
LLaMA-1B	Attention	$(2048, 2048)$	2.01	64.1	**0.14**	**56.0**
	MLP	$(2048, 5461)$	4.68	119.3	**0.17**	**87.3**
LLaMA-7B	Attention	$(4096, 4096)$	14.83	224.0	**0.34**	**192.0**
	MLP	$(4096, 11008)$	30.22	472.0	**1.45**	**344.0**
LLaMA-70B	Attention	$(8192, 8192)$	110.79	896.0	**2.19**	**512.0**
	MLP	$(8192, 28672)$	184.33	1536.0	**4.35**	**1024.0**

3.3 Theoretical Analysis

Computational Overhead. Relative to SGD with momentum, Mano adds two dimension-wise normalizations and one tangent projection. For $W \in \mathbb{R}^{m\times n}$, the normalizations require $3mn$ FLOPs each and the projection at most $5mn$ FLOPs, for at most $11mn$ additional FLOPs without `MatMul`. For a baseline of approximately $6mnB$ FLOPs to process B token representations through the corresponding linear layer in the forward and backward passes, Mano's optimizer-side FLOP ratio is at most $11/(6B)$. This ratio is independent of the matrix dimensions m and n and decreases as the token batch B grows. Meanwhile, Muon's Newton–Schulz update has a dimension-dependent overhead of approximately $5m/B$ for a square $m \times m$ matrix (Jordan et al., 2024). Thus, Mano replaces Muon's matrix multiplications with linear-cost reductions and element-wise operations. On the other hand, Mano stores one momentum state, matching momentum-SGD and Muon, and uses one fewer optimizer state than AdamW (Kingma & Ba, 2015; Jordan et al., 2024). As an isolated-kernel view of the same effect, Table 1 reports per-call time and peak memory of Newton–Schulz and Mano's manifold normalization on representative LLaMA Attention and MLP matrices. Section 4 measures the end-to-end step-time overhead in pretraining LLaMA models.

Convergence Analysis. We analyze a simplified row-wise update without momentum, weight decay, or orientation rotation. The column-wise result follows by transposition.

Theorem 1 (Convergence of Simplified Mano)**.** *Let $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ be lower bounded by $f_{\inf}$ and L-smooth under the Frobenius norm. Let every row of W_0 be nonzero, and define*

$$G_t := \nabla f(W_t), \quad U_t := V_{\mathrm{row}}(W_t)[G_t], \quad \widehat{U}_t := U_t \oslash \|U_t\|_{2,\mathrm{row}}, \quad W_{t+1} := W_t - \eta\sqrt{n}\,\widehat{U}_t. \tag{6}$$

Assume $\sum_{i=1}^{m} \|U_{t,i,:}\|_2 \geq \gamma \|G_t\|_F$ for every $t \in \{0, \ldots, T\}$, where $\gamma \in (0, 1]$ is independent of T. For $\eta = C/\sqrt{T+1}$ with $C > 0$,

$$\min_{0\leq t\leq T} \|\nabla f(W_t)\|_F \leq \frac{1}{\sqrt{T+1}} \left(\frac{f(W_0) - f_{\inf}}{\gamma C\sqrt{n}} + \frac{LCm\sqrt{n}}{2\gamma} \right). \tag{7}$$

Theorem 1 gives first-order stationarity at a rate of $O(T^{-1/2})$. This theorem assumes uniform aggregate alignment between the retained tangent component and the full gradient, transferring Mano's optimization of the tangent gradient to the full gradient. The exact degeneracy is highly constrained, with orientation rotation reducing the common blind space to the one-dimensional space $\mathrm{span}\{W\}$. Appendix C provides the proof.

A Fisher-Information Perspective. For an ideal normalized block, uniform scaling leaves the output unchanged, $\mathrm{RMSNorm}(cWx) = \mathrm{RMSNorm}(Wx)$ for $c > 0$, so such changes are insensitive to model predictions. We use the Fisher-Rayleigh quotient to measure how strongly a parameter direction affects model predictions per unit update magnitude. For an active orientation k, decompose a nonzero momentum as $M = M_{\parallel} + M_{\perp}$, where $M_{\parallel}$ changes only active row or column amplitudes

and $M_\perp := V_k(W)[M]$ is Frobenius-orthogonal to it. Let $\alpha := \left\|M_\parallel\right\|_F^2 / \|M\|_F^2$, define the Fisher-Rayleigh quotient $R_F(W, \Delta) := \langle \Delta, F(W)\Delta\rangle / \|\Delta\|_F^2$, and let $\chi_F := \langle M_\parallel, F(W)M_\perp\rangle / \|M\|_F^2$, where $\langle A, F(W)B\rangle := \mathrm{vec}(A)^\top F(W)\,\mathrm{vec}(B)$.

Theorem 2 (Amplitude Removal and Conditional Fisher-Information Gain)**.** *Assume the active rows or columns of W are nonzero, $F(W) \succeq 0$, and $0 < \alpha < 1$. For the row orientation, let $M_\parallel = \Lambda_{\mathrm{row}} W$; for the column orientation, let $M_\parallel = W\Lambda_{\mathrm{col}}$, where Λ_{row} and Λ_{col} are diagonal. Then Mano's active projector satisfies*

$$V_k(W)[M_\parallel] = 0, \quad V_k(W)[M] = M_\perp, \quad \|M_\perp\|_F^2 = (1-\alpha)\,\|M\|_F^2\,, \tag{8}$$

and its change in Fisher-Rayleigh quotient is

$$\begin{aligned} R_F(W, M_\perp) - R_F(W, M) &= \alpha\big[R_F(W, M_\perp) - R_F(W, M_\parallel)\big] - 2\chi_F, \\ R_F(W, M_\perp) - R_F(W, M) > 0 \quad &\Longleftrightarrow \quad \alpha\big[R_F(W, M_\perp) - R_F(W, M_\parallel)\big] > 2\chi_F. \end{aligned} \tag{9}$$

Equation equation 8 denotes the geometric action of the projector: it exactly removes the active amplitude-aligned component without computing the Fisher matrix and retains its Frobenius-orthogonal component. Equation equation 9 gives the exact condition, including its dependence on the Fisher coupling χ_F, under which tangent projection yields a Fisher-information gain as measured by an increase in the Fisher-Rayleigh quotient; this is a condition rather than a guarantee. Overall, under the ideal normalized-block and regularity assumptions in Appendix D, these results suggest that Mano can suppress uniform or sufficiently near-uniform amplitude changes with low Fisher sensitivity and retain a more Fisher-sensitive tangent component when the stated condition holds.

4 Experiments

In this section, we empirically study Mano through LLaMA pretraining on C4 and compare it with the baseline optimizers AdamW and Muon in terms of validation loss, learning dynamics, and wall-clock efficiency.

4.1 Experiment Setup

We evaluate Mano on LLaMA models (Touvron et al., 2023) with 350M, 780M, and 1.3B parameters. Following Chinchilla-optimal scaling (Hoffmann et al., 2022), the models are trained on C4 (Raffel et al., 2020) for 7.0B, 15.6B, and 26.8B tokens, respectively. The 350M model uses batches of 0.26M tokens, while the 780M and 1.3B models use batches of 1.0M tokens. AdamW uses $(\beta_1, \beta_2) = (0.9, 0.95)$; Muon and Mano use a momentum coefficient of $\mu = 0.95$. All optimizers use cosine learning-rate decay with 1000 warmup steps and weight decay $\lambda_{\mathrm{wd}} = 0.1$. Additional configuration and timing-hardware details are in Appendix A.

Learning-rate tuning. We conduct optimizer-specific learning-rate sweeps for Mano, AdamW, and Muon on LLaMA-350M and 780M. For LLaMA-1.3B, we select each optimizer's learning rate based on the smaller-scale sweeps and use it for final validation. The sweep curves and 1.3B selections are reported in Appendix A.2 and Fig. 4.

Experiment results. Table 2 and Fig. 1 compare validation loss and sample efficiency across model scales. Mano achieves the lowest final validation loss at all three scales, improving over Muon by 0.0164, 0.0193, and 0.0140, respectively. Although Mano's validation loss decreases more slowly during early training, it improves more strongly in later stages and ultimately surpasses both AdamW and Muon, indicating distinct optimization dynamics. This trend also suggests that Mano may perform better in an overtraining regime. Additional projection and normalization axis ablations are reported in Appendix A.3.

Fig. 2 extends the evaluation to Qwen3 on C4 (Raffel et al., 2020) and the Pile (Gao et al., 2020), using shared per-corpus learning rates of 1×10^{-3} and 3×10^{-4}, respectively, following the learning-rate transfer strategy of Liu et al. (2025). The Qwen3-0.6B C4 run uses a Chinchilla-optimal token budget, while the Qwen3-0.6B and 1.7B Pile runs use fixed budgets. Mano attains the lowest final loss in every setting, outperforming AdamW and Muon on C4 and Muon on the Pile.

Table 2: Validation loss of models trained on C4 under Chinchilla-optimal token budgets. Mano improves over AdamW and Muon on all evaluated model architectures and scales.

Model	Tokens	Validation Loss			Mano − Muon
		AdamW	Muon	Mano	
LLaMA-350M	7.0B	2.7795	2.7551	**2.7387**	−0.0164
LLaMA-780M	15.6B	2.5403	2.5153	**2.4960**	−0.0193
LLaMA-1.3B	26.8B	2.3143	2.2878	**2.2738**	−0.0140
Qwen3-0.6B	12.0B	2.8192	2.7878	**2.7663**	−0.0215

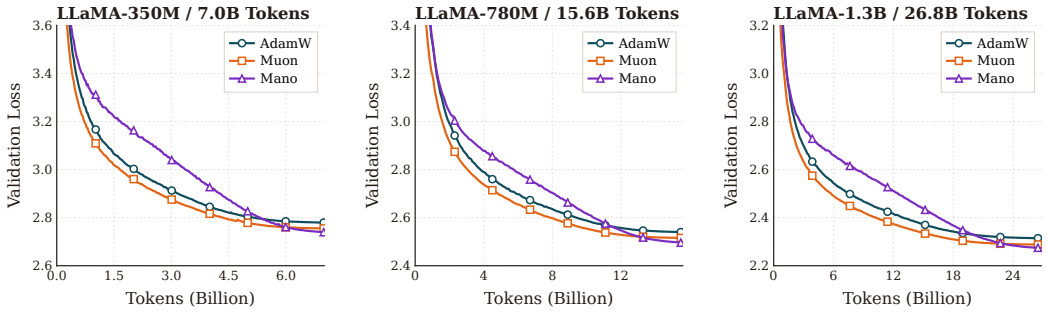


Figure 1: Validation-loss trajectories for the completed LLaMA-350M, 780M, and 1.3B runs on C4 using the selected optimizer-specific learning rates. Mano trails the baselines early but achieves the lowest final validation loss at all three scales, demonstrating greater late-stage improvement.

4.2 Learning Dynamics

In this subsection, we examine the learning dynamics of Mano and compare it with AdamW and Muon from multiple perspectives.

Gradient signal-to-noise ratio. Figure 3 reports the gradient signal-to-noise ratio (SNR), measured as gradient norm divided by residual energy, and shows that Mano maintains a higher ratio than Muon over substantial portions of pretraining. Higher SNR indicates a more stable gradient signal relative to noise. This observation is consistent with the Fisher-information perspective in Section 3.3: tangent projection may remove weakly informative amplitude components, which could contribute to Mano's stronger late-stage improvement in Fig. 1.

Wall-clock efficiency. We previously derived the theoretical FLOP overhead of Mano and showed that its manifold normalization uses no `MatMul` operations. To assess practical computational efficiency, we report complete training-step time for Mano and the baseline optimizers. Table 3 measures complete training-step time rather than isolated optimizer kernels. Across three model scales, Mano adds 0.2%–0.8% overhead to AdamW, while Muon adds 4.1%–11.3%. The absolute times are setup-dependent, but the within-row comparison supports the low-overhead prediction in Section 3.3.

Table 3: Training-step time on LLaMA models. The 350M and 780M runs use 8× NVIDIA H20 GPUs; the 1.3B timing run uses 16× NVIDIA H20 GPUs. Wall-clock overhead is relative to AdamW in the same setup and is computed from unrounded measurements; displayed times are rounded to milliseconds.

Model	AdamW	Muon		Mano	
		Step Time	Wall-clock Overhead	Step Time	Wall-clock Overhead
LLaMA-350M	**0.918 s**	1.021 s	+11.3%	0.925 s	+0.8%
LLaMA-780M	**6.335 s**	6.596 s	+4.1%	6.358 s	+0.4%
LLaMA-1.3B	**8.415 s**	9.253 s	+10.0%	8.431 s	+0.2%

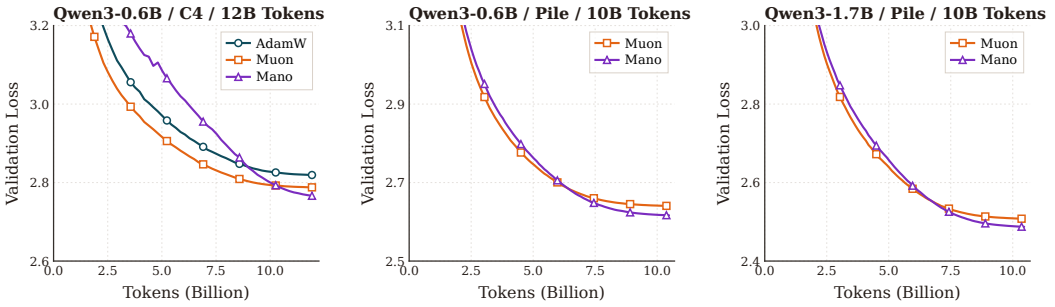


Figure 2: Validation-loss trajectories for Qwen3-0.6B on C4 and Qwen3-0.6B and 1.7B on the Pile. Mano achieves a lower final validation loss than Muon in all three settings. AdamW was skipped in the Pile experiments due to computational constraints.

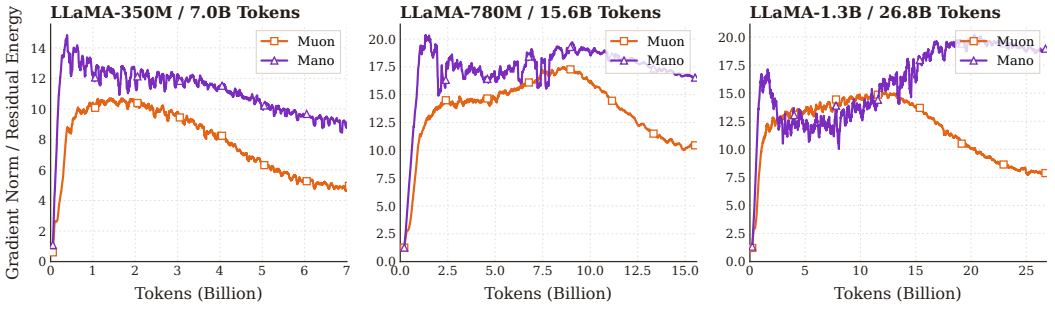


Figure 3: Gradient signal-to-noise ratio, measured as gradient norm divided by residual energy, for LLaMA-350M, 780M, and 1.3B models trained on C4 with Muon and Mano. The horizontal axis reports training tokens. Mano maintains a higher SNR than Muon over substantial portions of training, indicating a more stable gradient signal relative to residual noise.

The experimental results demonstrate, within the evaluated settings, Mano's stronger late-stage sample efficiency under fixed token budgets and near-AdamW computational efficiency. The gradient-SNR analysis offers an empirical perspective on its distinct learning dynamics and suggests that future optimizer designs may benefit from jointly considering update geometry, signal quality, and systems overhead.

5 Related Work

Optimizers for LLM pretraining. Adam-based optimizers remain the most widely used optimizers in deep learning, including both pretraining and fine-tuning decoder-only Transformers (Zhao et al., 2024b). Their popularity stems from a simple design, flexible per-parameter adaptive learning rates, and robust performance across diverse domains. However, the first- and second-moment estimates of AdamW consume twice the optimizer-state memory of a single model-weight or gradient copy, resulting in substantial memory overhead at LLM scale. Several approaches have therefore sought more memory-efficient optimizers. Adam-mini uses block-wise learning rates based on Hessian partitions (Zhang et al., 2024b; Wang et al., 2025); Lion uses momentum-sign updates to eliminate the second moment (Chen et al., 2023); and Cautious-Adam and Cautious-Lion apply gradient-aligned selective updates (Liang et al., 2024). Other notable methods include Swan, MARS, Sophia, Shampoo, and SOAP (Ma et al., 2024; Yuan et al., 2024; Liu et al., 2023; Gupta et al., 2018; Vyas et al., 2024).

Many adaptive methods can also be interpreted as structured approximations to the empirical Fisher matrix (Gong et al., 2025). Adam uses a diagonal approximation; K-FAC and EKFAC use Kronecker-factored statistics (Martens & Grosse, 2015; George et al., 2018); Shampoo maintains tensor-wise Kronecker preconditioners (Gupta et al., 2018; Morwani et al., 2024); and SOAP ap-

plies Adam-style scaling in a Shampoo-derived eigenbasis (Vyas et al., 2024). Low-rank alternatives include GaLore and RACS/Alice (Zhao et al., 2024a; Gong et al., 2025).

Another promising line of research focuses on matrix-based spectral preconditioning. Muon uses Newton–Schulz iterations to approximate spectral normalization of the momentum (Jordan et al., 2024). The resulting matrix-sign approximation normalizes the magnitudes of spectral directions, including low-magnitude directions that may be important for model generalization. Subsequent studies have extended Muon to larger LLM training regimes (Liu et al., 2025; Shah et al., 2025; Team et al., 2025; Zeng et al., 2025). Muon is memory efficient and lifts weak spectral directions, but its matrix multiplications add computational overhead and the matrix-sign limit discards relative singular-value magnitudes. Mano instead uses dimension-wise tangent projection and normalization without matrix multiplication in the optimizer transformation. Benchmarking studies find that matrix-based optimizers often outperform scalar counterparts, although no optimizer dominates every tested setting (Schlotthauer et al., 2025; Wen et al., 2025b; Semenov et al., 2025). We therefore treat the experiments as hypothesis testing in the reported regimes rather than as evidence of universal optimizer dominance.

Manifold and norm-constrained optimization. Geometric optimization methods exploit intrinsic geometric structures of an objective and have been applied in many areas, including deep learning (Fei et al., 2025). For objectives defined on a Riemannian manifold with a differentiable structure and smooth inner product, manifold methods seek solutions through tangent gradients and retractions. Geometrically aware counterparts of conventional Euclidean methods include Riemannian SGD, SGD with momentum, RMSProp, Adam, AdaGrad, and AMSGrad (Bonnabel, 2013; Zhang et al., 2016; Bécigneul & Ganea, 2018; Roy et al., 2018). A broad range of architecture-dependent manifold techniques has been proposed for CNNs (Ozay & Okatani, 2016; Wang et al., 2020), recurrent networks (Arjovsky et al., 2016; Wisdom et al., 2016; Huang et al., 2018; Jing et al., 2019), graph neural networks (Zhu et al., 2020; Liu et al., 2021; de Ocáriz Borde et al., 2023), and other deep-learning settings (Zhang et al., 2018; Chaudhry et al., 2020).

Although manifold optimization is well established in other deep-learning domains, it has received comparatively little attention in LLM training. Recent work studies parameter-efficient low-rank training through Riemannian manifolds (Jiang et al.; Zhang et al., 2024a; Mo et al., 2025; Park et al., 2025), with related geometric strategies for gradient tracking (Rajabi & Rambhatla, 2024; Rajabi et al., 2025), representation regularization (Zhang & Dong, 2025; Kingswell et al., 2025; Wren et al., 2025), and parameter pruning (Liu et al., 2024). Mano differs from conventional manifold methods because it does not retract the model weights; the alternating Oblique geometry acts only on the optimizer update.

6 Conclusion

Limitations. First, although we find that training LLMs with neuron-wise hyperspherical geometry is effective even with simple momentum, we lack a theoretical explanation for why this subspace or manifold is effective. We believe studying neuron-wise learning dynamics in LLMs is a promising direction for future theoretical and empirical work. Second, our current convergence analysis holds for a simplified version of Mano without momentum or orientation rotation. Extending the theory to cover these dynamics and broader optimization regimes remains another future direction.

Summary. To the best of our knowledge, Mano is the first work to successfully reformulate manifold optimization methods for LLM training. We find that simple momentum projection under a rotational Oblique geometry corresponding to a product of neuron-wise hyperspheres can outperform modern optimizers that perform spectral preconditioning or second-moment estimation. Our Fisher-information analysis shows that Mano's tangent projection removes the active amplitude-aligned component and characterizes when the retained direction has a higher Fisher-Rayleigh quotient per unit magnitude. In our experiments, Mano not only outperforms AdamW and Muon across the evaluated model scales but also incurs lower computational overhead than Muon and lower optimizer-state memory consumption than AdamW. Our work highlights the overlooked value of utilizing neuron-aware manifold techniques for training LLMs in conjunction with modern optimization strategies.

Reproducibility Statement

We support reproducibility by specifying the complete Mano update in Algorithm 1. Section 4 and Appendix A report the training configurations, model architectures, learning-rate selection procedure, token budgets, and timing hardware. Appendix C provides the convergence proofs, while Appendix D provides the full derivations for the Fisher-information analysis. Additional implementation details and ablation settings are also provided in the appendix.

Generative AI Usage Statement

Generative AI tools (LLMs) were used to polish the writing, assist with mathematical derivations, refine the presentation, suggest revisions, and review code. The authors verified all AI-assisted content and take full responsibility for the paper's claims, derivations, and implementation.

Ethics Statement

This work concerns optimization methodology and theory, with experiments conducted solely to evaluate optimizer behavior. It does not involve human subjects or publish models that may produce harmful or sensitive content. We therefore do not identify direct ethical risks specific to this study.

References

P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.

Guozhong An. The effects of adding noise during backpropagation training on a generalization performance. *Neural computation*, 8(3):643–674, 1996.

Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In *International conference on machine learning*, pp. 1120–1128. PMLR, 2016.

Gary Bécigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. *arXiv preprint arXiv:1810.00760*, 2018.

Silvere Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

Arslan Chaudhry, Naeemullah Khan, Puneet Dokania, and Philip Torr. Continual learning in low-rank orthogonal subspaces. *Advances in Neural Information Processing Systems*, 33:9900–9911, 2020.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, et al. Symbolic discovery of optimization algorithms. *Advances in neural information processing systems*, 36:49205–49233, 2023.

Md Naseef Ur Rahman Chowdhury, Ahshanul Haque, and Hamdy Soliman. The hidden cost of ai: Unraveling the power-hungry nature of large language models. 2025.

Haitz Sáez de Ocáriz Borde, Anees Kazi, Federico Barbero, and Pietro Lio. Latent graph inference using product manifolds. In *The eleventh international conference on learning representations*, 2023.

Ahmed Douik and Babak Hassibi. Manifold optimization over the set of doubly stochastic matrices: A second-order geometry. *IEEE Transactions on Signal Processing*, 67(22):5761–5774, 2019.

Yanhong Fei, Yingjie Liu, Chentao Jia, Zhengyu Li, Xian Wei, and Mingsong Chen. A survey of geometric optimization for deep learning: from euclidean space to riemannian manifold. *ACM Computing Surveys*, 57(5):1–37, 2025.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Mingqi Gao, Xinyu Hu, Xunjian Yin, Jie Ruan, Xiao Pu, and Xiaojun Wan. Llm-based nlg evaluation: Current status and challenges. *Computational Linguistics*, pp. 1–27, 2025.

Thomas George, César Laurent, Xavier Bouthillier, Nicolas Ballas, and Pascal Vincent. Fast approximate natural gradient descent in a kronecker factored eigenbasis. *Advances in neural information processing systems*, 31, 2018.

Wenbo Gong, Meyer Scetbon, Chao Ma, and Edward Meeds. Towards efficient optimizer design for llm via structured fisher approximation with a low-rank extension. *arXiv preprint arXiv:2502.07752*, 2025.

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

Alexander Hägele, Alejandro Hernández-Cano, Atli Kosson, and Martin Jaggi. Improving neural network training by decoupling the magnitude and direction of weight vectors. *arXiv preprint arXiv:2606.25971*, 2026.

Jeff Z HaoChen, Colin Wei, Jason Lee, and Tengyu Ma. Shape matters: Understanding the implicit bias of the noise covariance. In *Conference on Learning Theory*, pp. 2315–2357. PMLR, 2021.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Jiang Hu, Xin Liu, Zai-Wen Wen, and Ya-Xiang Yuan. A brief introduction to manifold optimization. *Journal of the Operations Research Society of China*, 8(2):199–248, 2020.

Wenqing Hu, Chris Junchi Li, Lei Li, and Jian-Guo Liu. On the diffusion approximation of nonconvex stochastic gradient descent. *Annals of Mathematical Sciences and Applications*, 4(1):3–32, 2019.

Lei Huang, Xianglong Liu, Bo Lang, Adams Yu, Yongliang Wang, and Bo Li. Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Stanislaw Jastrzkebski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

Xiaowen Jiang, Xun Wang, and Sebastian U Stich. Loram: Low-rank adaptation of large language models on manifold. In *Sparsity in LLMs (SLLM): Deep Dive into Mixture of Experts, Quantization, Hardware, and Inference*.

Li Jing, Caglar Gulcehre, John Peurifoy, Yichen Shen, Max Tegmark, Marin Soljacic, and Yoshua Bengio. Gated orthogonal recurrent units: On learning to forget. *Neural computation*, 31(4): 765–783, 2019.

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

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diego, CA, 2015.

Jonathan Kingswell, Leo Whitstable, Oliver Troughton, Kent Blumberg, and Arabella Sutherland. Sequential manifold regularization for large language model contextual stability. 2025.

Philip A Knight. The sinkhorn–knopp algorithm: convergence and applications. *SIAM Journal on Matrix Analysis and Applications*, 30(1):261–275, 2008.

Pranjal Kumar. Large language models (llms): survey, technical frameworks, and future challenges. *Artificial Intelligence Review*, 57(10):260, 2024.

Frederik Kunstner, Alan Milligan, Robin Yadav, Mark Schmidt, and Alberto Bietti. Heavy-tailed class imbalance and why adam outperforms gradient descent on language models. *Advances in Neural Information Processing Systems*, 37:30106–30148, 2024.

Qianxiao Li, Cheng Tai, et al. Stochastic modified equations and adaptive stochastic gradient algorithms. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2101–2110. JMLR. org, 2017.

Zhiyuan Li, Sadhika Malladi, and Sanjeev Arora. On the validity of modeling sgd with stochastic differential equations (sdes). *arXiv preprint arXiv:2102.12470*, 2021.

Kaizhao Liang, Lizhang Chen, Bo Liu, and Qiang Liu. Cautious optimizers: Improving training with one line of code. *arXiv preprint arXiv:2411.16085*, 2024.

Deyuan Liu, Zhanyue Qin, Hairu Wang, Zhao Yang, Zecheng Wang, Fangying Rong, Qingbin Liu, Yanchao Hao, Bo Li, Xi Chen, et al. Pruning via merging: Compressing llms via manifold alignment based layer merging. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17817–17829, 2024.

Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. *arXiv preprint arXiv:2305.14342*, 2023.

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.

Weifeng Liu, Sichao Fu, Yicong Zhou, Zheng-Jun Zha, and Liqiang Nie. Human activity recognition by manifold regularization based dynamic graph convolutional networks. *Neurocomputing*, 444: 217–225, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Ilya Loshchilov, Cheng-Ping Hsieh, Simeng Sun, and Boris Ginsburg. ngpt: Normalized transformer with representation learning on the hypersphere. In *International Conference on Learning Representations*, volume 2025, pp. 74014–74038, 2025.

Chao Ma, Wenbo Gong, Meyer Scetbon, and Edward Meeds. Swan: Sgd with normalization and whitening enables stateless llm training. *arXiv preprint arXiv:2412.13148*, 2024.

James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pp. 2408–2417. PMLR, 2015.

Zhanfeng Mo, Long-Kai Huang, and Sinno Jialin Pan. Parameter and memory efficient pretraining via low-rank riemannian optimization. In *The Thirteenth International Conference on Learning Representations*, 2025.

Depen Morwani, Itai Shapira, Nikhil Vyas, Eran Malach, Sham Kakade, and Lucas Janson. A new perspective on shampoo's preconditioner. *arXiv preprint arXiv:2406.17748*, 2024.

Arvind Neelakantan, Luke Vilnis, Quoc V Le, Ilya Sutskever, Lukasz Kaiser, Karol Kurach, and James Martens. Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*, 2015.

Thanh Huy Nguyen, Umut Simsekli, Mert Gurbuzbalaban, and Gaël Richard. First exit time analysis of stochastic gradient descent under heavy-tailed gradient noise. In *Advances in Neural Information Processing Systems*, pp. 273–283, 2019.

Mete Ozay and Takayuki Okatani. Optimization on submanifolds of convolution kernels in cnns. *arXiv preprint arXiv:1610.07008*, 2016.

Juneyoung Park, Minjae Kang, Seongbae Lee, Haegang Lee, Seongwan Kim, and Jaeho Lee. Riemannian optimization for lora on the stiefel manifold. *arXiv preprint arXiv:2508.17901*, 2025.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Sahar Rajabi and Sirisha Rambhatla. Optimizing fine-tuning efficiency: Gradient subspace tracking on grassmann manifolds for large language models. In *NeurIPS 2024 Workshop on Mathematics of Modern Machine Learning*, 2024.

Sahar Rajabi, Nayeema Nonta, and Sirisha Rambhatla. Subtrack++: Gradient subspace tracking for scalable llm training. *arXiv preprint arXiv:2502.01586*, 2025.

Paul Harry Roberts and Harold Douglas Ursell. Random walk on a sphere and on a riemannian manifold. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 252(1012):317–356, 1960.

Soumava Kumar Roy, Zakaria Mhammedi, and Mehrtash Harandi. Geometry aware constrained optimization techniques for deep learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4460–4469, 2018.

Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems*, 29, 2016.

Siddharth Samsi, Dan Zhao, Joseph McDonald, Baolin Li, Adam Michaleas, Michael Jones, William Bergeron, Jeremy Kepner, Devesh Tiwari, and Vijay Gadepally. From words to watts: Benchmarking the energy costs of large language model inference. In *2023 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–9. IEEE, 2023.

Joel Schlotthauer, Christian Kroos, Chris Hinze, Viktor Hangya, Luzian Hahn, and Fabian Küch. Pre-training llms on a budget: A comparison of three optimizers. *arXiv preprint arXiv:2507.08472*, 2025.

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

Ishaan Shah, Anthony M Polloreno, Karl Stratos, Philip Monk, Adarsh Chaluvaraju, Andrew Hojel, Andrew Ma, Anil Thomas, Ashish Tanwer, Darsh J Shah, et al. Practical efficiency of muon for pretraining. *arXiv preprint arXiv:2505.02222*, 2025.

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

Weijie Su. Isotropic curvature model for understanding deep learning optimization: Is gradient orthogonalization optimal? *arXiv preprint arXiv:2511.00674*, 2025.

Qian-Yuan Tang, Yufei Gu, Yunfeng Cai, Mingming Sun, Ping Li, Zeke Xie, et al. Investigating the overlooked hessian structure: From cnns to llms. In *Forty-second International Conference on Machine Learning*, 2025.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Mujin Kwun, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. Soap: Improving and stabilizing shampoo using adam. *arXiv preprint arXiv:2409.11321*, 2024.

Jiayun Wang, Yubei Chen, Rudrasis Chakraborty, and Stella X Yu. Orthogonal convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11505–11515, 2020.

Jinbo Wang, Mingze Wang, Zhanpeng Zhou, Junchi Yan, Lei Wu, et al. The sharpness disparity principle in transformers for accelerating language model pre-training. *arXiv preprint arXiv:2502.19002*, 2025.

Kaiyue Wen, Xingyu Dang, Kaifeng Lyu, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them ii: From weight decay to hyperball optimization, 11 2025a. URL https://whenwen.github.io/wd_blog/public/hyperball-part-1.html.

Kaiyue Wen, David Hall, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them. *arXiv preprint arXiv:2509.02046*, 2025b.

Scott Wisdom, Thomas Powers, John Hershey, Jonathan Le Roux, and Les Atlas. Full-capacity unitary recurrent neural networks. *Advances in neural information processing systems*, 29, 2016.

Alistair Wren, Beatrice Loxley, Hamish Cadwallader, Simon Beckwith, Fabian Pargeter, and James Blades. Contextual subspace manifold projection for structural refinement of large language model representations. *arXiv preprint arXiv:2502.08026*, 2025.

Lei Wu, Chao Ma, and E Weinan. How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective. In *Advances in Neural Information Processing Systems*, pp. 8279–8288, 2018.

Tian Xie, Haoming Luo, Haoyu Tang, Yiwen Hu, Jason Klein Liu, Qingnan Ren, Yang Wang, Wayne Xin Zhao, Rui Yan, Bing Su, et al. Controlled llm training on spectral sphere. *arXiv preprint arXiv:2601.08393*, 2026.

Zeke Xie, Issei Sato, and Masashi Sugiyama. A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=wXgk_iCiYGo.

Zeke Xie, Fengxiang He, Shaopeng Fu, Issei Sato, Dacheng Tao, and Masashi Sugiyama. Artificial neural variability for deep learning: On overfitting, noise memorization, and catastrophic forgetting. *Neural Computation*, 33(8), 2021a.

Zeke Xie, Li Yuan, Zhanxing Zhu, and Masashi Sugiyama. Positive-negative momentum: Manipulating stochastic gradient noise to improve generalization. In *International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 11448–11458. PMLR, 18–24 Jul 2021b.

Zeke Xie, Qianyuan Tang, Mingming Sun, and Ping Li. On the overlooked structure of stochastic gradients. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Pan Xu, Jinghui Chen, Difan Zou, and Quanquan Gu. Global convergence of langevin dynamics based algorithms for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pp. 3122–3133, 2018.

Greg Yang, James B Simon, and Jeremy Bernstein. A spectral condition for feature learning. *arXiv preprint arXiv:2310.17813*, 2023.

Huizhuo Yuan, Yifeng Liu, Shuang Wu, Xun Zhou, and Quanquan Gu. Mars: Unleashing the power of variance reduction for training large models. *arXiv preprint arXiv:2411.10438*, 2024.

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*, 2025.

Hongyi Zhang, Sashank J Reddi, and Suvrit Sra. Riemannian svrg: Fast stochastic optimization on riemannian manifolds. *Advances in Neural Information Processing Systems*, 29, 2016.

Tong Zhang, Wenming Zheng, Zhen Cui, and Chaolong Li. Deep manifold-to-manifold transforming network. In *2018 25th IEEE international conference on image processing (ICIP)*, pp. 4098–4102. IEEE, 2018.

Yuan Zhang, Jiang Hu, Jiaxi Cui, Lin Lin, Zaiwen Wen, and Quanzheng Li. Retraction-free optimization over the stiefel manifold with application to the lora fine-tuning. 2024a.

Yukun Zhang and Qi Dong. Multi-scale manifold alignment: A unified framework for enhanced explainability of large language models. *arXiv preprint arXiv:2505.20333*, 2025.

Yushun Zhang, Congliang Chen, Ziniu Li, Tian Ding, Chenwei Wu, Diederik P Kingma, Yinyu Ye, Zhi-Quan Luo, and Ruoyu Sun. Adam-mini: Use fewer learning rates to gain more. *arXiv preprint arXiv:2406.16793*, 2024b.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient llm training by gradient low-rank projection. *arXiv preprint arXiv:2403.03507*, 2024a.

Rosie Zhao, Depen Morwani, David Brandfonbrener, Nikhil Vyas, and Sham Kakade. Deconstructing what makes a good optimizer for language models. *arXiv preprint arXiv:2407.07972*, 2024b.

Mo Zhou, Tianyi Liu, Yan Li, Dachao Lin, Enlu Zhou, and Tuo Zhao. Toward understanding the importance of noise in training neural networks. In *International Conference on Machine Learning*, 2019.

Shichao Zhu, Shirui Pan, Chuan Zhou, Jia Wu, Yanan Cao, and Bin Wang. Graph geometry interaction learning. *Advances in Neural Information Processing Systems*, 33:7548–7558, 2020.

Zhanxing Zhu, Jingfeng Wu, Bing Yu, Lei Wu, and Jinwen Ma. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. In *ICML*, pp. 7654–7663, 2019.

A Details for Reproducibility

A.1 Training Configurations

Table 4 consolidates the reported training configurations. The C4 corpus is used for the LLaMA and Qwen3-0.6B C4 final-loss experiments; additional Qwen3 experiments use the Pile. The 350M model uses batches of 0.26M tokens; the 780M and 1.3B models use batches of 1.0M tokens. The corresponding Chinchilla-optimal budgets are 7.0B, 15.6B, and 26.8B tokens. All methods use cosine learning-rate decay with 1000 warmup steps and weight decay 0.1. AdamW uses $(\beta_1, \beta_2) = (0.9, 0.95)$, whereas Muon and Mano use a momentum coefficient of $\mu = 0.95$.

Table 4: Configurations for the final-loss experiments. Learning rates are selected separately for each optimizer from the sweeps in Fig. 4.

Model	Tokens	Batch	Seq. Length	Iterations	Warmup	AdamW LR	Muon LR	Mano LR
LLaMA-350M	7.0B	0.26M	512	27000	1000	1×10^{-3}	1.5×10^{-3}	8×10^{-4}
LLaMA-780M	15.6B	1.0M	1024	14877	1000	1.5×10^{-3}	2×10^{-3}	8×10^{-4}
LLaMA-1.3B	26.8B	1.0M	1024	25558	1000	1×10^{-3}	1.5×10^{-3}	8×10^{-4}

Table 5: Architecture details from the model configuration files. The configured maximum sequence length is an architectural limit; the sequence length used for each final-loss run is reported separately in Table 4.

Model	Layers	Hidden Size	FFN Size	Attention Heads	Vocabulary	Activation
LLaMA-350M	24	1024	2736	16	32000	SiLU
LLaMA-780M	24	1536	4096	24	32000	SiLU
LLaMA-1.3B	24	2048	5461	32	32000	SiLU

The timing measurements in Table 3 include complete training steps. The 350M and 780M timing runs use $8\times$ NVIDIA H20 GPUs, and the 1.3B timing run uses $16\times$ NVIDIA H20 GPUs. Because model FLOPs utilization changes with model scale, GPU count, and kernel efficiency, absolute timing values should be compared only within each row.

A.2 Learning-Rate Tuning

Figure 4 reports the learning-rate sweeps and selected rates described in Section 4.

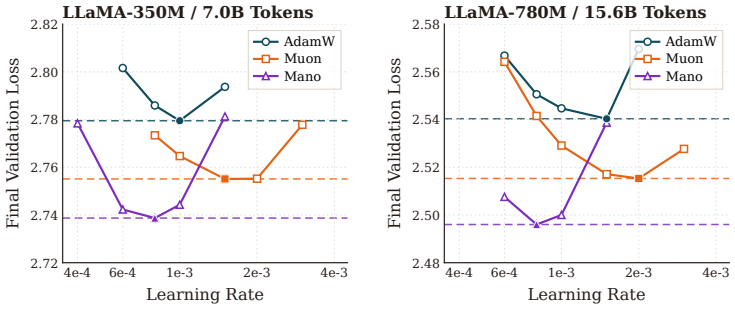
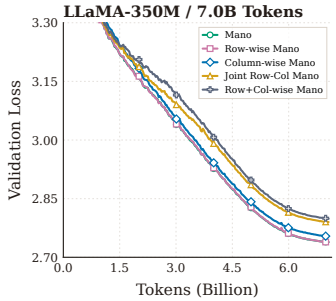


Figure 4: Optimizer-specific learning-rate sweeps for LLaMA-350M and 780M on C4 under Chinchilla-optimal token budgets.

A.3 Ablation Studies

We ablate the active axes for projection and normalization on LLaMA-350M at a learning rate of 8×10^{-4} using full-strength projection and normalization. Mano and Row-wise Mano attain similar final losses, whereas the column-only and dual-axis variants perform worse in this setting.



Model	Tokens	Variant	Valid Loss
LLaMA-350M	7.0B	Mano	**2.7387**
		Row-wise Mano	2.7393
		Column-wise Mano	2.7541
		Joint Row-Col Mano	2.7902
		Row+Col-wise Mano	2.7988

Figure 5: Projection and normalization axis ablation for LLaMA-350M trained on C4 for 7.0B tokens. Left: validation-loss trajectories. Right: final validation losses. Joint Row-Col Mano applies both row and column projections before both normalizations; Row+Col-wise Mano applies row projection and normalization followed by column projection and normalization.

A.4 Additional Implementation Details

Nesterov accelerated gradient. Empirical studies have shown that Nesterov-style momentum can improve Muon, and it is therefore used in public Muon implementations (Jordan et al., 2024; Liu et al., 2025). For Mano, Nesterov accelerated gradient is retained as an implementation option, but all experiments reported in this paper use the standard momentum update in Algorithm 1.

Input and output parameters. The Mano update can be applied to parameters of arbitrary dimensionality by traversing the corresponding Oblique orientation at each step. Following Muon's implementation, however, the input embedding, output head, and one-dimensional bias parameters are optimized with AdamW (Jordan et al., 2024). The embedding and output layers have different structural properties from the internal matrix parameters because vocabulary activations are sparse; the experiments therefore reserve Mano for the internal matrix parameters.

B Relationship to Existing Optimizers

Adafactor. By employing a factored rank-one approximation of second moments, Adafactor is related to Mano in that both aggregate and normalize updates along parameter dimensions rather than only globally or per coordinate (Shazeer & Stern, 2018). However, Adafactor relies on exponential-moving-average second-moment statistics and controls per-parameter RMS in a manner similar to Adam, whereas Mano uses manifold normalization and explicitly projects the update onto an Oblique tangent space. Consequently, Adafactor's scaling is statistical, while Mano's is geometric.

Spectral optimizers. Mano differs from spectral optimizers such as Shampoo, SOAP, and Muon because it does not rely on matrix-wide eigenspaces, inverse roots, or spectral normalization. It performs vector-level operations that apply geometric constraints independently along the active matrix orientation. Although these row- and column-wise operations modify the update spectrum, Mano is not a spectral preconditioner and does not target a flat spectrum.

SSO and Hyperball. Hyperball and the Spectral Sphere Optimizer (SSO) also integrate manifold constraints into LLM pretraining (Wen et al., 2025a; Xie et al., 2026). Hyperball uses a Frobenius-norm constraint to regulate effective step size and weight norms, while SSO constrains weights and updates to a spectral sphere related to maximal update parameterization (Yang et al., 2023). Mano differs from both approaches: its parameters remain unconstrained, and the alternating Oblique geometry acts on momentum without spectral preconditioning.

Weight normalization and reparameterization. Weight Normalization separates each vector into magnitude and direction, while nGPT places weights and hidden states on hyperspheres (Salimans & Kingma, 2016; Loshchilov et al., 2025). Magnitude–direction parameterizations similarly introduce fixed-norm directions with learnable gains (Hägele et al., 2026). These methods modify

the model or its parameterization. Mano preserves the forward parameterization and unconstrained weights; its geometric constraint acts on lagged momentum rather than on the model itself.

Dynamics-Inspired Optimizers: Learning dynamics has been used as a tool to study how optimizers select minima (Jastrzkebski et al., 2017; Li et al., 2017; Wu et al., 2018; Xu et al., 2018; Hu et al., 2019; Nguyen et al., 2019; Zhu et al., 2019; Xie et al., 2020; Li et al., 2021). This line of research also suggests that designing optimizers with favorable dynamical properties that help escape saddle points and select flat minima is a promising approach (An, 1996; Neelakantan et al., 2015; Zhou et al., 2019; Xie et al., 2021a;b; HaoChen et al., 2021; Xie et al., 2023; Tang et al., 2025). However, although this line of research achieved considerable success in training deep neural networks before LLMs, it has not yet yielded practical algorithms that perform well in LLM training.

C Proof of Convergence

In this section, we establish convergence for the simplified row-wise Mano update without momentum, weight decay, or orientation rotation. We first consider deterministic gradients and then extend the analysis to stochastic gradients under relative projected-noise control. The corresponding column-wise results follow by transposition.

C.1 Deterministic Gradient

We prove Theorem 1 for a simplified deterministic row-oriented update. For completeness, the update is

$$\begin{cases} G_t = \nabla f(W_t), \\ U_t = V_{\mathrm{row}}(W_t)[G_t], \\ \widehat{U}_t = U_t \oslash \|U_t\|_{2,\mathrm{row}}, \\ W_{t+1} = W_t - \eta\sqrt{n}\,\widehat{U}_t. \end{cases} \tag{10}$$

Division maps a zero row to zero. The objective is lower bounded by $f_{\inf}$ and L-smooth in the Frobenius norm, every row of W_0 is nonzero, and the retained tangent component obeys

$$\sum_{i=1}^{m} \|U_{t,i,:}\|_2 \geq \gamma \|G_t\|_F \quad \text{for every } t, \text{ where } \gamma \in (0,1] \text{ is independent of } T. \tag{11}$$

Lemma 3 (Tangent Alignment)**.** *Under Equation 11,*

$$\langle G_t, \widehat{U}_t \rangle \geq \gamma \|G_t\|_F . \tag{12}$$

Proof. Because $V_{\mathrm{row}}(W_t)$ is a row-wise orthogonal projector, each $U_{t,i,:}$ is orthogonal to $G_{t,i,:} - U_{t,i,:}$. Therefore, using the zero-row convention,

$$\langle G_t, \widehat{U}_t \rangle = \sum_{i=1}^{m} \left\langle U_{t,i,:}, \frac{U_{t,i,:}}{\|U_{t,i,:}\|_2} \right\rangle = \sum_{i=1}^{m} \|U_{t,i,:}\|_2 \geq \gamma \|G_t\|_F . \tag{13}$$

The proof is now complete. □

Proof of Theorem 1. We first verify that the projector remains well-defined. If row i of W_t is nonzero, then $\widehat{U}_{t,i,:} \perp W_{t,i,:}$, and hence

$$\|W_{t+1,i,:}\|_2^2 = \|W_{t,i,:}\|_2^2 + \eta^2 n \|\widehat{U}_{t,i,:}\|_2^2 > 0. \tag{14}$$

Thus every row remains nonzero by induction from W_0.

By L-smoothness, Equation 10, and Lemma 3,

$$\begin{aligned} f(W_{t+1}) &\leq f(W_t) - \eta\sqrt{n}\,\langle G_t, \widehat{U}_t \rangle + \frac{L}{2}\eta^2 n \left\|\widehat{U}_t\right\|_F^2 \\ &\leq f(W_t) - \eta\gamma\sqrt{n}\,\|G_t\|_F + \frac{L}{2}\eta^2 mn, \end{aligned} \tag{15}$$

where $\left\|\widehat{U}_t\right\|_F^2 \le m$ because every nonzero row has unit ℓ_2 norm. Summing Equation 15 from $t = 0$ to T and using $f(W_{T+1}) \ge f_{\inf}$ gives

$$\frac{1}{T+1}\sum_{t=0}^{T}\|G_t\|_F \le \frac{f(W_0) - f_{\inf}}{\eta\gamma\sqrt{n}(T+1)} + \frac{L\eta m\sqrt{n}}{2\gamma}. \tag{16}$$

Setting $\eta = C/\sqrt{T+1}$ and bounding the minimum by the average yields Equation 7. □

C.2 Stochastic Extension

We next consider convergence of Mano with stochastic gradients. Define

$$\begin{cases} G_t = \nabla f(W_t),\ \widetilde{G}_t = G_t + \Xi_t, \\ A_t = V_{\mathrm{row}}(W_t)[G_t],\ Z_t = V_{\mathrm{row}}(W_t)[\Xi_t], \\ \widehat{U}_t = (A_t + Z_t) \oslash \|A_t + Z_t\|_{2,\mathrm{row}}, \\ W_{t+1} = W_t - \eta\sqrt{n}\,\widehat{U}_t. \end{cases} \tag{17}$$

We impose a conditional bound on the projected noise relative to the retained gradient.

Theorem 4 (Stochastic Convergence under Projected Noise)**.** *Let $(\mathcal{F}_t)_{t\ge0}$ be a filtration. Let W_0 be deterministic with nonzero rows, let W_t be $\mathcal{F}_t$-measurable, and let Ξ_t be an integrable, $\mathcal{F}_{t+1}$-measurable stochastic error. Suppose there exist $\gamma \in (0,1]$ and $\rho \in [0, 1/2)$, independent of T, such that almost surely, for every $t \in \{0,\dots,T\}$,*

$$\sum_{i=1}^{m}\|A_{t,i,:}\|_2 \ge \gamma\|G_t\|_F, \quad \mathbb{E}\left[\sum_{i=1}^{m}\|Z_{t,i,:}\|_2 \,\middle|\, \mathcal{F}_t\right] \le \rho\sum_{i=1}^{m}\|A_{t,i,:}\|_2. \tag{18}$$

Let f be lower bounded by $f_{\inf}$ and L-smooth under the Frobenius norm. For $\eta = C/\sqrt{T+1}$ with $C > 0$ and $\beta := \gamma(1-2\rho)$,

$$\min_{0\le t\le T}\mathbb{E}[\|\nabla f(W_t)\|_F] \le \frac{1}{\sqrt{T+1}}\left(\frac{f(W_0) - f_{\inf}}{\beta C\sqrt{n}} + \frac{LCm\sqrt{n}}{2\beta}\right). \tag{19}$$

The projected-noise condition in Equation 18 is stronger than standard conditional unbiasedness or bounded variance. In particular, if $A_t = 0$, it forces $Z_t = 0$ almost surely.

Proof of Theorem 4. Because $\widehat{U}_{t,i,:} \perp W_{t,i,:}$,

$$\|W_{t+1,i,:}\|_2^2 = \|W_{t,i,:}\|_2^2 + \eta^2 n\|\widehat{U}_{t,i,:}\|_2^2 > 0. \tag{20}$$

Thus every row remains nonzero by induction from W_0. Under the zero-block convention, for any vectors a, z,

$$\left\langle a, \frac{a+z}{\|a+z\|_2}\right\rangle \ge \|a\|_2 - 2\|z\|_2. \tag{21}$$

Indeed,

$$\begin{aligned}\left\langle a, \frac{a+z}{\|a+z\|_2}\right\rangle &= \|a+z\|_2 - \left\langle z, \frac{a+z}{\|a+z\|_2}\right\rangle \\ &\ge \|a+z\|_2 - \|z\|_2 \ge \|a\|_2 - 2\|z\|_2.\end{aligned} \tag{22}$$

Fix t and condition on $\mathcal{F}_t$. Because $A_t + Z_t$ lies in the range of the row-wise orthogonal projector,

$$\langle G_{t,i,:}, \widehat{U}_{t,i,:}\rangle = \langle A_{t,i,:}, \widehat{U}_{t,i,:}\rangle.$$

Applying Equation 21 row-wise and using Equation 18 gives

$$\begin{aligned}\mathbb{E}\Big[\langle G_t, \widehat{U}_t\rangle \,\Big|\, \mathcal{F}_t\Big] &\ge \sum_{i=1}^{m}\|A_{t,i,:}\|_2 - 2\mathbb{E}\left[\sum_{i=1}^{m}\|Z_{t,i,:}\|_2 \,\middle|\, \mathcal{F}_t\right] \\ &\ge (1-2\rho)\sum_{i=1}^{m}\|A_{t,i,:}\|_2 \ge \beta\|G_t\|_F.\end{aligned} \tag{23}$$

By L-smoothness, Equation 17, and $\left\|\widehat{U}_t\right\|_F^2 \le m$,

$$\mathbb{E}[f(W_{t+1}) \mid \mathcal{F}_t] \le f(W_t) - \eta\beta\sqrt{n}\,\|G_t\|_F + \frac{L}{2}\eta^2 mn. \tag{24}$$

Taking total expectation, summing Equation 24 from $t = 0$ to T, and using $f(W_{T+1}) \ge f_{\inf}$ gives

$$\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}[\|G_t\|_F] \le \frac{f(W_0) - f_{\inf}}{\eta\beta\sqrt{n}(T+1)} + \frac{L\eta m\sqrt{n}}{2\beta}. \tag{25}$$

Setting $\eta = C/\sqrt{T+1}$ and bounding the minimum by the average yields Equation 19. □

C.3 Alignment and Orientation Rotation

Theorems 1 and 4 impose the same aggregate alignment assumption on the true gradient, while the stochastic result additionally controls the projected noise. The alignment assumption is quantitative: it requires the active tangent component to retain a uniform fraction of the full gradient. For row-wise Mano, the limiting case of zero retained signal occurs when a nonzero gradient lies in $\ker V_{\mathrm{row}}(W)$, so every gradient row is parallel to the corresponding parameter row. Orientation rotation makes this degeneracy more restrictive by requiring blindness under both the row- and column-wise projectors. We next characterize the common blind space of orientation-rotating Mano when the gradient-alignment assumption fails.

Common blind space. If every row and column of W is nonzero, then

$$\begin{aligned}\ker V_{\mathrm{row}}(W) &= \{\Lambda_{\mathrm{row}}W : \Lambda_{\mathrm{row}} \text{ diagonal}\},\\ \ker V_{\mathrm{col}}(W) &= \{W\Lambda_{\mathrm{col}} : \Lambda_{\mathrm{col}} \text{ diagonal}\}.\end{aligned} \tag{26}$$

A direction in both kernels satisfies $\Lambda_{\mathrm{row}}W = W\Lambda_{\mathrm{col}}$, or equivalently $(\lambda_i - \mu_j)W_{ij} = 0$. If the bipartite support graph of W is connected, all row and column coefficients are equal, and hence

$$\ker V_{\mathrm{row}}(W) \cap \ker V_{\mathrm{col}}(W) = \mathrm{span}\{W\}. \tag{27}$$

For dense W without exact zero entries, the support graph is complete; the row- and column-side kernels have dimensions m and n in the mn-dimensional matrix space, while their intersection is the one-dimensional space $\mathrm{span}\{W\}$. Thus, in the simplified deterministic setting, an exactly zero row-oriented step is followed by a nonzero column-oriented step unless $G \in \mathrm{span}\{W\}$. This conclusion concerns exact blindness; gradients near the common blind space can still make the quantitative alignment constant arbitrarily small.

D Fisher Interpretation of Tangent Projection

This section gives a self-contained derivation of the amplitude-removal and Fisher-Rayleigh results summarized in Theorem 2. The exact projection statements apply to Mano's matrix update generally. The low-sensitivity statements additionally rely on the idealized normalized block and regularity assumptions below.

D.1 Definitions and Assumptions

Let $W \in \mathbb{R}^{m\times n}$ be a linear-layer weight, $x \in \mathbb{R}^n$ its input, and $z = Wx$. Consider the idealized directly normalized block

$$h_W(x) := \mathrm{RMSNorm}(z), \quad \mathrm{RMSNorm}(z)_i := \frac{z_i}{\|z\|_{\mathrm{rms}}}, \quad \|z\|_{\mathrm{rms}} := \sqrt{\frac{1}{m}\sum_j z_j^2}. \tag{28}$$

We omit the stabilizing constant and additive residual, assume $z \neq 0$ almost surely, and assume that W affects the predictive distribution only through $h_W(x)$. Let $\mathrm{vec}(W)$ denote vectorization. Let Q_x denote the input distribution and $p(y \mid x; W)$ the predictive distribution parameterized by W. For $x \sim Q_x$ and model-sampled outputs $y \sim p(\cdot \mid x; W)$, define

$$F(W) := \mathbb{E}_x\mathbb{E}_{y|x,W}\left[\nabla_{\mathrm{vec}(W)}\log p(y \mid x; W)\nabla_{\mathrm{vec}(W)}\log p(y \mid x; W)^\top\right]. \tag{29}$$

For a candidate direction $\Delta \in \mathbb{R}^{m\times n}$, let

$$J_W[\Delta](x) := \frac{d}{d\epsilon}\,\mathrm{RMSNorm}\big((W+\epsilon\Delta)x\big)\Big|_{\epsilon=0}. \tag{30}$$

Define the downstream Fisher metric with respect to $h = h_W(x)$ by

$$F_h(x) := \mathbb{E}_{y\sim p(\cdot|x;W)}\left[\nabla_h \log p(y \mid x;h)\nabla_h \log p(y \mid x;h)^\top\right]_{h=h_W(x)}. \tag{31}$$

For $A, B \in \mathbb{R}^{m\times n}$, define the Fisher bilinear pairing $\langle A, F(W)B\rangle := \mathrm{vec}(A)^\top F(W)\,\mathrm{vec}(B)$. The Fisher pullback identity (Martens, 2020) gives

$$\langle \Delta, F(W)\Delta\rangle = \mathbb{E}_x\left[J_W[\Delta](x)^\top F_h(x) J_W[\Delta](x)\right]. \tag{32}$$

Assume that $F_h(x) \preceq \kappa I_m$ almost surely for some $\kappa > 0$. Then

$$\langle \Delta, F(W)\Delta\rangle \le \kappa\,\mathbb{E}_x\left[\|J_W[\Delta](x)\|_2^2\right]. \tag{33}$$

For nonzero Δ, define the scale-invariant Fisher-Rayleigh quotient

$$R_F(W,\Delta) := \frac{\langle \Delta, F(W)\Delta\rangle}{\|\Delta\|_F^2}. \tag{34}$$

The row- and column-amplitude families and their Frobenius-orthogonal complements are

$$\begin{aligned} \mathcal{A}_{\mathrm{row}}(W) &:= \{\Lambda_{\mathrm{row}} W : \Lambda_{\mathrm{row}} \text{ diagonal}\}, & \mathcal{A}_{\mathrm{col}}(W) &:= \{W\Lambda_{\mathrm{col}} : \Lambda_{\mathrm{col}} \text{ diagonal}\},\\ \mathcal{T}_{\mathrm{row}}(W) &:= \mathcal{A}_{\mathrm{row}}(W)^{\perp_F}, & \mathcal{T}_{\mathrm{col}}(W) &:= \mathcal{A}_{\mathrm{col}}(W)^{\perp_F}. \end{aligned} \tag{35}$$

Here $^{\perp_F}$ denotes the orthogonal complement with respect to the Frobenius inner product. Directions in $\mathcal{A}_{\mathrm{row}}(W)$ or $\mathcal{A}_{\mathrm{col}}(W)$ change row or column amplitudes without changing the corresponding directions. With dimension-wise division on a zero block mapped to zero, define

$$\begin{aligned} V_{\mathrm{row}}(W)[M] &:= M - \hat{W}^{\mathrm{row}} \odot \left\langle M, \hat{W}^{\mathrm{row}}\right\rangle_{\mathrm{row}}, & \hat{W}^{\mathrm{row}} &:= W \oslash \|W\|_{2,\mathrm{row}},\\ V_{\mathrm{col}}(W)[M] &:= M - \hat{W}^{\mathrm{col}} \odot \left\langle M, \hat{W}^{\mathrm{col}}\right\rangle_{\mathrm{col}}, & \hat{W}^{\mathrm{col}} &:= W \oslash \|W\|_{2,\mathrm{col}}. \end{aligned} \tag{36}$$

D.2 Normalization-Induced Low-Sensitivity Directions

Proposition 5 (Row-Side Normalized-Sensitivity Bound)**.** *Suppose every row of W is nonzero, and let $w_{\mathrm{row,min}} := \min_i \|W_{i,:}\|_2$. For every nonzero diagonal $\Lambda_{\mathrm{row}} = \mathrm{diag}(\lambda_1,\dots,\lambda_m)$, let $\lambda := (\lambda_1,\dots,\lambda_m)^\top$ and $\bar{\lambda} := m^{-1}\sum_i \lambda_i$.*

$$R_F(W, \Lambda_{\mathrm{row}}W) \le \frac{\kappa m^2}{w_{\mathrm{row,min}}^2}\frac{\mathrm{Var}(\lambda)}{\|\lambda\|_2^2}, \quad \mathrm{Var}(\lambda) := \frac{1}{m}\sum_i (\lambda_i - \bar{\lambda})^2. \tag{37}$$

In particular, uniform row scaling $\Lambda_{\mathrm{row}} \propto I_m$ has $R_F = 0$.

Proof. For $\Delta = \Lambda_{\mathrm{row}}W$, the perturbed pre-activation is $z_\epsilon = (I_m + \epsilon\Lambda_{\mathrm{row}})z$. Define

$$r(\epsilon) := \sqrt{\frac{1}{m}\sum_j (1+\epsilon\lambda_j)^2 z_j^2}, \quad \bar{\lambda}_z := \frac{\sum_i z_i^2\lambda_i}{\sum_i z_i^2}. \tag{38}$$

Since $r(0) = \|z\|_{\mathrm{rms}}$ and $r'(0) = \|z\|_{\mathrm{rms}}\bar{\lambda}_z$, differentiating $(1+\epsilon\lambda_i)z_i/r(\epsilon)$ yields

$$J_W[\Lambda_{\mathrm{row}}W](x) = \frac{1}{\|z\|_{\mathrm{rms}}}(\lambda - \bar{\lambda}_z\mathbf{1}_m)\odot z. \tag{39}$$

Consequently,

$$\|J_W[\Lambda_{\mathrm{row}}W](x)\|_2^2 = m\ \mathrm{Var}_{z^2}(\lambda), \quad \mathrm{Var}_{z^2}(\lambda) := \frac{\sum_i z_i^2(\lambda_i - \bar{\lambda}_z)^2}{\sum_i z_i^2}. \tag{40}$$

Because a weighted mean minimizes weighted squared deviation,

$$\mathrm{Var}_{z^2}(\lambda) \le \frac{\sum_i z_i^2(\lambda_i - \bar{\lambda})^2}{\sum_i z_i^2} \le \sum_i (\lambda_i - \bar{\lambda})^2 = m\,\mathrm{Var}(\lambda). \tag{41}$$

Combining this inequality with Equation 33 bounds the Fisher energy by $\kappa m^2 \mathrm{Var}(\lambda)$. The denominator obeys

$$\|\Lambda_{\mathrm{row}} W\|_F^2 = \sum_i \lambda_i^2 \|W_{i,:}\|_2^2 \ge w_{\mathrm{row,min}}^2 \|\lambda\|_2^2. \tag{42}$$

Dividing proves Equation 37. For $\Lambda_{\mathrm{row}} \propto I_m$, Equation 39 is zero, so the pullback energy and R_F are zero. □

Proposition 6 (Column-Side Normalized-Sensitivity Bound)**.** *Suppose every column of W is nonzero and*

$$q_W := \mathbb{E}_x\left[\frac{\|x\|_2^2}{\|Wx\|_2^2}\right] < \infty, \quad w_{\mathrm{col,min}} := \min_j \|W_{:,j}\|_2. \tag{43}$$

For nonzero diagonal $\Lambda_{\mathrm{col}} = \mathrm{diag}(\lambda'_1, \ldots, \lambda'_n)$, let $\lambda' := (\lambda'_1, \ldots, \lambda'_n)^\top$ and $\bar{\lambda}' := n^{-1}\sum_j \lambda'_j$.

$$R_F(W, W\Lambda_{\mathrm{col}}) \le C_{\mathrm{col}} \frac{\|\lambda' - \bar{\lambda}'\mathbf{1}_n\|_2^2}{\|\lambda'\|_2^2}, \quad C_{\mathrm{col}} := \frac{\kappa m \|W\|_{\mathrm{op}}^2 q_W}{w_{\mathrm{col,min}}^2}. \tag{44}$$

Here $\|W\|_{\mathrm{op}} := \sup_{\|u\|_2=1} \|Wu\|_2$ is the induced ℓ_2 operator norm (equivalently, the spectral norm). Uniform column scaling has $R_F = 0$.

Proof. Let $P_z := I_m - zz^\top/\|z\|_2^2$ and $D := \Lambda_{\mathrm{col}} - \bar{\lambda}' I_n$. The RMSNorm derivative is

$$J_W[W\Lambda_{\mathrm{col}}](x) = \frac{1}{\|z\|_{\mathrm{rms}}} P_z W \Lambda_{\mathrm{col}} x = \frac{1}{\|z\|_{\mathrm{rms}}} P_z W D x, \tag{45}$$

where the second equality uses $P_z W x = P_z z = 0$. Therefore,

$$\|J_W[W\Lambda_{\mathrm{col}}](x)\|_2^2 \le m\|W\|_{\mathrm{op}}^2 \|D\|_F^2 \frac{\|x\|_2^2}{\|Wx\|_2^2}. \tag{46}$$

Taking expectation and applying Equation 33 bounds the numerator of R_F. The denominator satisfies

$$\|W\Lambda_{\mathrm{col}}\|_F^2 = \sum_j (\lambda'_j)^2 \|W_{:,j}\|_2^2 \ge w_{\mathrm{col,min}}^2 \|\lambda'\|_2^2. \tag{47}$$

Combining the two bounds proves Equation 44. □

Propositions 5 and 6 identify uniform scaling and amplitude coefficients with small centered energy relative to their total energy as low-sensitivity directions when the corresponding constants remain controlled. They do not imply that every amplitude-aligned direction is low-Fisher.

D.3 Amplitude-Aligned Component Removal

Theorem 7 (Exact Amplitude-Aligned Component Removal)**.** *For any diagonal Λ_{row} and Λ_{col},*

$$V_{\mathrm{row}}(W)[\Lambda_{\mathrm{row}} W] = 0, \quad V_{\mathrm{col}}(W)[W\Lambda_{\mathrm{col}}] = 0. \tag{48}$$

If every row of W is nonzero, $V_{\mathrm{row}}(W)$ is the Frobenius-orthogonal projector onto $\mathcal{T}_{\mathrm{row}}(W)$ and $\ker V_{\mathrm{row}}(W) = \mathcal{A}_{\mathrm{row}}(W)$. If every column is nonzero, the corresponding statement holds for $V_{\mathrm{col}}(W)$, $\mathcal{T}_{\mathrm{col}}(W)$, and $\mathcal{A}_{\mathrm{col}}(W)$.

Proof. Row i of $\Lambda_{\mathrm{row}} W$ is $\lambda_i W_{i,:}$. If $W_{i,:} = 0$, then $(\Lambda_{\mathrm{row}} W)_{i,:} = 0$, and the projected row is zero by the zero-block convention. Otherwise, substitution into Equation 36 gives

$$V_{\mathrm{row}}(W)[\Lambda_{\mathrm{row}} W]_{i,:} = \lambda_i W_{i,:} - \frac{\langle \lambda_i W_{i,:}, W_{i,:}\rangle}{\|W_{i,:}\|_2^2} W_{i,:} = 0. \tag{49}$$

When every row of W is nonzero, every output row of $V_{\mathrm{row}}(W)$ is orthogonal to the corresponding row of W, so its range is contained in $\mathcal{T}_{\mathrm{row}}(W)$. Conversely, every matrix in $\mathcal{T}_{\mathrm{row}}(W)$ is unchanged by the projector; hence its range is $\mathcal{T}_{\mathrm{row}}(W)$ and its kernel is $\mathcal{A}_{\mathrm{row}}(W)$. For the column-side removal identity, a zero column again yields zero by convention, while a nonzero column follows from the same scalar projection calculation. The range and kernel statement follows analogously when every column is nonzero. □

Unlike Propositions 5 and 6, Theorem 7 does not require RMSNorm, a downstream Fisher bound, or weight-conditioning assumptions. It is an exact algebraic property of Mano's tangent projection.

D.4 Change in Fisher-Rayleigh Quotient

For the row orientation, any nonzero momentum admits the unique Frobenius-orthogonal decomposition

$$\begin{aligned} M &= M_{\parallel} + M_{\perp}, \\ M_{\parallel} &:= \hat{W}^{\mathrm{row}} \odot \left\langle M, \hat{W}^{\mathrm{row}} \right\rangle_{\mathrm{row}} \in \mathcal{A}_{\mathrm{row}}(W), \\ M_{\perp} &:= V_{\mathrm{row}}(W)[M] \in \mathcal{T}_{\mathrm{row}}(W). \end{aligned} \tag{50}$$

Define

$$\alpha := \frac{\left\|M_{\parallel}\right\|_F^2}{\|M\|_F^2}, \quad \chi_F(W, M) := \frac{\langle M_{\parallel}, F(W) M_{\perp} \rangle}{\|M\|_F^2}. \tag{51}$$

Theorem 8 (Fisher-Rayleigh Effect of Tangent Projection)**.** *Let every row of W be nonzero and $0 < \alpha < 1$. Then*

$$\|V_{\mathrm{row}}(W)[M]\|_F^2 = (1 - \alpha) \|M\|_F^2 \tag{52}$$

and

$$R_F\big(W, V_{\mathrm{row}}(W)[M]\big) - R_F(W, M) = \alpha\big(R_F(W, M_{\perp}) - R_F(W, M_{\parallel})\big) - 2\chi_F(W, M). \tag{53}$$

Because $F(W) \succeq 0$,

$$|\chi_F(W, M)| \leq \sqrt{\alpha(1-\alpha) R_F(W, M_{\parallel}) R_F(W, M_{\perp})}. \tag{54}$$

Consequently, the gain in Equation 53 is strictly positive if

$$\alpha\big(R_F(W, M_{\perp}) - R_F(W, M_{\parallel})\big) > 2\sqrt{\alpha(1-\alpha) R_F(W, M_{\parallel}) R_F(W, M_{\perp})}. \tag{55}$$

Proof. Introduce

$$N_{\perp} := \|M_{\perp}\|_F^2, \quad N_{\parallel} := \left\|M_{\parallel}\right\|_F^2, \quad E_{\perp} := \langle M_{\perp}, F(W) M_{\perp} \rangle, \quad E_{\parallel} := \langle M_{\parallel}, F(W) M_{\parallel} \rangle. \tag{56}$$

Theorem 7 makes $V_{\mathrm{row}}(W)$ an orthogonal projector, so $\|M\|_F^2 = N_{\perp} + N_{\parallel}$, $M_{\perp} = V_{\mathrm{row}}(W)[M]$, and $N_{\perp} = (1 - \alpha) \|M\|_F^2$. Expanding the Fisher energy gives

$$\langle M, F(W) M \rangle = E_{\perp} + E_{\parallel} + 2\langle M_{\parallel}, F(W) M_{\perp} \rangle. \tag{57}$$

Dividing by $\|M\|_F^2$ yields

$$R_F(W, M) = (1 - \alpha) R_F(W, M_{\perp}) + \alpha R_F(W, M_{\parallel}) + 2\chi_F(W, M). \tag{58}$$

Subtracting Equation 58 from $R_F(W, M_{\perp})$ proves Equation 53. Since $F(W) \succeq 0$, Cauchy–Schwarz in the Fisher seminorm gives

$$|\langle M_{\parallel}, F(W) M_{\perp} \rangle| \leq \sqrt{E_{\parallel} E_{\perp}}. \tag{59}$$

Dividing by $\|M\|_F^2$ proves Equation 54; substituting the upper bound for the cross term gives Equation 55. □

Corollary 9 (Column-Side Fisher-Rayleigh Effect)**.** *Let every column of W be nonzero and define*

$$M_{\parallel}^{\mathrm{col}} := \hat{W}^{\mathrm{col}} \odot \left\langle M, \hat{W}^{\mathrm{col}} \right\rangle_{\mathrm{col}}, \qquad M_{\perp}^{\mathrm{col}} := V_{\mathrm{col}}(W)[M],$$

$$\alpha_{\mathrm{col}} := \frac{\left\| M_{\parallel}^{\mathrm{col}} \right\|_F^2}{\|M\|_F^2}, \qquad \chi_F^{\mathrm{col}}(W, M) := \frac{\langle M_{\parallel}^{\mathrm{col}}, F(W) M_{\perp}^{\mathrm{col}} \rangle}{\|M\|_F^2}. \tag{60}$$

If $0 < \alpha_{\mathrm{col}} < 1$, then

$$R_F\big(W, V_{\mathrm{col}}(W)[M]\big) - R_F(W, M) = \alpha_{\mathrm{col}} \left[R_F(W, M_{\perp}^{\mathrm{col}}) - R_F(W, M_{\parallel}^{\mathrm{col}}) \right] - 2\chi_F^{\mathrm{col}}(W, M), \tag{61}$$

with

$$|\chi_F^{\mathrm{col}}(W, M)| \le \sqrt{\alpha_{\mathrm{col}}(1 - \alpha_{\mathrm{col}}) R_F(W, M_{\parallel}^{\mathrm{col}}) R_F(W, M_{\perp}^{\mathrm{col}})}. \tag{62}$$

Proof. By Theorem 7, $V_{\mathrm{col}}(W)$ is the Frobenius-orthogonal projector onto $\mathcal{T}_{\mathrm{col}}(W)$. Applying the decomposition and energy expansion from the proof of Theorem 8 with column-wise quantities gives Equation 61; Cauchy–Schwarz in the Fisher seminorm gives the cross-term bound. □

Thus Mano always removes the active amplitude-aligned component exactly. Under the normalized-block assumptions, members with small relative centered coefficient energy have bounded Fisher-Rayleigh quotient; when the retained tangent component is sufficiently more informative and the cross term is controlled, tangent projection increases the Fisher-Rayleigh quotient.

E Predictable Parameter Norm of Mano

We briefly characterize Mano's parameter norm under decoupled weight decay. Assume every active row or column of V_t is nonzero. Since $\hat{V}_t$ is tangent to the active Oblique geometry, we have $\langle W_t, \hat{V}_t \rangle = 0$. Moreover, $\left\| \hat{V}_t \right\|_F^2 = m$ on row-oriented steps and $\left\| \hat{V}_t \right\|_F^2 = n$ on column-oriented steps. Since $d_{\mathrm{row}} = n$ and $d_{\mathrm{col}} = m$, both orientations satisfy $d_{k_t} \left\| \hat{V}_t \right\|_F^2 = mn$.

Expanding the Mano update in Algorithm 1 therefore gives the exact recurrence

$$\|W_{t+1}\|_F^2 = (1 - \eta_t \lambda_{\mathrm{wd}})^2 \|W_t\|_F^2 + \eta_t^2 s^2 mn. \tag{63}$$

Let $a_t := 1 - \eta_t \lambda_{\mathrm{wd}}$. Iterating equation 63 yields

$$\|W_T\|_F^2 = \left(\prod_{j=0}^{T-1} a_j^2 \right) \|W_0\|_F^2 + s^2 mn \sum_{t=0}^{T-1} \eta_t^2 \prod_{j=t+1}^{T-1} a_j^2. \tag{64}$$

Thus, for fixed hyperparameters and learning-rate schedule, the final norm of every Mano-optimized matrix is determined independently of its gradient and momentum trajectory. In particular, for constant $\eta_t = \eta$, $\lambda_{\mathrm{wd}} > 0$, and $0 < \eta \lambda_{\mathrm{wd}} < 2$,

$$\lim_{t \to \infty} \|W_t\|_F^2 = \frac{\eta s^2 mn}{\lambda_{\mathrm{wd}}(2 - \eta \lambda_{\mathrm{wd}})}. \tag{65}$$

Consequently, Mano's update-to-weight ratio $\eta_t s \sqrt{mn} / \|W_t\|_F$, which determines its global angular effective learning rate, is also predictable from the training schedule.

Muon (Jordan et al., 2024; Liu et al., 2025) does not generally admit the same trajectory-independent law. Writing its update as $W_{t+1} = (1 - \eta_t \lambda_{\mathrm{wd}}) W_t - \eta_t D_t^{\mathrm{Muon}}$, its norm satisfies

$$\|W_{t+1}\|_F^2 = a_t^2 \|W_t\|_F^2 + \eta_t^2 \left\| D_t^{\mathrm{Muon}} \right\|_F^2 - 2 a_t \eta_t \langle W_t, D_t^{\mathrm{Muon}} \rangle. \tag{66}$$

Even when ideal matrix-sign normalization fixes $\left\| D_t^{\mathrm{Muon}} \right\|_F^2$ through its rank, it does not ensure $\langle W_t, D_t^{\mathrm{Muon}} \rangle = 0$. This alignment term depends on the optimization trajectory, while finite Newton–Schulz iterations additionally retain dependence on the momentum singular values. Muon's final parameter norm therefore requires assumptions on the parameter-update alignment, whereas Mano's tangent construction eliminates this term exactly.