DEMYSTIFYING MANIFOLD CONSTRAINTS IN LLM PRE-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

The recent success of matrix optimizers (e.g., Muon) suggests that specific normalization of momentum—such as orthogonalization and row-wise normalization—benefits both the stability and acceleration of LLM training. Consequently, several recent studies have suggested that weights should also be normalized, leading to a Riemannian optimization problem. While such constrained training frameworks demonstrate superior performance, the effects of explicitly constraining weights, and their interaction with existing stabilization mechanisms, remain less understood. To bridge this gap, we study manifold constrained training dynamics through activation scales, rotational dynamics, and the update-to-weight ratio. We propose a Riemannian spectral steepest descent optimizer called `MACRO`, alongside a radius selection principle to serve as our testbed. Our analysis and numerical experiments reveal that RMSNorm and manifold constraints serve overlapping roles, and that weight decay can be completely eliminated when manifold constraints are applied. By controlling the update-to-weight ratio, constrained training significantly alleviates update cancellation, empirically demonstrating that MACRO is robust to low-precision computation and competitive with existing algorithms for standard LLM pre-training.

1 INTRODUCTION

Large language model (LLM) pre-training has benefited from matrix optimizers that apply structured normalization to gradients or momentum, including orthogonalization (Jordan, 2024) and row-wise normalization (Ma et al., 2025; Glentis et al., 2026). These have motivated a growing line of research work that restrict the weights on manifolds that serve the purpose of orthogonalization/normalization on weights (Zhao, 2026; Jiang et al., 2026; Arefin et al., 2026; Gu and Xie, 2026; Newhouse et al., 2025; Xie et al., 2026; Yang and Lai, 2026; Dolatabadi et al., 2026; Su, 2025a;b; Wen et al., 2026; Bernstein, 2025), leading to optimization problems over Riemannian manifolds. Promising numerical results have been reported for solving this Riemannian optimization problem.

However, the underlying mechanisms driving the empirical gains of Riemannian optimization are still not well understood, and existing algorithms often suffer from at least one of three critical drawbacks. First, some algorithms (Wen et al., 2026) lack rigorous theoretical guarantees. Second, some other algorithms relying on double-loop formulations (Xie et al., 2026) incur substantial computational overhead that is prohibitive for large-scale pre-training. Finally, existing algorithms that follow the Riemannian optimization principle with convergence guarantees (Bernstein, 2025; Yang and Lai, 2026) lack evaluation on modern LLM architectures, and it is thus not clear if they are suitable for large-scale pre-training. A comprehensive discussion of additional related works is in Appendix A.

The role of weight constraints is not immediate in modern LLM architectures. Some popular heuristics such as RMSNorm explicitly rescales activations (Zhang and Sennrich, 2019), while weight decay changes weight norms and rotational dynamics (Loshchilov and Hutter, 2019; Kosson et al., 2024a; Li et al., 2020). A change in weight scale can therefore be partly offset by normalization, and some effects of explicit constraints may overlap with those of weight decay. At the same time, the size of an update relative to its weight matters for whether it survives low-precision rounding (Micikevicius et al., 2018; Zamirai et al., 2020). These interactions make it difficult to infer the role of a weight constraint from final validation loss alone, motivating our central question:

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.

How do manifold constraints shape the training dynamics of modern LLMs?

In this paper, we consider constraining the weights on Frobenius spheres, spectral spheres, and oblique manifolds, and examine their connections to RMSNorm, weight decay, and FP32/BF16 master-weight training. We inspect the training dynamics through three complementary aspects: activation scales, rotation between consecutive weight iterates, and the update-to-weight ratio.

Existing approaches for solving manifold constrained LLM pre-training include heuristic normalization (Wen et al., 2026), double-loop constrained solvers (Xie et al., 2026), and general Riemannian solvers (Bernstein, 2025; Yang and Lai, 2026). To study different geometries within a common practical framework, we introduce the **M**sign-**A**ligned **C**onstrained **R**iemannian **O**ptimizer (MACRO), a single-loop optimizer based on Riemannian spectral steepest descent, and establish its convergence under standard assumptions. We also derive norm-specific radius scalings under normalized Gaussian initialization, accounting for the input and output dimensions of each weight matrix. Together, the optimizer and radius selection serve as our testbed. Our main contributions are summarized below.

1. **Activation Dynamics Effect (Section 4.1).** We identify overlaps of functions between manifold constraints and RMSNorm. In an ablation architecture that removes the standard learnable RMSNorm layers following all linear layers, MACRO is more stable than Muon (Jordan, 2024). Under the standard architecture with learnable RMSNorm layers, the performance of MACRO is more robust to changes in the constraint radius because of the recaling from RMSNorm.
2. **Rotational Dynamics Effect (Section 4.2).** We theoretically and empirically analyze how manifold constraints alter rotational dynamics during LLM pre-training. Specifically, we demonstrate that manifold constraints intrinsically govern the rotation angle between consecutive updates. The Frobenius constraint imposes global static rotation angles, while adaptive subspace rotations are imposed under spectral constraints. This demonstrates that when manifold constraints are present, weight decay is not necessary.
3. **Weight-Update-Ratio Regulation Effect (Section 4.3).** When switching master-weight precision from FP32 to BF16 during training, the validation loss naturally increases. Our experiments find that the loss increase incurred by MACRO is substantially smaller than the one incurred by Muon. We connect the controlled relative learning rate to update cancellation through a Gaussian coordinate-threshold model to provide an explanation for such a difference.
4. **Standard Pretraining Results (Section 5).** Across standard Qwen3-style architectures ranging from 240M to 1.2B parameters, we demonstrate that MACRO, which solves the manifold-constrained problem, consistently outperforms Muon, which solves the unconstrained problem. MACRO is competitive with other constrained optimizers, such as MuonH. MACRO also has substantially lower per-step costs than double-loop constrained optimizers like SSO.

2 Notation and Preliminaries

Notation. Throughout this paper, we use capital bold letters such as $\mathbf{W}$ to represent matrices and $\mathbf{W}_{i,:}$ ($\mathbf{W}_{:,j}$) to denote the i-th row (j-th column) of $\mathbf{W}$. Moreover, $\|\mathbf{W}\|_F$ denotes the Frobenius norm of $\mathbf{W}$; $\|\mathbf{W}\|_2$ denotes the spectral norm of a matrix $\mathbf{W}$, and $\|\mathbf{W}\|_*$ denotes the nuclear norm of $\mathbf{W}$, which is the dual norm of $\|\mathbf{W}\|_2$; $\|\mathbf{x}\|_2$ denotes the Euclidean norm of a vector $\mathbf{x}$; $\mathrm{msign}(\mathbf{W})$ denotes the Matrix Sign Operator of matrix $\mathbf{W}$, i.e. $\mathrm{msign}(\mathbf{W}) = \mathbf{U}\mathbf{V}^\top$, where $\mathbf{U}(\mathbf{V})$ is the left(right) singular vectors of $\mathbf{W}$. For manifold $\mathcal{M}$, we use $\mathrm{T}_{\mathcal{M}}\mathbf{W}$ to denote its tangent space at point $\mathbf{W}$. Further, we denote by $\mathrm{proj}_{\mathrm{T}_{\mathcal{M}}\mathbf{W}}(G)$ the projection of G onto the tangent space to the point $\mathbf{W}$ on the manifold, and use $\mathcal{P}_{\mathcal{M}}(X)$ to denote the projection (retraction) of X onto the manifold $\mathcal{M}$. We use $\nabla\mathcal{L}$ to denote the Euclidean gradient of $\mathcal{L}$ and $\nabla_{\mathcal{M}}\mathcal{L}(\mathbf{W})$ to denote the Riemannian gradient of $\mathcal{L}$. For embedded submanifolds considered in this paper, Riemannian gradient is the projection of the Euclidean gradient to the tangent space, i.e., $\nabla_{\mathcal{M}}\mathcal{L}(\mathbf{W}) = \mathrm{proj}_{\mathrm{T}_{\mathcal{M}}\mathbf{W}}(\nabla\mathcal{L}(\mathbf{W}))$.

Manifold Constraints. We consider four constraint choices for the weight matrix $\mathbf{W} \in \mathbb{R}^{D_{\text{out}} \times D_{\text{in}}}$:

- **Frobenius Sphere (Wen et al., 2026)**: $\mathcal{M}_F(R) = \{\mathbf{W} : \|\mathbf{W}\|_F = R\}$.
- **Spectral Sphere (Xie et al., 2026):** $\mathcal{M}_S(R) = \{\mathbf{W} : \|\mathbf{W}\|_2 = R\}$ *
- **Input (resp. Output) Oblique Manifold (Gu and Xie, 2026):** $\mathcal{M}_{O_{\text{in}}}(R_{\text{in}}) = \{\mathbf{W} : \|\mathbf{W}_{:,j}\|_2 = R_{\text{in}}, \forall j\}$ (resp. $\mathcal{M}_{O_{\text{out}}}(R_{\text{out}}) = \{\mathbf{W} : \|\mathbf{W}_{i,:}\|_2 = R_{\text{out}}, \forall i\}$.).

*See the detailed discussions of spectral sphere in Appendix C.1 & B.1.

For each of these manifolds, we use $\|\cdot\|_{\mathcal{M}}$ to denote the norm in their definition, i.e. for Frobenius sphere $\|\cdot\|_{\mathcal{M}} = \|\cdot\|_F$, for spectral sphere $\|\cdot\|_{\mathcal{M}} = \|\cdot\|_2$, for input oblique manifold, $\|\cdot\|_{\mathcal{M}} = \max_j \|\cdot_{:,j}\| = \|\cdot\|_{1\to 2}$ and for output oblique manifold $\|\cdot\|_{\mathcal{M}} = \max_i \|\cdot_{i,:}\| = \|\cdot\|_{2\to\infty}$.

3 Constrained Training Algorithm and Radius Selection

3.1 MACRO Algorithm and Convergence

We consider the following manifold constrained LLM training problem:

$$\min_{\mathbf{W}} \mathcal{L}(\mathbf{W}) = \mathbb{E}_{\xi\sim\mathcal{D}}\left[\ell(\mathbf{W},\xi)\right] \text{ s.t. } \mathbf{W}\in\mathcal{M}, \tag{1}$$

where $\mathcal{M}$ is a manifold discussed in Sec 2. To solve equation 1, we propose a single-loop optimization framework, named **Msign-Aligned Constrained Riemannian Optimizer (MACRO)**. MACRO is a provably convergent Riemannian algorithm that maintains the spectral preconditioning benefits of the Muon optimizer. The complete description of MACRO is given in Alg 1 and its main update is:

$$O_t = \mathrm{msign}(\nabla_{\mathcal{M}}\mathcal{L}(\mathbf{W}_t)), \quad \mathbf{W}_{t+1} = \mathcal{P}_{\mathcal{M}}\left(\mathbf{W}_t - \eta_t R\cdot O_t/\left(\|O_t\|_{\mathcal{M}}+\epsilon\right)\right), \tag{2}$$

where η_t is the learning rate and ϵ is a numerical stabilizer. This update scheme fundamentally differs from standard optimizers and the original Muon in the following key aspects:

- **Tangent Space Projection:** We project it onto the tangent space at the current weight $\mathbf{W}_t$ to obtain the Riemannian gradient $\nabla_{\mathcal{M}}\mathcal{L}(\mathbf{W}_t)$. The exact computations are detailed in Appendix B.1.
- **Riemannian Spectral Steepest Descent:** MACRO evaluates the Linear Minimization Oracle (LMO) on the **Riemannian gradient**. This extracts the optimal steepest descent direction reflecting the local geometry of the manifold.
- **Update-Weight Ratio Alignment:** After obtaining the update direction O_t, we rescale it by the constraint radius R. Together with the fixed weight norm, this controls the proposed update-weight ratio through η_t. Section 4.3 analyzes this relative step.
- **Manifold Projection/Retraction:** We apply a projection $\mathcal{P}_{\mathcal{M}}$ (acting as a retraction) at every iteration to map the weights back onto the target constraint manifold $\mathcal{M}$. The exact computations are detailed in Appendix B.1.

Algorithm 1 Msign-Aligned Constrained Riemannian Optimizer (MACRO)

Input: Learning rate η_t, Manifold Constraint $\mathcal{M}$, Hyperparameters β, R, T
for $t = 0, 1, \ldots, T$ **do**
  $G_t = \nabla\ell(\mathbf{W}_t, \xi_t)$
  $M_t = \beta M_{t-1} + (1-\beta)G_t$
  $\Phi_t = \mathrm{proj}_{\mathrm{T}_{\mathcal{M}}\mathbf{W}_t}(M_t)$ ▷ Projection to the Tangent Space
  $O_t = \mathrm{msign}(\Phi_t)$ ▷ Compute the steepest descent direction via LMO
  $\widetilde{\nabla}_t = R\cdot O_t/(\|O_t\|_{\mathcal{M}}+\epsilon)$ ▷ Normalization and Scaling
  $\mathbf{W}_{t+1} = \mathcal{P}_{\mathcal{M}}\left(\mathbf{W}_t - \eta_t\cdot\widetilde{\nabla}_t\right)$ ▷ Descent Step and Manifold Projection
end for

Under standard assumptions (Assumption 1 & 2 in Appendix C.1), we have the following theorem.

Theorem 1. *Denote $\Delta = \mathcal{L}(\mathbf{W}_0) - \mathcal{L}^*$. Setting $\beta = 1-1/\sqrt{T}$, $\eta_t = \eta = \Omega(\sqrt{\Delta/(LT^{3/2})})$, the sequence $\{\mathbf{W}_t\}$ generated by Algorithm 1 satisfies $\min_{t=0,\ldots,T-1}\mathbb{E}\|\nabla_{\mathcal{M}}\mathcal{L}(\mathbf{W}_t)\|_* \le \mathcal{O}\left(\sqrt{L\Delta}\sigma T^{-1/4}\right)$.*

3.2 Initialization-Based Radius Selection

We now address the choice of radius for each weight constraint in MACRO. We consider a linear layer $\mathbf{Y} = \mathbf{X}\mathbf{W}^\top$, where $\mathbf{X}\in\mathcal{R}^{T\times n}$ and $\mathbf{W}\in\mathcal{R}^{p\times n}$, with $n := D_{\text{in}}$ and $p := D_{\text{out}}$. Following Yang et al. (2023) and Su (2026a), our goal is measuring the activation of a liner layer given the initialized Gaussian initialization, and the input RMSNorm is $O(1)$. The following proposition summarizes the initialization result in Theorem 2 of Appendix C.2.

Proposition 1 (Initialization-Based Radius Selection). *Define* $r_{\mathbf{X}} := \frac{\|\mathbf{X}\|_F}{\sqrt{Tn}} = O(1)$ *and* $r_{\mathbf{Y}} := \frac{\|\mathbf{X}\mathbf{W}^\top\|_F}{\sqrt{Tp}}$. *Let* $T, n, p \geq 1$, *and* $\mathbf{G} \in \mathcal{R}^{p\times n}$ *be independent of* $\mathbf{X} \in \mathcal{R}^{T\times n}$ *with i.i.d.* $\mathcal{N}(0,1)$ *entries:*

- ***Frobenius Sphere:*** *If* $\mathbf{W}_0 = R_{fro}\mathbf{G}/\|\mathbf{G}\|_F$ *and* $R_{fro} = r\sqrt{p}$, *then* $\mathbb{E}_{\mathbf{G}}[r_{\mathbf{Y}}^2 \mid \mathbf{X}] = r^2 r_{\mathbf{X}}^2$.
- ***Spectral Sphere:*** *If* $\mathbf{W}_0 = R_{spec}\mathbf{G}/\|\mathbf{G}\|_2$ *and* $R_{spec} = r\sqrt{p/n}$, *then* $r_{\mathbf{Y}} \leq r r_{\mathbf{X}}$ *deterministically. For every* $\varepsilon \in (0,1)$ *and* $a > 0$, *the lower bound* $r_{\mathbf{Y}} \geq \frac{r\sqrt{1-\varepsilon}}{\sqrt{n/p}+1+a} r_{\mathbf{X}}$ *holds with probability at least* $1 - e^{-p\varepsilon^2/4} - e^{-a^2 p/2}$ *over* $\mathbf{G}$, *conditional on the fixed* $\mathbf{X}$.

For $r, r_{\mathbf{X}} = \Theta(1)$, *the concentration bounds in Theorem 2 give* $r_{\mathbf{Y}} = \Theta(1)$ *with high probability.*

According to the above proposition, the right table summarizes the radii chosen based on the Gaussian initialization. The hyperparameter $r = \Theta(1)$ is separate from the dimension-dependent scaling derived above; Table 3 studies its empirical sensitivity.

Manifold	**Radius Setting**
Spectral Sphere	$r\sqrt{p/n}$
Frobenius Sphere	$r\sqrt{p}$
Output Oblique	r
Input Oblique	$r\sqrt{p/n}$

Oblique vs. Frobenius Constraints. For the oblique constraints, we match the total Frobenius norm to R_{fro}, using $\|\mathbf{W}\|_F^2 = pR_{\text{out}}^2$ or $\|\mathbf{W}\|_F^2 = nR_{\text{in}}^2$ on the respective manifolds. In the comparison in Appendix D.5, the Frobenius constraint achieves lower validation loss than the oblique constraints across the tested learning rates, while their spectral and maximum row/column norm trajectories are similar (Figure 13). We therefore focus on the Frobenius and spectral spheres in the remaining experiments.

Empirical Norm Geometry during the Training Process. The proposed Frobenius radius ensures that the single-layer amplification upper bound is preserved beyond initialization. Define

$$\kappa(\mathbf{W}) := d\,\frac{\|\mathbf{W}\|_2^2}{\|\mathbf{W}\|_F^2}, \qquad d := \min\{n, p\}. \tag{3}$$

For any input $\mathbf{X}$ and any weight matrix satisfying $\|\mathbf{W}\|_F = r_{\text{fro}}\sqrt{p}$, the output $\mathbf{Y} = \mathbf{X}\mathbf{W}^\top$ satisfies $r_{\mathbf{Y}} \leq \sqrt{\frac{n}{p}}\,\|\mathbf{W}\|_2\, r_{\mathbf{X}} = r_{\text{fro}}\sqrt{\frac{n}{d}\,\kappa(\mathbf{W})}\, r_{\mathbf{X}}$. Figure 6 in Appendix C.2.1 shows that the reported values of $\kappa(\mathbf{W})$ remain $\Theta(1)$ during training. Combined with $n/d = \Theta(1)$, we get $r_{\mathbf{Y}} = \mathcal{O}(r_{\mathbf{X}})$, i.e. the proposed Frobenius radius ensures that the single-layer amplification is bounded.

4 Effects of Manifold Constraints on Training Dynamics

So far we have discussed an explicit manifold-constrained optimizer and radius selection for LLM training. We now examine how manifold constraints interact with commonly used training techniques such as RMSNorm layers, weight decay, and low-precision training. Additional experiments on intersection with learning-rate warmup are reported in the appendix (Table 12).

4.1 Activation Dynamics Effect: Interplay with RMSNorm

In standard LLM architectures, normalization layers (Ioffe and Szegedy, 2015; Wu and He, 2018; Ba et al., 2016; Salimans and Kingma, 2016; Qiao et al., 2020) explicitly control the forward activation scale. For example, the RMSNorm layer (Zhang and Sennrich, 2019) uses a learnable affine parameter, γ, to scale activations. However, Proposition 1 shows that manifold constraints also determine the output scale at Gaussian initialization, suggesting an interaction with normalization during training. Figure 1 illustrates this interaction: larger constraint radii R are associated with smaller learned norms of γ. This compensatory relation can buffer changes in weight scale. To examine the respective effects of RMSNorm and manifold constraints on training dynamics, we remove the standard learnable RMSNorm layers. The modified model retains a parameter-free RMSNorm ($\gamma = 1$) immediately after the attention block, QK-norm, and a parameter-free normalization after the initial embedding layer. It also uses Norm-Gated SwiGLU, which normalizes the Swish branch before the element-wise multiplication:

$$\mathbf{Y} = \Big(\text{Swish}(\mathbf{X}\mathbf{W}_1^\top)/\|\,\text{Swish}(\mathbf{X}\mathbf{W}_1^\top)\|_{\text{RMS}}\Big) \odot (\mathbf{X}\mathbf{W}_2^\top). \tag{4}$$

By training a 330M model with the ablation setting in Table 1, we evaluate how the optimizer stabilizes training after the standard learnable RMSNorm layers are removed. The empirical results using Muon, MACRO-fro, and MACRO-spec are summarized in Table 2. The results show that Muon diverges at 3×10^{-2}, while both MACRO variants remain stable across the tested learning rates. MACRO-spec achieves the lowest best-tuned validation loss (2.739), compared with 2.758 for MACRO-fro and 2.813 for Muon. This result highlights the normalizing effect of manifold constraints: while MACRO maintains robust stability at the tested learning rates, Muon may fail once the safeguard of learnable RMSNorm layers is removed.

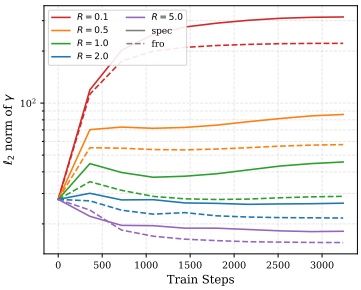


Figure 1: ℓ_2 norm of γ during training.

Table 2: Validation loss for QWEN3-like 330M pre-training in the ablation architecture without standard learnable RMSNorm layers.

Optimizer	**Learning Rate**				
	3×10^{-3}	5×10^{-3}	7×10^{-3}	1×10^{-2}	3×10^{-2}
Muon (baseline)	2.912	2.849	2.823	2.813	NaN
MACRO-fro (Ours)	2.901	2.825	2.787	**2.758**	2.781
MACRO-spec (Ours)	2.820	2.768	2.747	**2.739**	2.819

Radius Sensitivity with Standard RMSNorm. We now return to the standard RMSNorm architecture, distinct from the 330M ablation above. Table 3 evaluates the radius multiplier r at the 240M and 600M scales, using the dimension-dependent radii in Section 3.2. MACRO is relatively insensitive to the radius multiplier r over the practical range $r \in [1, 5]$, particularly at the 600M scale. At 600M, validation perplexity stays within 15.296–15.329 for MACRO-fro and 15.277–15.312 for MACRO-spec over this range. RMSNorm can buffer changes in weight scale, providing an explanation for this hyperparameter robustness consistent with the learned normalization rescales with radius in Figure 1.

Table 1: Architecture Comparison

	Standard	**Ablation**
Activation	SwiGLU	Eq. 4
Attn: Norm	`False`	`True`
RMSNorm Layer	`True`	`False`

Table 3: Radius sensitivity of MACRO under Frobenius and spectral constraints in the standard RMSNorm architecture at the 240M and 600M model scales. We report final validation perplexity.

Constraint	**Learning Rate**	$r = 0.5$	$r = 1$	$r = 2$	$r = 5$
		240M Model			
MACRO-fro	0.008	20.911	20.816	20.781	20.772
	0.016	20.669	20.542	20.526	20.518
	0.032	21.739	21.381	21.286	21.288
MACRO-spec	0.004	21.500	21.129	21.062	21.044
	0.008	21.103	20.678	20.535	20.457
	0.016	21.385	20.981	20.833	20.738
		600M Model			
MACRO-fro	0.012	15.329	15.296	15.297	15.329
MACRO-spec	0.008	15.434	15.312	15.277	15.288

☞ ***Key Takeaway:*** *Manifold constraints and RMSNorm have aligned roles: constraints improve stability in the ablation architecture, while performance with standard RMSNorm is relatively insensitive to the radius multiplier.*

4.2 Rotational Dynamics Effect: Interplay with Weight Decay

Decoupled weight decay (Loshchilov and Hutter, 2019) is standard practice in model training. Recent works (D'Angelo et al., 2024; Xie and Li, 2024; Kosson et al., 2024a; Defazio, 2025; Li et al., 2020; Chen et al., 2025a) have studied its effects on weight scale and optimization geometry. Here, we focus on the rotational dynamics of the weights. Weight decay balances radial shrinkage against the growth induced by optimization updates, allowing weight norms and angular updates to approach a rotational equilibrium (Kosson et al., 2024a; Li et al., 2020). In other word, the late stage training with weight decay only changes the *rotational angle* of consecutive updates, defined as $\theta_t = \angle(\mathbf{W}_{t+1}, \mathbf{W}_t)$ (Kosson et al., 2024a). Such equilibrium is achieved by weight decay empirically as analyzed in Kosson et al. (2024a); Li et al. (2020). Because of this gradual process, standard optimizers undergo a long *transient phase*. During this early phase, parameter scales and update angles remain unregulated. In contrast, explicit manifold constraints offer an explicit solution. By restricting the optimization trajectory, the rotation angle θ_t exactly from the first iteration.

Static Rotational Equilibrium under the Frobenius Sphere. Manifold constraints also fundamentally alter the angular trajectory. Traditional weight decay requires a transient phase to reach rotational equilibrium (Kosson et al., 2024a). Manifold constraints bypass this phase entirely and enforce a pure rotational state from the very first step. We explicitly quantify the rotational angle for the Frobenius sphere $\mathcal{M}_F(R)$. Define $\alpha_t := \cos\angle(\mathbf{W}_t, O_t)$, substituting $\widetilde{\nabla}_t = (R/\|O_t\|_F)O_t$ in the update formula we have:

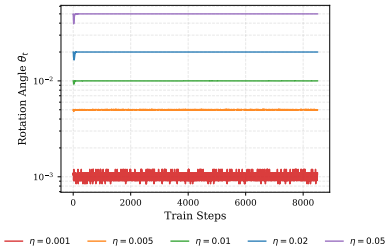


Figure 2: θ_t under Frobenius Sphere.

$$\langle \mathbf{W}_t - \eta_t \widetilde{\nabla}_t, \mathbf{W}_t \rangle_F = R^2(1 - \eta_t \alpha_t), \qquad \|\mathbf{W}_t - \eta_t \widetilde{\nabla}_t\|_F = R\sqrt{1 - 2\eta_t\alpha_t + \eta_t^2}.$$

Using this inner product constraint, we explicitly compute the rotational angle θ_t between consecutive steps:

$$\theta_t = \arccos\left(\frac{\langle \mathbf{W}_{t+1}, \mathbf{W}_t \rangle_F}{\|\mathbf{W}_{t+1}\|_F \|\mathbf{W}_t\|_F}\right) = \arccos\left(\frac{1 - \eta_t\alpha_t}{\sqrt{1 - 2\eta_t\alpha_t + \eta_t^2}}\right)$$

Applying the second-order Taylor expansion for the cosine function, we obtain a simple geometric identity: $\theta_t \approx \eta_t \sqrt{1 - \alpha_t^2} \approx \eta_t$ as $\alpha_t \approx 0$ (details in Appendix C.3). This indicate that the Frobenius sphere constraint directly enforces a steady-state rotation while standard weight decay penalizes the norm to force a rotation. The full algebraic derivation is in Appendix C.3.

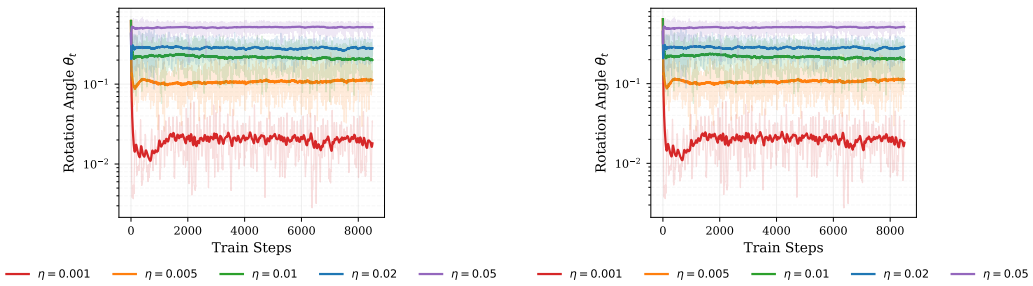
To empirically validate this dynamic, we train a 330M Qwen3-like architecture using a constant learning rate 0.01 and various alignment ratios c ranging from $[0.1, 5.0]$. As Figure 2 shows, the Frobenius rotation angle remains flat, confirming our derived identity $\theta_t^{\text{fro}} \approx \eta_t$.

Adaptive Rotational Equilibrium under the Spectral Sphere. We now analyze the rotation angle θ_t for spectral sphere, which requires a different approach because spectral norm is not induced by an inner product. Instead of a single global rotation angle, the rotation is anisotropic: different singular subspaces rotate by different amounts. We focus on the maximum singular subspace, which is a natural choice because the spectral sphere explicitly regulates the maximum singular value. Let $\mathbf{u}_t$ and $\mathbf{v}_t$ denote the leading left and right singular vectors of $\mathbf{W}_t$. We define the effective rotational angle for this primary direction as $\theta_t = \max\{\theta_u, \theta_v\}$, where $\theta_u = \arccos(|\mathbf{u}_{t+1}^\top \mathbf{u}_t|)$ and $\theta_v = \arccos(|\mathbf{v}_{t+1}^\top \mathbf{v}_t|)$. We treat our unprojected update step $-\eta_t \cdot \widetilde{\nabla}_t$ as the perturbation matrix, with magnitude $\|\Delta\mathbf{W}_t\|_2 = \eta_t R$. By Wedin $\sin\Theta$ theorem (Davis and Kahan, 1970; Wedin, 1972), we provide the following bound (see Appendix C.4 for proofs):

$$\max\{\sin\theta_u, \sin\theta_v\} \le \|\Delta\mathbf{W}_t\|_2 / \Delta_1 \approx \eta_t R / \Delta_1.$$

Here we assume the leading singular value is unique, so the spectral gap $\Delta_1 := R - \sigma_2(\mathbf{W}_{t+1}) > 0$. Substituting this definition into the above bound yields: $\sin\theta_t \lesssim \frac{\eta_t R}{R - \sigma_2(\mathbf{W}_{t+1})}$. Applying the small-angle approximation ($\sin\theta \approx \theta$), we obtain:

$$\theta_t \lesssim \eta_t \left(\frac{R}{R - \sigma_2(\mathbf{W}_{t+1})}\right). \tag{5}$$



(a) Principal rotation angle of the left leading singular vector u_1.

(b) Principal rotation angle of the right leading singular vector v_1.

Figure 3: Rotational angle under the spectral constraint: The spectral rotation angles display high-frequency variance. Despite this, both the u_1 and v_1 trajectories remain well-balanced and stable. This dynamic behavior supports our Wedin bound analysis: the local spectral gap continuously modulates these adaptive rotations.

Figure 3 visualizes the principal rotation angles under the spectral constraint. We used the same setting as used in the Frobenius Sphere experiment depicted in Figure 2. Geometrically, the left and right singular vectors u_1 and v_1 rotate the output and input features, respectively. Despite this difference, both subspaces exhibit highly similar rotational trajectories, which indicates that the input and output feature rotations remain well-balanced during training. Furthermore, comparing the rotation angle under the spectral constraint (Eq. equation 5) to the Frobenius case ($\theta_t \approx \eta_t$, Figure 2) reveals a key difference. The Frobenius constraint imposes a static, global rotation determined purely by constant hyperparameters. In contrast, the spectral constraint induces an *adaptive* rotation. For the leading feature dimension, the training-dependent spectral gap $R - \sigma_2(\mathbf{W}_{t+1})$ explicitly modulates the rotation angle. These distinct behaviors support our theoretical analysis. Overall, manifold constraints provide a more principled approach for rotational regulation than weight decay. We further study the effect of weight decay on MACRO and Muon. Table 11 in Appendix D.2 shows that Muon is relatively sensitive to weight decay, and the optimal weight decay is also coupled with the learning rate. In contrast, Table 10 in Appendix D.2 shows that adding weight decay to MACRO changes the validation loss by at most 0.001 across both constraints and learning rates.

> ☞ ***Key Takeaway:*** *(i) Weight decay is not needed if we use a manifold constraint optimizer. (ii) Unlike weight decay, manifold constraints regulate rotation angle from the very first iteration; while the Frobenius sphere enforces a static global rotation, the Spectral sphere induces an adaptive, anisotropic rotation.*

4.3 Weight-Update-Ratio Regulation Effect: Interplay with Training Precision

Low-precision master weight training can lose small increments during rounding. Standard remedies include FP32 master weights (Micikevicius et al., 2018), stochastic rounding, and compensated summation (Zamirai et al., 2020; Ozkara et al., 2025). Here, we examine how manifold constraints affect training under reduced master weight precision. We train the 240M and 600M Qwen3 models using TorchTitan (Liang et al., 2024), comparing the `float32` and `bfloat16` master weight precision settings. The BF16 configuration reduces the storage precision of parameters, gradients, and optimizer state. Figure 4 shows the validation trajectories. Muon's final validation-loss increases are 1.1470 and 0.9394 at the two scales, compared with 0.0172 and 0.0239 for MACRO-fro and 0.0317 and 0.0476 for MACRO-spec. MACRO has substantially less performance degradation than Muon. The experimental setting and tuned results can be found in Appendix D.3.

Update-Weight Ratio Alignment and update cancellation. The update-to-weight ratio (Wen et al., 2026; Xie et al., 2026) plays a crucial role here. To see this, we define $\rho_t := \|\Delta\mathbf{W}_t\|_F / \|\mathbf{W}_t\|_F = \|\Delta\mathbf{W}_t\|_{\mathrm{RMS}} / \|\mathbf{W}_t\|_{\mathrm{RMS}}$. For full-rank matrix-sign directions, the three optimizers give

$$\rho_t = \begin{cases} \eta_t \sqrt{d} / \|\mathbf{W}_t\|_F, & \text{Muon}, \\ \eta_t, & \text{MACRO-fro}, \\ \eta_t \sqrt{\kappa(\mathbf{W}_t)}, & \text{MACRO-spec}, \end{cases} \tag{6}$$

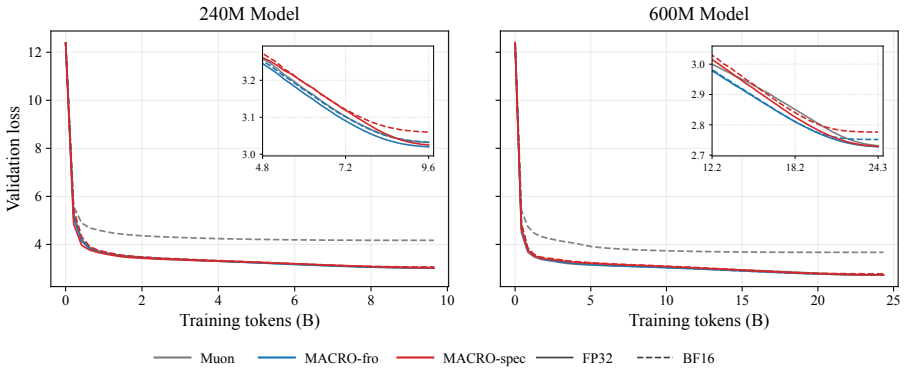


Figure 4: Validation-loss trajectories in the FP32 and BF16 configurations for the 240M and 600M models. Insets enlarge differences during the late stage of training.

where d and κ are defined in equation 3. The Frobenius ratio follows the learning rate directly; the spectral ratio is of the same order when κ remains bounded.

We now use these relative-update scales to analyze coordinate cancellation under BF16. For a scalar w and an increment δ, nearest rounding produces $Q_{\mathrm{BF16}}(w+\delta)$. The increment is canceled if $w+\delta$ remains within the rounding interval of w (Zamirai et al., 2020). A normal BF16 value, $|w| = (1.m)_2 2^E = (1.m_1, \ldots, m_7)_2 2^E$, has a unit of least precision (ULP) of 2^{E-7} and a half-ULP of 2^{E-8}. Any increment δ will be canceled under nearest rounding if $|\delta| < 2^{E-8}$, i.e., the increment will be canceled if $\frac{|\delta|}{|w|} < \frac{2^{E-8}}{(1.m)_2 2^E} = \frac{1}{(1.m)_2 2^8}$. Note that the quantity $\frac{1}{(1.m)_2 2^8}$ is always between $1/510$ and $1/256$.

Now we inspect optimizer update step. Although the parameter is a matrix rather than a scalar, we can still use the update-weight ratio ρ_t to approximate the average relative scale between the update and the weight. Without a manifold constraint, Muon may keep increasing the weight RMS norm during the early stage of training while the update RMS norm remains constant (Liu et al., 2025). Such misalignment will introduce update cancellation in Muon, leading to a much larger performance gap relative to full precision. Fig 5 empirically shows that Muon fails to apply the weight-decay update because $\eta\lambda \ll 1/510$, which leads to weight accumulation from the very first step (**?**). After about 2500 steps, the Frobenius norm of the key-projection weight under Muon increases to approximately 129. The update scale is $\eta|O_t|_{\mathrm{F}} = \Theta(0.01\sqrt{\min\{n,p\}}) = \Theta(0.32)$, so the update-weight ratio is approximately $\Theta(2\times 10^{-3}) \approx 1/510$. This tiny update-weight-ratio can easily lead to update cancellation, which corresponds to the performance degradation from a very early stage observed in Fig 4. On the other hand, both MACRO variants control the weight Frobenius norm and incur loss degradation only when η decays below a certain threshold during the late phase of training.

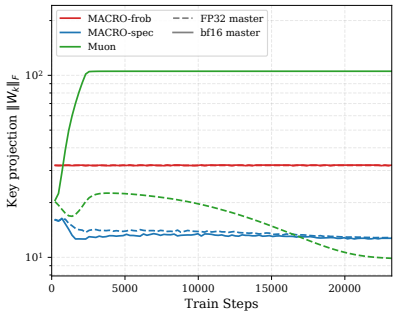


Figure 5: The $\|W_t\|_{\mathrm{F}}$ of Key projection parameter

To theoretically explain the loss degradation during the late stage, we build on the observation that weight matrices can be modeled as zero-mean Gaussian random matrices with i.i.d. entries (Huang et al., 2024; Kedia et al., 2024; Babaoglu et al., 2026). We then use two independent random variables, $w \sim \mathcal{N}(0, \|\mathbf{W}_t\|^2_{\mathrm{RMS}})$ and $\delta \sim \mathcal{N}(0, \|\Delta\mathbf{W}_t\|^2_{\mathrm{RMS}})$, to simulate update cancellation. Under this assumption, we can estimate the retained fraction of update magnitude R_t^{fro} under Frobenius Sphere as

$$R_t^{\mathrm{fro}} := \frac{\mathbb{E}|\delta\,\mathbf{1}\{|\delta| > \tau|w|\}|}{\mathbb{E}|\delta|} = \frac{1}{\sqrt{1+(\tau/\rho)^2}} \overset{\rho_t=\eta_t}{=} \left[1+\left(\frac{\tau}{\eta_t}\right)^2\right]^{-1/2}.$$

Under BF16 precision, let $\tau = 1/510 \approx 1.96\times 10^{-3}$. The above equation shows that the retained fraction of update magnitude depends only on the relation between τ and η_t. Under the cosine-decay learning-rate schedule, once the learning rate decays to near or even below τ, the retained fraction R_t^{fro}

also decreases, leading to update cancellation during the late stage of training in Fig 4. MACRO-spec exhibits a similar effect, as it also controls the Frobenius norm during training, as shown in Fig 5. The detailed proposition and discussion are provided in Appendix C.5 and Appendix D.3.

> ☞ ***Key Takeaway:*** *MACRO exhibits less performance degradation than Muon in BF16 master-weight training. Controlling the update-to-weight ratio provides a scale-based interpretation by analyzing update cancellation as the weight norm and learning rate change.*

5 Numerical Results

To evaluate the empirical performance of MACRO, we compare optimizers on standard Qwen3 architectures at the 240M, 600M, and 1.2B scales using TorchTitan (Liang et al., 2024) training pipeline. Across all model sizes, we train on the token budget using 40 Token Per Parameters(TPP) and 5 seeds. Detailed experimental settings are reported in Table 16 in Appendix D.4. Alongside Muon (Jordan, 2024), we evaluate MuonH and the Spectral Sphere Optimization (SSO) algorithm. To compare both sphere constraints, we extend SSO to the Frobenius sphere and MuonH to the spectral sphere, yielding FSO and MuonH-spec, respectively (Appendix B.2). Table 4 summarizes training loss and validation perplexity (PPL) and Table 17 in Appendix D.3 shows the optimizer efficiency.

Table 4: Training loss and validation perplexity (PPL) for Qwen3 models from 240M to 1.2B.

Optimizers	**240M**		**600M**		**1.2B**	
	Train	Val PPL	Train	Val PPL	Train	Val PPL
Muon (Baseline)	3.042	20.72 ± 0.010	2.755	15.37 ± 0.006	2.577	12.99 ± 0.004
MuonH-fro	3.033	20.55 ± 0.007	**2.735**	**15.23 ± 0.006**	2.569	12.92 ± 0.005
MuonH-spec	3.037	20.63 ± 0.014	2.742	15.33 ± 0.006	2.566	12.88 ± 0.006
FSO†	3.040	20.98 ± 0.013	2.744	15.33 ± 0.014	-	-
SSO†	3.035	20.59 ± 0.021	2.742	15.32 ± 0.007	-	-
MACRO-fro (Ours)	**3.031**	**20.50 ± 0.011**	2.739	15.29 ± 0.002	**2.563**	**12.85 ± 0.006**
MACRO-spec (Ours)	3.034	20.58 ± 0.011	2.740	15.30 ± 0.005	2.568	12.90 ± 0.009

† Scaling these double-loop algorithms requires custom CUDA kernels. Due to limited computational resources and the absence of such operator-level optimizations, we omit SSO and FSO for the 1.2B model.

We draw conclusions from these results: Firstly, both MACRO variants improve validation PPL over the Muon baseline at all three model scales. These results support the use of weight-norm constraints in standard architectures, consistent with the findings in Wen et al. (2026). From the efficiency perspective (Table 17), MACRO-fro shows only a small efficiency gap compared with Muon. Its model FLOPs utilization (MFU) and end-to-end (E2E) time decrease by 0.62 percentage points and increase by 1.6%, respectively. Secondly, two constraints exhibit similar overall performance. For MACRO, the validation PPL gap between them is at most 0.08 across the three scales. The Frobenius constraint performs better among all three scales. On the other hand, the Frobenius constraint is consistently more efficient than the spectral constraint. Finally, Compared to MuonH, MACRO achieves comparable performance: MACRO attains the lowest validation PPL at 240M and 1.2B. In contrast of MuonH, MACRO retains the convergence guarantee established in Section 3. Moreover, MACRO single-loop update achieves lower validation PPL than SSO and FSO at both evaluated scales, with the computational savings quantified in Table 17. Without specialized kernel optimizations, the MFU of MACRO-spec (35.15%) is 2.45 times that of SSO (14.35%), while MACRO-fro (39.06%) is 1.89 times that of FSO (20.72%).

6 Conclusions

In this paper, we study manifold constraints in LLM pre-training and their interplay with normalization layers, weight decay, and training precision. Our analysis motivates norm-specific radii under Gaussian initialization and reveals functional overlap between manifold constraints and RMSNorm. Our analysis also explains imply the regulation of weight rotation when using manifold constraints, and demonstrates that weight decay can be removed when manifold constraints are present. Relative-learning-rate control helps explain the smaller BF16 validation-loss penalty observed for MACRO compared with Muon. Furthermore, MACRO achieves validation losses comparable to, or better than, existing heuristic methods while still maintains convergence guarantee from Riemannian optimization principles.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools for implementing methods and interpret results. We have not used generative AI tools for helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, assisting with translation and Generating synthetic data sets, cleaning and reformat dataset, supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for summarizing existing literature and polishing writing. We have reviewed all AI-assisted work. We checked LLM-generated literature survey for research purpose. LLM-generated optimizer implementation code was verified and tested for correctness by two authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

(This section is **recommended** and does not count toward the page limit.)

In this work, we are committed to the reproducibility of our results. For the empirical results, we provide an anonymous repository link (https://anonymous.4open.science/r/matrix-optimizers-A841/) to the source code and also uploaded the source code in the supplementary materials. The detailed hyperparameters are included in the Appendix. For the theoretical results, a clear explanation and a complete proof are provided in the Appendix.

REFERENCES

Keller Jordan. Muon: An optimizer for hidden layers, 2024. URL https://github.com/KellerJordan/Muon.

Chao Ma, Wenbo Gong, Meyer Scetbon, and Edward Meeds. SWAN: SGD with normalization and whitening enables stateless LLM training, 2025. URL https://arxiv.org/abs/2412.13148.

Athanasios Glentis, Jiaxiang Li, Andi Han, and Mingyi Hong. Memory-efficient llm pretraining via minimalist optimizer design, 2026. URL https://arxiv.org/abs/2506.16659.

John Zhao. Towards a principled Muon under μP: Ensuring spectral conditions throughout training, 2026. URL https://arxiv.org/abs/2601.01306.

Xiaowen Jiang, Andrei Semenov, and Sebastian U. Stich. Enhancing LLM training via spectral clipping, 2026. URL https://arxiv.org/abs/2603.14315.

Md Rifat Arefin, Ravid Shwartz-Ziv, Ernie Chang, Chinnadhurai Sankar, Rylan Conway, Aristide Baratin, Adithya Sagar, and Patrick Huber. Learning in transformers under spectral constraints. In *ICLR 2026 Workshop on Geometry-grounded Representation Learning and Generative Modeling*, 2026.

Yufei Gu and Zeke Xie. Mano: Restriking manifold optimization for LLM training, 2026. URL https://arxiv.org/abs/2601.23000.

Laker Newhouse, R. Preston Hess, Franz Cesista, Andrii Zahorodnii, Jeremy Bernstein, and Phillip Isola. Training transformers with enforced Lipschitz constants, 2025. URL https://arxiv.org/abs/2507.13338.

Tian Xie, Haoming Luo, Haoyu Tang, Yiwen Hu, Jason Klein Liu, Qingnan Ren, Yang Wang, Wayne Xin Zhao, Rui Yan, Bing Su, Chong Luo, and Baining Guo. Controlled LLM training on spectral sphere, 2026. URL https://arxiv.org/abs/2601.08393.

Kaiwei Yang and Lexiao Lai. Manifold constrained steepest descent for smooth and closed-set optimization, 2026. URL https://arxiv.org/abs/2601.21487.

Hadi Mohaghegh Dolatabadi, Thalaiyasingam Ajanthan, Sameera Ramasinghe, Chamin P Hewa Koneputugodage, Shamane Siriwardhana, Violetta Shevchenko, Karol Pajak, James Snewin, Gil Avraham, and Alexander Long. NuMuon: Nuclear-norm-constrained Muon for compressible LLM training, 2026. URL https://arxiv.org/abs/2603.03597.

Jianlin Su. Fastest descent on a manifold: 4. Muon + spectral sphere, Aug 2025a. URL https://spaces.ac.cn/archives/11241. (In Chinese).

Jianlin Su. Fastest descent on a manifold: 2. Muon + orthogonal, Aug 2025b. URL https://spaces.ac.cn/archives/11215. (In Chinese).

Kaiyue Wen, Xingyu Dang, Kaifeng Lyu, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them ii: Hyperball optimization, 2026. URL https://arxiv.org/abs/2606.16899.

Jeremy Bernstein. Modular manifolds. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20250926. https://thinkingmachines.ai/blog/modular-manifolds/.

Biao Zhang and Rico Sennrich. Root mean square layer normalization, 2019. URL https://arxiv.org/abs/1910.07467.

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

Atli Kosson, Bettina Messmer, and Martin Jaggi. Rotational equilibrium: How weight decay balances learning across neural networks, 2024a. URL https://arxiv.org/abs/2305.17212.

Zhiyuan Li, Kaifeng Lyu, and Sanjeev Arora. Reconciling modern deep learning with traditional optimization analyses: The intrinsic learning rate, 2020. URL https://arxiv.org/abs/2010.02916.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1710.03740.

Pedram Zamirai, Jian Zhang, Christopher R. Aberger, and Christopher De Sa. Revisiting BFloat16 training, 2020. URL https://arxiv.org/abs/2010.06192. Revised March 2021.

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

Jianlin Su. Beyond MuP: 2. linear layers and steepest descent, Feb 2026a. URL https://spaces.ac.cn/archives/11605. (In Chinese).

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. URL https://arxiv.org/abs/1502.03167.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks, 2016. URL https://arxiv.org/abs/1602.07868.

Siyuan Qiao, Huiyu Wang, Chenxi Liu, Wei Shen, and Alan Yuille. Micro-batch training with batch-channel normalization and weight standardization, 2020. URL https://arxiv.org/abs/1903.10520.

Francesco D'Angelo, Maksym Andriushchenko, Aditya Varre, and Nicolas Flammarion. Why do we need weight decay in modern deep learning?, 2024. URL https://arxiv.org/abs/2310.04415.

Shuo Xie and Zhiyuan Li. Implicit bias of AdamW: ℓ_∞ norm constrained optimization, 2024. URL https://arxiv.org/abs/2404.04454.

Aaron Defazio. Why gradients rapidly increase near the end of training, 2025. URL https://arxiv.org/abs/2506.02285.

Lizhang Chen, Jonathan Li, and Qiang Liu. Muon optimizes under spectral norm constraints. *arXiv preprint arXiv:2506.15054*, 2025a.

Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

Per-Åke Wedin. Perturbation bounds in connection with singular value decomposition. *BIT Numerical Mathematics*, 12(1):99–111, 1972.

Kaan Ozkara, Tao Yu, and Youngsuk Park. Stochastic rounding for LLM training: Theory and practice. In *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pages 4402–4410. PMLR, 2025. URL https://proceedings.mlr.press/v258/ozkara25b.html.

Wanchao Liang, Tianyu Liu, Less Wright, Will Constable, Andrew Gu, Chien-Chin Huang, Iris Zhang, Wei Feng, Howard Huang, Junjie Wang, Sanket Purandare, Gokul Nadathur, and Stratos Idreos. TorchTitan: One-stop PyTorch native solution for production ready LLM pre-training, 2024. URL https://arxiv.org/abs/2410.06511.

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

Wei Huang, Yangdong Liu, Haotong Qin, Ying Li, Shiming Zhang, Xianglong Liu, Michele Magno, and Xiaojuan Qi. Billm: Pushing the limit of post-training quantization for llms. *arXiv preprint arXiv:2402.04291*, 2024.

Akhil Kedia, Mohd Abbas Zaidi, Sushil Khyalia, Jungho Jung, Harshith Goka, and Haejun Lee. Transformers get stable: An end-to-end signal propagation theory for language models. *arXiv preprint arXiv:2403.09635*, 2024.

Liza Babaoglu, Shuangyi Chen, and Ashish Khisti. Multi-bitwidth quantization for llms using additive codebooks. *arXiv preprint arXiv:2606.12876*, 2026.

Thomas Pethick, Wanyun Xie, Kimon Antonakopoulos, Zhenyu Zhu, Antonio Silveti-Falls, and Volkan Cevher. Training deep learning models with norm-constrained LMOs, 2025. URL https://arxiv.org/abs/2502.07529.

Runa Eschenhagen, Aaron Defazio, Tsung-Hsien Lee, Richard E. Turner, and Hao-Jun Michael Shi. Purifying Shampoo: Investigating Shampoo's heuristics by decomposing its preconditioner, 2025. URL https://arxiv.org/abs/2506.03595.

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

Donglin Zhan and James Anderson. Data-efficient and robust task selection for meta-learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 8056–8065, 2024. doi: 10.1109/CVPRW63382.2024.00805.

Hao-Jun Michael Shi, Tsung-Hsien Lee, Shintaro Iwasaki, Jose Gallego-Posada, Zhijing Li, Kaushik Rangadurai, Dheevatsa Mudigere, and Michael Rabbat. A distributed data-parallel pytorch implementation of the distributed Shampoo optimizer for training neural networks at-scale, 2023. URL https://arxiv.org/abs/2309.06497.

Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology. *arXiv preprint arXiv:2409.20325*, 2024.

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

Donglin Zhan, Leonardo F. Toso, and James Anderson. Coreset-based task selection for sample-efficient meta-reinforcement learning. In *2025 IEEE 64th Conference on Decision and Control (CDC)*, pages 4167–4173, 2025a. doi: 10.1109/CDC57313.2025.11312186.

Donglin Zhan, Haoting Zhang, Rhonda Righter, Zeyu Zheng, and James Anderson. Collaborative bayesian optimization via wasserstein barycenters. In *2025 IEEE 64th Conference on Decision and Control (CDC)*, pages 6284–6291, 2025b. doi: 10.1109/CDC57313.2025.11312220.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

Yang Liu, Jeremy Bernstein, Markus Meister, and Yisong Yue. Learning by turning: Neural architecture aware optimisation, 2021. URL https://arxiv.org/abs/2102.07227.

Ilya Loshchilov, Cheng-Ping Hsieh, Simeng Sun, and Boris Ginsburg. nGPT: Normalized transformer with representation learning on the hypersphere, 2024. URL https://arxiv.org/abs/2410.01131.

Yonggan Fu, Xin Dong, Shizhe Diao, Matthijs Van keirsbilck, Hanrong Ye, Wonmin Byeon, Yashaswi Karnati, Lucas Liebenwein, Hannah Zhang, Nikolaus Binder, Maksim Khadkevich, Alexander Keller, Jan Kautz, Yingyan Celine Lin, and Pavlo Molchanov. Nemotron-flash: Towards latency-optimal hybrid small language models, 2025. URL https://arxiv.org/abs/2511.18890.

Jörg K.H. Franke, Urs Spiegelhalter, Marianna Nezhurina, Jenia Jitsev, Frank Hutter, and Michael Hefenbrock. Learning in compact spaces with approximately normalized transformer, 2025. URL https://arxiv.org/abs/2505.22014.

Ruosi Wan, Zhanxing Zhu, Xiangyu Zhang, and Jian Sun. Spherical motion dynamics: Learning dynamics of normalized neural network using SGD and weight decay. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 21759–21770. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/326a8c055c0d04f5b06544665d8bb3ea-Abstract.html.

Weiyang Liu, Zhen Liu, Zhiding Yu, Bo Dai, Rongmei Lin, Yisen Wang, James M. Rehg, and Le Song. Decoupled networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Takeru Miyato, Sindy Löwe, Andreas Geiger, and Max Welling. Artificial kuramoto oscillatory neurons, 2024. URL https://arxiv.org/abs/2410.13821.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models, 2024. URL https://arxiv.org/abs/2312.02696.

Louis Owen, Abhay Kumar, Nilabhra Roy Chowdhury, and Fabian Güra. Variance control via weight rescaling in LLM pre-training, 2025. URL https://arxiv.org/abs/2503.17500.

Guodong Zhang, Chaoqi Wang, Bowen Xu, and Roger Grosse. Three mechanisms of weight decay regularization, 2018. URL https://arxiv.org/abs/1810.12281.

Zhenxun Zhuang, Mingrui Liu, Ashok Cutkosky, and Francesco Orabona. Understanding AdamW through proximal methods and scale-freeness, 2022. URL https://arxiv.org/abs/2202.00089.

Zhiyuan Li and Sanjeev Arora. An exponential learning rate schedule for deep learning, 2019. URL https://arxiv.org/abs/1910.07454.

Byeongho Heo, Sanghyuk Chun, Seong Joon Oh, Dongyoon Han, Sangdoo Yun, Gyuwan Kim, Youngjung Uh, and Jung-Woo Ha. AdamP: Slowing down the slowdown for momentum optimizers on scale-invariant weights, 2021. URL https://arxiv.org/abs/2006.08217.

Twan van Laarhoven. L2 regularization versus batch and weight normalization, 2017. URL https://arxiv.org/abs/1706.05350.

Atli Kosson, Bettina Messmer, and Martin Jaggi. Analyzing & reducing the need for learning rate warmup in GPT training, 2024b. URL https://arxiv.org/abs/2410.23922.

Zeke Xie, Zhiqiang Xu, Jingzhao Zhang, Issei Sato, and Masashi Sugiyama. On the overlooked pitfalls of weight decay and how to mitigate them: A gradient-norm perspective, 2024. URL https://arxiv.org/abs/2011.11152.

Ruihan Xu, Jiajin Li, and Yiping Lu. On the width scaling of neural optimizers under matrix operator norms i: Row/column normalization and hyperparameter transfer, 2026. URL https://arxiv.org/abs/2603.09952.

Jianlin Su. Beyond MuP: 4. ensuring parameter stability, Apr 2026b. URL https://spaces.ac.cn/archives/11729. (In Chinese).

Lizhang Chen, Bo Liu, Kaizhao Liang, and Qiang Liu. Lion secretly solves constrained optimization: As Lyapunov predicts, 2025b. URL https://arxiv.org/abs/2310.05898.

Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1737–1746. PMLR, 2015. URL https://proceedings.mlr.press/v37/gupta15.html.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. Scaling language models: Methods, analysis & insights from training Gopher, 2021. URL https://arxiv.org/abs/2112.11446. Version 2 revised January 2022.

Tao Yu, Gaurav Gupta, Karthick Gopalswamy, Amith Mamidala, Hao Zhou, Jeffrey Huynh, Youngsuk Park, Ron Diamant, Anoop Deoras, and Luke Huan. Collage: Light-weight low-precision strategy for llm training, 2024. URL https://arxiv.org/abs/2405.03637.

Mahdi Nikdan, Amir Zandieh, Dan Alistarh, and Vahab Mirrokni. ECO: Quantized training without full-precision master weights, 2026. URL https://arxiv.org/abs/2601.22101.

Xiaoyuan Liang, Sebastian Loeschcke, Mads Toftrup, and Anima Anandkumar. M+Adam: Low-precision training via additive-multiplicative optimization. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2607.10611.

Zekai Shang. Auditing invisible weight updates with reference traces, 2026. URL https://arxiv.org/abs/2607.09800v4. Version 4, August 27, 2026.

Erfan Miahi and Eugene Belilovsky. Understanding and exploiting weight update sparsity for communication-efficient distributed RL, 2026. URL https://arxiv.org/abs/2602.03839.

Adrian S Lewis and Hristo S Sendov. Nonsmooth analysis of singular values. part i: Theory. *Set-Valued Analysis*, 13(3):213–241, 2005.

John M Lee. Smooth manifolds. In *Introduction to smooth manifolds*, pages 1–29. Springer, 2003.

G. W. Stewart and Ji-Guang Sun. *Matrix Perturbation Theory*. Computer Science and Scientific Computing. Academic Press, 1990.

A Related Works

Muon optimizers and manifold-constrained variants. Muon (Jordan, 2024; Pethick et al., 2025) introduced a spectral-norm steepest descent update for weight matrices, achieving strong empirical performance in LLM pre-training. It is a parallel work of matrix structured preconditioning methods such as Shampoo (Eschenhagen et al., 2025; Gupta et al., 2018; Zhan and Anderson, 2024; Shi et al., 2023). Bernstein and Newhouse (2024) provided theoretical grounding by interpreting Muon as an approximate Newton's method on the Stiefel manifold. Subsequent work has extended Muon in several directions: Zhao (2026) ensured spectral conditions under the μP parameterization; An et al. (2025) proposed ASGO, which generalizes structured gradient preconditioning across different matrix decompositions; and Dolatabadi et al. (2026) introduced NuMuon, which replaces the spectral constraint with a nuclear norm constraint to encourage low-rank structure during training. Some recent studies on structure-aware optimizers Ma et al. (2025); Glentis et al. (2026); Zhan et al. (2025a;b) also consider row/column-wise normalization for the gradient. In parallel, several methods have explored explicit manifold constraints for LLM training. SSO (Xie et al., 2026) formulates training on the spectral sphere with provable double-loop projections, while Mano (Gu and Xie, 2026) targets oblique manifolds and demonstrates competitive pre-training results. MuonH (Wen et al., 2026) constrains weights to a Frobenius sphere via heuristic normalization after each Muon step. Yang and Lai (2026) study Stiefel manifold steepest descent from a general Riemannian optimization perspective.

Training deep learning models with constraints. Imposing structural constraints on neural network weights has a long history. Spectral normalization (Miyato et al., 2018) was originally proposed to stabilize GAN training by controlling the Lipschitz constant of discriminator layers. Newhouse et al. (2025) extended this idea to enforce Lipschitz bounds in transformers. Pethick et al. (2025) trained deep networks with norm-constrained linear minimization oracles (LMOs), offering a Frank-Wolfe perspective on constrained optimization. Liu et al. (2021) proposed optimization directly on the Stiefel manifold by decomposing updates into rotations, and Arefin et al. (2026) studied spectral constraints for feature learning in transformers. At the architecture level, nGPT (Loshchilov et al., 2024) normalizes all representations to lie on a hypersphere, and Nemotron-Flash (Fu et al., 2025) builds on this design for latency-efficient models. Franke et al. (2025) further explore approximately normalized transformers that learn in compact parameter spaces. Bernstein (2025) provides a conceptual framework connecting modular network design to manifold structure. While these methods each address specific constraint types, none systematically investigates how different manifold geometries interact with normalization layers and weight decay during LLM pre-training.

Normalization layers and their variants. Batch Normalization (Ioffe and Szegedy, 2015) introduced per-channel normalization of activations and was followed by Layer Normalization (Ba et al., 2016), Group Normalization (Wu and He, 2018), and RMSNorm (Zhang and Sennrich, 2019). In particular, RMSNorm is now standard in transformer-based LLMs. In the weight space, Weight Normalization (Salimans and Kingma, 2016) decouples the magnitude and direction of weight vectors, and Weight Standardization (Qiao et al., 2020) normalizes the rows of weight matrices to improve micro-batch training. Wan et al. (2021) analyze how the combination of batch normalization and weight decay induces spherical motion dynamics, revealing that the effective learning rate is governed by the angular update rather than the Euclidean step size. Liu et al. (2018) propose Decoupled Networks, which separate the angular and radial components of inner products in linear layers. Miyato et al. (2024) introduce Artificial Kuramoto Oscillatory Neurons, whose phase-based representations naturally reside on the unit circle. Karras et al. (2024) demonstrate that weight normalization of convolution layers stabilizes diffusion model training by controlling magnitude growth. Owen et al. (2025) propose weight rescaling to control activation variance during LLM pre-training. Our work

demonstrates that manifold constraints on weight matrices can formally subsume the stabilizing role of explicit normalization layers, enabling normalization-free linear blocks without sacrificing training stability.

Weight decay and its dynamics. Weight decay was originally introduced as ℓ_2 regularization to improve generalization (Zhang et al., 2018), and Loshchilov and Hutter (2019) showed that decoupling it from the gradient (as in AdamW) leads to substantially different behavior than classical ℓ_2 penalties. Zhuang et al. (2022) further clarify this distinction through proximal operator interpretations. A key insight from Li et al. (2020) is that for scale-invariant networks, weight decay controls the *intrinsic learning rate*—the ratio of the update norm to the weight norm—rather than acting as a direct regularizer. Li and Arora (2019) further show that this mechanism induces an implicit exponential learning rate schedule. Kosson et al. (2024a) formalize this as *rotational equilibrium*, showing that weight decay balances the angular velocity of weight updates across layers. Heo et al. (2021) propose AdamP, which explicitly removes the radial component of momentum updates to prevent the slowdown effect on scale-invariant weights. van Laarhoven (2017) analyze the interaction between ℓ_2 regularization and batch/weight normalization, showing that their combined effect reduces to a modulation of the effective step size. Kosson et al. (2024b) connect weight decay dynamics to learning rate warmup, demonstrating that careful weight initialization can reduce the need for warmup in GPT training. Xie et al. (2024) identify gradient-norm pathologies caused by weight decay, and D'Angelo et al. (2024) revisit why weight decay remains necessary in modern architectures, attributing its benefit primarily to scale invariance. Xu et al. (2026), Su (2026b) ,Chen et al. (2025b) and Chen et al. (2025a) show that weight decay together with bounded update directions keeps the parameters in a uniformly bounded ball. Finally, Defazio (2025) show that gradient norms increase rapidly near the end of training, a phenomenon linked to the decay of weight norms. Our analysis shows that manifold constraints intrinsically govern the quantities that weight decay controls heuristically—locking the relative learning rate and rotation angle—and thus provide a principled geometric alternative to weight decay tuning.

Low-precision training and update cancellation. Low-precision training requires preserving optimizer updates that can be lost during parameter rounding. Micikevicius et al. (2018) retain FP32 master weights for mixed-precision training, while Gupta et al. (2015) demonstrate the importance of stochastic rounding in low-precision fixed-point training. For BF16, Zamirai et al. (2020) identify update cancellation as a key limitation and study stochastic rounding and Kahan summation. Gopher (Rae et al., 2021) uses stochastic rounding for BF16 parameters but reports a remaining gap relative to mixed precision, and Ozkara et al. (2025) provide further theoretical and empirical analysis of stochastic rounding for LLMs. Collage (Yu et al., 2024) compensates numerical errors using multi-component low-precision representations and explicitly discusses the loss of small weight-decay updates. More recently, ECO (Nikdan et al., 2026) incorporates weight-quantization errors into optimizer momentum, whereas M+Adam (Liang et al., 2026) combines additive and multiplicative updates to address precision-dependent stagnation. Shang (2026) examines coordinatewise update invisibility and the limitations of forecasts based on high-precision reference traces. Related BF16 update sparsity is exploited for distributed RL communication by PULSE (Miahi and Belilovsky, 2026). Our work focuses on how manifold constraints regulate the update-to-weight ratio and thereby affect susceptibility to BF16 update cancellation. We use a Gaussian coordinate-threshold model to interpret the observed precision gap, without introducing stochastic rounding or explicit error compensation.

B Algorithm Supplementary Materials

B.1 Tangent Space Projection and Projectional Retraction

Tangent Space Projection. A critical component of our single-loop algorithm is the orthogonal projection of the momentum M_t (or gradient) onto the tangent space of the constraint manifold. We define the projected vector Φ_t^* as the solution to the following proximity problem:

$$\Phi_t^* = \arg \min_{\Phi \in \mathrm{T}_{\mathcal{M}} \mathbf{W}_t} \|\Phi - M_t\|_F^2.$$

Given that the constraints we consider can be locally characterized by a scalar function $h(W) = c$, the tangent space is a linear subspace. This yields a simple closed-form solution:

$$\Phi_t^* = M_t - \frac{\langle M_t, \Theta \rangle}{\|\Theta\|_F^2}\Theta,$$

where $\Theta = \nabla_W h(W)$ represents the normal vector to the constraint surface. This projection is highly computationally efficient, as it typically involves only fundamental matrix-vector or inner products. The specific projection operators for our four constraints are detailed below:

- Frobenius Sphere Constraint ($\mathcal{M}_F$): The normal vector is the weight matrix itself, $\Theta = W$. The projection is given by:
$$\Phi_t^* = M_t - \frac{\langle M_t, W \rangle}{\|W\|_F^2}W.$$
- Spectral Sphere Constraint ($\mathcal{M}_S$): The normal vector is $\Theta = u_1 v_1^\top$, where u_1 and v_1 are the leading singular vectors of weight matrices $\mathbf{W}$. Since $\|\Theta\|_F^2 = \mathrm{Trace}(v_1 u_1^\top u_1 v_1^\top) = 1$, the projection simplifies to:
$$\Phi_t^* = M_t - \mathrm{Trace}(M_t^\top u_1 v_1^\top)u_1 v_1^\top = M_t - (u_1^\top M_t v_1)u_1 v_1^\top.$$
- Output Oblique Manifold Constraint ($\mathcal{M}_{O_{\text{out}}}$): Since the constraint is applied to each row independently, we project each row vector $M_t[i,:]$ onto the orthogonal complement of the corresponding weight row $W_{i,:}$:
$$\Phi_t^*[i,:] = M_t[i,:] - \frac{M_t[i,:]W_{i,:}^\top}{\|W_{i,:}\|_2^2}W_{i,:}.$$
- Input Oblique Manifold Constraint ($\mathcal{M}_{O_{\text{in}}}$): Symmetrically, the constraint applies to the columns. We project each column vector $M_t[:,j]$ independently:
$$\Phi_t^*[:,j] = M_t[:,j] - \frac{W_{:,j}^\top M_t[:,j]}{\|W_{:,j}\|_2^2}W_{:,j}.$$

Projectional Retraction. The second critical component of our single-loop algorithm is the retraction of the update $\tilde{\mathbf{W}}_t := \mathbf{W}_t - \eta_t \tilde{\nabla}_t$ to the constraint manifold. Again we use the projectional retraction:

$$\mathbf{W}_{t+1} = \mathcal{P}_{\mathcal{M}}(\tilde{\mathbf{W}}_t) := \arg\min_{\mathbf{W}\in\mathcal{M}} \|\mathbf{W} - \tilde{\mathbf{W}}_t\|_F^2.$$

The specific projectional retraction operators for our four constraints are detailed below:

- Frobenius Sphere Constraint ($\mathcal{M}_F(R)$): The projection is given by:
$$\mathbf{W}_{t+1} = \mathcal{P}_{\mathcal{M}}(\tilde{\mathbf{W}}_t) = R \cdot \tilde{\mathbf{W}}_t / \|\tilde{\mathbf{W}}_t\|_F.$$
- Spectral Sphere Constraint ($\mathcal{M}_S(R)$): The projection is given by:
$$\mathbf{W}_{t+1} = \mathcal{P}_{\mathcal{M}}(\tilde{\mathbf{W}}_t) = U\Sigma' V^\top,$$
where $U\Sigma V^\top = \tilde{\mathbf{W}}_t$ is the singular value decomposition of $\tilde{\mathbf{W}}_t$ and Σ' is in the following structure: if Σ has elements that are greater than R, then clip them to R; otherwise if all elements in Σ are smaller than R, then turn the largest element of Σ to R. However, this operation is quite expensive to compute in practice, and we **use the following approximate operation in all our experiments**:
$$\mathbf{W}_{t+1} = R \cdot \tilde{\mathbf{W}}_t / \|\tilde{\mathbf{W}}_t\|_2$$
to approximately project the matrix $\tilde{\mathbf{W}}_t$ back to the spectral manifold.
- Output Oblique Manifold Constraint ($\mathcal{M}_{O_{\text{out}}}$): Since the constraint is applied to each row independently, the projectional retraction is simply divide each row with its own norm, and multiply with R.
- Input Oblique Manifold Constraint ($\mathcal{M}_{O_{\text{in}}}$): Symmetrically, the constraint applies to the columns. The projectional retraction is simply divide each column with its own norm, and multiply with R.

B.2 FROBENIUS SPHERE OPTIMIZATION AND MUONH-SPEC

Inspired by (Xie et al., 2026), we introduce the Frobenius Sphere Optimization(FSO) by considering the following subproblem:

$$\begin{aligned} \max_{\mathbf{\Phi}} \quad & \langle \mathbf{G}, \mathbf{\Phi} \rangle \\ \text{s.t.} \quad & \|\mathbf{\Phi}\|_2 = 1, \quad \|\mathbf{W} - \eta R \mathbf{\Phi}\|_F = \|\mathbf{W}\|_F = R, \end{aligned} \tag{7}$$

where $\mathbf{G}$ is the stochastic gradient and $\mathbf{W}$ is the weight matrix. Using the first-order Taylor's expansion of the Frobenius norm, we have:

$$\|\mathbf{W} - \eta R \mathbf{\Phi}\|_F = \|\mathbf{W}\|_F - \eta \langle \mathbf{W}, \mathbf{\Phi} \rangle + \mathcal{O}(\eta^2).$$

Using the first order approximation, problem equation 7 reduces to:

$$\max_{\mathbf{\Phi}} \langle \mathbf{G}, \mathbf{\Phi} \rangle \quad \text{s.t.} \quad \|\mathbf{\Phi}\|_2 = 1, \quad \langle \mathbf{W}, \mathbf{\Phi} \rangle = 0. \tag{8}$$

To solve Eq. 8, we consider its Lagrangian relaxation:

$$\max_{\mathbf{\Phi}} \langle \mathbf{G} + \lambda \mathbf{W}, \mathbf{\Phi} \rangle \quad \text{s.t.} \quad \|\mathbf{\Phi}\|_2 = 1.$$

Following Theorem A.1 in (Xie et al., 2026), the closed-form solution to this relaxed problem is:

$$\mathbf{\Phi}(\lambda) = \arg \max_{\|\mathbf{\Phi}\|_2 = 1} \langle \mathbf{G} + \lambda \mathbf{W}, \mathbf{\Phi} \rangle = \operatorname{msign}(\mathbf{G} + \lambda \mathbf{W}).$$

To satisfy the tangent space feasibility condition in Eq. 8, we require $\langle \mathbf{W}, \mathbf{\Phi}(\lambda^*) \rangle = 0$. Theorem A.2 in Xie et al. (2026) establishes that the multiplier λ is monotonic under the spectral sphere constraint. We show that this monotonicity also holds for the Frobenius sphere constraint:

Lemma 1. *The function*

$$h(\lambda) = \langle \mathbf{W}, \mathbf{\Phi}(\lambda) \rangle = \langle \mathbf{W}, \operatorname{msign}(\mathbf{G} + \lambda \mathbf{W}) \rangle$$

is monotonically non-decreasing with respect to λ. Moreover, there exists a root $\lambda^* \in \mathbb{R}$ such that $h(\lambda^*) = 0$, and any such root rigorously satisfies the bound $|\lambda^*| \leq \frac{2\|\mathbf{G}\|_*}{\|\mathbf{W}\|_*}$.

Proof. We first prove that $h(\lambda)$ is a monotonic non-decreasing function. By the Theorem A.1 in Xie et al. (2026), we have:

$$\operatorname{msign}(\mathbf{X}) = \operatorname{argmax}_{\|\mathbf{T}\|_2 = 1} \langle \mathbf{X}, \mathbf{T} \rangle. \tag{9}$$

Consider two arbitrary scalar values $\lambda_2 > \lambda_1$. By the definition of $\mathbf{\Phi}(\lambda) = \operatorname{msign}(\mathbf{G} + \lambda \mathbf{W})$ as the precise maximizer of the inner product $\langle \mathbf{G} + \lambda \mathbf{W}, \cdot \rangle$ over the unit spectral ball, we establish the following two cross-inequalities:

$$\begin{aligned} & \langle \mathbf{G} + \lambda_1 \mathbf{W}, \mathbf{\Phi}(\lambda_1) \rangle \geq \langle \mathbf{G} + \lambda_1 \mathbf{W}, \mathbf{\Phi}(\lambda_2) \rangle \\ \Rightarrow \quad & \langle \mathbf{G}, \mathbf{\Phi}(\lambda_1) \rangle + \lambda_1 \langle \mathbf{W}, \mathbf{\Phi}(\lambda_1) \rangle \geq \langle \mathbf{G}, \mathbf{\Phi}(\lambda_2) \rangle + \lambda_1 \langle \mathbf{W}, \mathbf{\Phi}(\lambda_2) \rangle, \end{aligned} \tag{10}$$

$$\begin{aligned} & \langle \mathbf{G} + \lambda_2 \mathbf{W}, \mathbf{\Phi}(\lambda_2) \rangle \geq \langle \mathbf{G} + \lambda_2 \mathbf{W}, \mathbf{\Phi}(\lambda_1) \rangle \\ \Rightarrow \quad & \langle \mathbf{G}, \mathbf{\Phi}(\lambda_2) \rangle + \lambda_2 \langle \mathbf{W}, \mathbf{\Phi}(\lambda_2) \rangle \geq \langle \mathbf{G}, \mathbf{\Phi}(\lambda_1) \rangle + \lambda_2 \langle \mathbf{W}, \mathbf{\Phi}(\lambda_1) \rangle. \end{aligned} \tag{11}$$

Summing Equation (10) and Equation (11), the cross-terms involving $\mathbf{G}$ naturally cancel out, allowing us to derive:

$$\begin{aligned} & \lambda_1 \langle \mathbf{W}, \mathbf{\Phi}(\lambda_1) \rangle + \lambda_2 \langle \mathbf{W}, \mathbf{\Phi}(\lambda_2) \rangle \geq \lambda_1 \langle \mathbf{W}, \mathbf{\Phi}(\lambda_2) \rangle + \lambda_2 \langle \mathbf{W}, \mathbf{\Phi}(\lambda_1) \rangle \\ \Rightarrow \quad & (\lambda_2 - \lambda_1) \langle \mathbf{W}, \mathbf{\Phi}(\lambda_2) \rangle \geq (\lambda_2 - \lambda_1) \langle \mathbf{W}, \mathbf{\Phi}(\lambda_1) \rangle. \end{aligned} \tag{12}$$

Given that $\lambda_2 - \lambda_1 > 0$, we can divide both sides by $\lambda_2 - \lambda_1$ to obtain $\langle \mathbf{W}, \mathbf{\Phi}(\lambda_2) \rangle \geq \langle \mathbf{W}, \mathbf{\Phi}(\lambda_1) \rangle$. This algebraically confirms that $h(\lambda_2) \geq h(\lambda_1)$, thus $h(\lambda)$ is strictly monotonic non-decreasing in λ.

We next localize the root λ^*. For any $\lambda > \frac{2\|\mathbf{G}\|_*}{\|\mathbf{W}\|_*} > 0$ and setting $\mathbf{T} = \mathbf{\Phi}(\lambda) = \operatorname{msign}(\mathbf{G} + \lambda \mathbf{W})$, the definition in Equation (9) ensures $\|\mathbf{T}\|_2 \leq 1$. We have the relation:

$$\lambda h(\lambda) = \lambda \langle \mathbf{W}, \mathbf{T} \rangle = \langle \mathbf{G} + \lambda \mathbf{W}, \mathbf{T} \rangle - \langle \mathbf{G}, \mathbf{T} \rangle. \tag{13}$$

By the duality of norms, the inner product is bounded by $\langle \mathbf{G}, \mathbf{T} \rangle \leq \|\mathbf{G}\|_* \|\mathbf{T}\|_2 \leq \|\mathbf{G}\|_*$. Simultaneously, because $\mathbf{T}$ maximizes the inner product with $\mathbf{G}+\lambda\mathbf{W}$, we have $\langle \mathbf{G}+\lambda\mathbf{W}, \mathbf{T} \rangle = \|\mathbf{G}+\lambda\mathbf{W}\|_*$. Applying the reverse triangle inequality:

$$\|\mathbf{G} + \lambda\mathbf{W}\|_* \geq \|\lambda\mathbf{W}\|_* - \|\mathbf{G}\|_* = \lambda\|\mathbf{W}\|_* - \|\mathbf{G}\|_* \tag{14}$$

Substituting Equation (14) and the dual norm bound into Equation (13), we derive:

$$\lambda h(\lambda) \geq (\lambda\|W\|_* - \|\mathbf{G}\|_*) - \|\mathbf{G}\|_* = \lambda\|W\|_* - 2\|\mathbf{G}\|_*. \tag{15}$$

Since we assumed $\lambda > \frac{2\|\mathbf{G}\|_*}{\|\mathbf{W}\|_*}$, it follows that $\lambda h(\lambda) > 0$. Because λ is positive, this implies $h(\lambda) > 0$. Through symmetrical logical deduction, if $\lambda < \frac{-2\|\mathbf{G}\|_*}{\|\mathbf{W}\|_*}$, we obtain $\lambda h(\lambda) > 0$, which inherently forces $h(\lambda) < 0$ since λ is strictly negative.

Because $h(\lambda)$ represents a monotonically non-decreasing mapping (as established above) that transitions from a negative state at $\lambda = \frac{-2\|\mathbf{G}\|_*}{\|W\|_*}$ to a positive state at $\lambda = \frac{2\|\mathbf{G}\|_*}{\|W\|_*}$, the Intermediate Value Property mathematically guarantees the existence of at least one root λ^* satisfying $h(\lambda^*) = 0$. Consequently, any valid root must reside within the bounded interval $[\frac{-2\|\mathbf{G}\|_*}{\|W\|_*}, \frac{2\|\mathbf{G}\|_*}{\|W\|_*}]$, proving $|\lambda^*| \leq \frac{2\|\mathbf{G}\|_*}{\|W\|_*}$. □

Leveraging Lemma 1, we can solve the subproblem in Eq. 8 via bisection search. This forms the basis of their double-loop algorithms: Frobenius Sphere Optimization (FSO), as summarized in Algorithm 2.

For MuonH, Wen et al. (2026) proposed Muon-Hyperball method under Frobenius Sphere based on the update step:

$$\mathbf{W}_{t+1} = \mathcal{P}_{\mathcal{M}}\left(\mathbf{W}_t - \eta R \cdot \text{Normalize}\left(\text{msign}(M_t)\right)\right).$$

According to the projection Retraction we mentioned in B.1 we can get the MuonH-spec using the following update step:

$$\mathbf{W}_{t+1} = \frac{R \cdot (\mathbf{W}_t - \eta R\,\text{msign}(M_t))}{\|\mathbf{W}_t - \eta R\,\text{msign}(M_t)\|_2}.$$

Algorithm 2 Frobenius Sphere Optimization (FSO)

Input: Weights $\mathbf{W}_t$, Learning rate η_t, Hyperparameters β, R
Output: Updated Weights $\mathbf{W}_{t+1}$
for $t = 0, 1, \ldots$ **do**
  $G_t = \nabla\ell(\mathbf{W}_t, \xi_t)$
  $M_t = \beta M_{t-1} + (1-\beta)G_t$
  Define $h(\lambda) = \langle \mathbf{W}_t, \text{msign}(M_t + \lambda\mathbf{W}_t) \rangle$
  $\lambda_t^* \leftarrow \text{Bisection}(h, \epsilon)$ ▷ Find the multiplier λ
  $O_t \leftarrow \text{msign}(M_t + \lambda_t^*\mathbf{W}_t)$ ▷ Compute the update direction
  $\widetilde{\nabla}_t = R \cdot O_t / (\|O_t\|_F + \epsilon)$ ▷ Normalization and Scaling
  $\mathbf{W}_{t+1} = \mathcal{P}_{\mathcal{M}}\left(\mathbf{W}_t - \eta_t \cdot \widetilde{\nabla}_t\right)$ ▷ Descent Step and Manifold Projection
end for

C Theoretical Supplementary Materials

C.1 Assumption and Proof for Theorem 1

In this subsection we provide theoretical proof of the MACRO (Algorithm 1) convergence. We firstly listed two standard assumption:

Assumption 1. *We assume that $\mathcal{M}$ is a compact C^3 manifold.*[†]

[†]See Appendix C.1 and Yang and Lai (2026) for more discussion.

Assumption 2. *We assume the loss function $\mathcal{L}$ is lower bounded by $\mathcal{L}^*$, and has L-Lipschitz continuous gradient, and the stochastic loss $\ell(W,\xi)$ satisfies: (i) the stochastic gradient is unbiased, i.e. $\mathbb{E}_\xi \nabla \ell(W,\xi) = \nabla \mathcal{L}(W)$; (ii) the variance is bounded, i.e., $\mathbb{E}_\xi \|\nabla \ell(W,\xi) - \nabla \mathcal{L}(W)\|_F^2 \le \sigma^2$.*

Then, we prove the convergence result in Theorem 1, which holds for Frobenius, input and output Oblique manifolds.[‡]

Proof of Theorem 1. This proof follows Yang and Lai (2026). By Assumption 1 and 2, for any $\mathbf{W} \in \mathcal{M}$ and G we have[§]

$$\mathcal{L} \circ \mathcal{P}_{\mathcal{M}}(\mathbf{W}+G) \le \mathcal{L}(\mathbf{W}) + \langle \nabla_{\mathcal{M}} \mathcal{L}(\mathbf{W}), G\rangle + \frac{L}{2}\|G\|_F^2, \tag{16}$$

where $\nabla_{\mathcal{M}}$ denotes the Riemannian gradient. Now from the algorithm we get

$$\begin{aligned}\mathcal{L}(\mathbf{W}_{t+1}) = \mathcal{L} \circ \mathcal{P}_{\mathcal{M}}(\mathbf{W} - \eta_t \tilde{\nabla}_t) \le& \mathcal{L}(\mathbf{W}_t) - \eta_t \langle \nabla_{\mathcal{M}} \mathcal{L}(\mathbf{W}_t), \tilde{\nabla}_t\rangle + \frac{L\eta_t^2}{2}\|\tilde{\nabla}_t\|_F^2 \\ =& \mathcal{L}(\mathbf{W}_t) - \langle \nabla_{\mathcal{M}} \mathcal{L}(\mathbf{W}_t), \tilde{\nabla}_t\rangle + \frac{R^2 L\eta_t^2}{2}\left(\frac{\|O_t\|_F}{\|O_t\|_{\mathcal{M}} + \epsilon}\right)^2.\end{aligned}$$

Note that $\|O_t\|_2 = 1$, therefore $1 \le \|O_t\|_F^2 \le d$ with $d := \min\{D_{\text{in}}, D_{\text{out}}\}$. Moreover, we may assume a uniform bound $c_0\|A\|_F \le \|A\|_{\mathcal{M}} \le c_1\|A\|_F$. Therefore,

$$\mathcal{L}(\mathbf{W}_{t+1}) \le \mathcal{L}(\mathbf{W}_t) - \tilde{c}\eta_t \langle \nabla_{\mathcal{M}} \mathcal{L}(\mathbf{W}_t), O_t\rangle + \frac{\tilde{c}^2 L\eta_t^2}{2} d,$$

where we denote $\tilde{c} := \frac{R}{c_0+\epsilon}$. Further, we have

$$\begin{aligned}&\mathcal{L}(\mathbf{W}_{t+1}) \\ \le& \mathcal{L}(\mathbf{W}_t) - \tilde{c}\eta_t \langle \nabla_{\mathcal{M}} \mathcal{L}(\mathbf{W}_t), O_t\rangle + \frac{\tilde{c}^2 L\eta_t^2}{2} d \\ =& \mathcal{L}(\mathbf{W}_t) - \tilde{c}\eta_t \left(\langle \nabla_{\mathcal{M}} \mathcal{L}(\mathbf{W}_t) - \text{proj}_{\text{T}_{\mathcal{M}}\mathbf{W}_t}(M_t), O_t\rangle + \langle \text{proj}_{\text{T}_{\mathcal{M}}\mathbf{W}_t}(M_t), O_t\rangle\right) + \frac{\tilde{c}^2 L\eta_t^2}{2} d.\end{aligned}$$

Since $O_t = \text{msign}(\Phi_t)$ with $\Phi_t = \text{proj}_{\text{T}_{\mathcal{M}}\mathbf{W}_t}(M_t)$, we know that $\langle \text{proj}_{\text{T}_{\mathcal{M}}\mathbf{W}_t}(M_t), O_t\rangle = \|\Phi_t\|_*$, therefore (Denote the dual norm of $\|\cdot\|_{\mathcal{M}}$ as $\|\cdot\|_{*,\mathcal{M}}$)

$$\begin{aligned}&\mathcal{L}(\mathbf{W}_{t+1}) \\ \le& \mathcal{L}(\mathbf{W}_t) + \tilde{c}\eta_t \|\nabla_{\mathcal{M}} \mathcal{L}(\mathbf{W}_t) - \text{proj}_{\text{T}_{\mathcal{M}}\mathbf{W}_t}(M_t)\|_* - \tilde{c}\eta_t \|\Phi_t\|_* + \frac{\tilde{c}^2 L\eta_t^2}{2} d.\end{aligned}$$

By Triangle inequality, $-\|\Phi_t\|_* \le -\|\nabla_{\mathcal{M}} \mathcal{L}(\mathbf{W}_t)\|_* + \|\nabla_{\mathcal{M}} \mathcal{L}(\mathbf{W}_t) - \text{proj}_{\text{T}_{\mathcal{M}}\mathbf{W}_t}(M_t)\|_*$, we have

$$\begin{aligned}&\mathcal{L}(\mathbf{W}_{t+1}) \\ \le& \mathcal{L}(\mathbf{W}_t) + \tilde{c}\eta_t \|\nabla_{\mathcal{M}} \mathcal{L}(\mathbf{W}_t) - \text{proj}_{\text{T}_{\mathcal{M}}\mathbf{W}_t}(M_t)\|_* - \tilde{c}\eta_t \|\nabla_{\mathcal{M}} \mathcal{L}(\mathbf{W}_t)\|_* + \frac{\tilde{c}^2 L\eta_t^2}{2} d,\end{aligned}$$

i.e.,

$$\begin{aligned}&\tilde{c}\eta_t \|\nabla_{\mathcal{M}} \mathcal{L}(\mathbf{W}_t)\|_* \\ \le& (\mathcal{L}(\mathbf{W}_t) - \mathcal{L}(\mathbf{W}_{t+1})) + \tilde{c}\eta_t \|\nabla_{\mathcal{M}} \mathcal{L}(\mathbf{W}_t) - \text{proj}_{\text{T}_{\mathcal{M}}\mathbf{W}_t}(M_t)\|_* + \frac{\tilde{c}^2 L\eta_t^2}{2} d.\end{aligned} \tag{17}$$

It remains to bound $\|\nabla_{\mathcal{M}} \mathcal{L}(\mathbf{W}_t) - \text{proj}_{\text{T}_{\mathcal{M}}\mathbf{W}_t}(M_t)\|_*$. Since $\nabla_{\mathcal{M}} \mathcal{L}(\mathbf{W}_t) = \text{proj}_{\text{T}_{\mathcal{M}}\mathbf{W}_t}(\nabla \mathcal{L}(\mathbf{W}_t))$, we have

$$\begin{aligned}&\|\nabla_{\mathcal{M}} \mathcal{L}(\mathbf{W}_t) - \text{proj}_{\text{T}_{\mathcal{M}}\mathbf{W}_t}(M_t)\|_* = \|\text{proj}_{\text{T}_{\mathcal{M}}\mathbf{W}_t}(\nabla \mathcal{L}(\mathbf{W}_t) - M_t)\|_* \\ \le& \sqrt{d}\|\text{proj}_{\text{T}_{\mathcal{M}}\mathbf{W}_t}(\nabla \mathcal{L}(\mathbf{W}_t) - M_t)\|_F \qquad \text{(WLOG assume full-rankness)} \\ \le& \sqrt{d}\|\nabla \mathcal{L}(\mathbf{W}_t) - M_t\|_F.\end{aligned}$$

[‡]The analysis follows Yang and Lai (2026) for Stiefel manifold. We also point out that the theoretical analysis in Gu and Xie (2026) explicitly assumes that the angles between the weights and the stochastic gradients are bounded away from zero, which is a very restrictive and non-verifiable assumption.

[§]Note that we do not require G to be on the tangent space, as proved in Yang and Lai (2026, Lemma 4.4)

Hence, we just need to bound $\|\nabla\mathcal{L}(\mathbf{W}_t) - M_t\|_F$. Denote $C_t := \beta C_{t-1} + (1-\beta)\nabla\mathcal{L}(\mathbf{W}_t)$ with $C_0 := \mathcal{L}(\mathbf{W}_0)$, we have:

$$\|\nabla\mathcal{L}(\mathbf{W}_t) - M_t\|_F \le \|\nabla\mathcal{L}(\mathbf{W}_t) - C_t\|_F + \|C_t - M_t\|_F.$$

For the first term, we have

$$\begin{aligned}
&\|\nabla\mathcal{L}(\mathbf{W}_t) - C_t\|_F \\
=&\|\nabla\mathcal{L}(\mathbf{W}_t) - (\beta C_{t-1} + (1-\beta)\nabla\mathcal{L}(\mathbf{W}_t))\|_F \\
=&\beta\|\nabla\mathcal{L}(\mathbf{W}_t) - C_{t-1}\|_F \\
\le&\beta\|\nabla\mathcal{L}(\mathbf{W}_t) - \nabla\mathcal{L}(\mathbf{W}_{t-1})\|_F + \beta\|\nabla\mathcal{L}(\mathbf{W}_{t-1}) - C_{t-1}\|_F \\
\le&\beta L\|\mathbf{W}_t - \mathbf{W}_{t-1}\|_F + \beta\|\nabla\mathcal{L}(\mathbf{W}_{t-1}) - C_{t-1}\|_F \\
\le&\beta L\eta_t\tilde{c} + \beta\|\nabla\mathcal{L}(\mathbf{W}_{t-1}) - C_{t-1}\|_F.
\end{aligned}$$

Applying recursively we get

$$\|\nabla\mathcal{L}(\mathbf{W}_t) - C_t\|_F \le L\eta_t\tilde{c}\sum_{i=0}^{t-1}\beta^i \le \frac{2L\eta_t\tilde{c}}{1-\beta}.$$

For the second term $\|C_t - M_t\|_F$, we have

$$\begin{aligned}
&\mathbb{E}\|C_t - M_t\|_F \\
\le&(1-\beta)\mathbb{E}\left\|\sum_{i=1}^{t}\beta^{t-i}(G_i - \nabla\mathcal{L}(\mathbf{W}_i))\right\|_F + \beta^t\mathbb{E}\|G_0 - \nabla\mathcal{L}(\mathbf{W}_0)\|_F \\
\le&(1-\beta)\sqrt{\mathbb{E}\left\|\sum_{i=1}^{t}\beta^{t-i}(G_i - \nabla\mathcal{L}(\mathbf{W}_i))\right\|_F^2} + \beta^t\sqrt{\mathbb{E}\|G_0 - \nabla\mathcal{L}(\mathbf{W}_0)\|_F^2} \\
\le&(1-\beta)\sqrt{\sum_{i=1}^{t}\beta^{2(t-i)}\sigma^2} + \beta^t\sigma \le \left(\sqrt{\frac{1-\beta}{1+\beta}} + \beta^t\right)\sigma.
\end{aligned}$$

Now plug in everything back to equation 17, we get

$$\begin{aligned}
&\tilde{c}\eta_t\mathbb{E}\|\nabla_{\mathcal{M}}\mathcal{L}(\mathbf{W}_t)\|_* \\
\le&\mathbb{E}(\mathcal{L}(\mathbf{W}_t) - \mathcal{L}(\mathbf{W}_{t+1})) + \tilde{c}\eta_t\sqrt{d}\mathbb{E}\|\nabla_{\mathcal{M}}\mathcal{L}(\mathbf{W}_t) - \mathrm{proj}_{\mathrm{T}_{\mathcal{M}}\mathbf{W}_t}(M_t)\|_F + \frac{\tilde{c}^2L\eta_t^2}{2}d \\
\le&\mathbb{E}(\mathcal{L}(\mathbf{W}_t) - \mathcal{L}(\mathbf{W}_{t+1})) + \tilde{c}(c_0+1)\eta_t\sqrt{d}\left(\frac{2L\eta_t\tilde{c}}{1-\beta} + \left(\sqrt{\frac{1-\beta}{1+\beta}} + \beta^t\right)\sigma\right) + \frac{\tilde{c}^2L\eta_t^2}{2}d.
\end{aligned}$$

Summing up the above inequality from $t = 0$ to $T-1$, we get

$$\sum_{t=0}^{T-1}\tilde{c}\eta_t\mathbb{E}\|\nabla_{\mathcal{M}}\mathcal{L}(\mathbf{W}_t)\|_* \lesssim \mathbb{E}[\mathcal{L}(\mathbf{W}_0) - \mathcal{L}^*] + \tilde{c}\sqrt{d}\sum_{t=0}^{T-1}\left(\frac{2\beta L\eta_t}{1-\beta} + (\sqrt{\frac{1-\beta}{1+\beta}} + \beta^t)\sigma\right)\eta_t + \frac{\tilde{c}^2dL}{2}\sum_{t=0}^{T-1}\eta_t^2.$$

The final result is obtained by plugging in

$$\beta = 1 - \frac{1}{\sqrt{T}},\ \eta_t = \eta = \Omega\left(\sqrt{\frac{\Delta}{LT^{3/2}}}\right)$$

to the above inequality. □

Discussion on the spectral sphere. The constraint $\mathcal{M}_S(R) = \{\mathbf{W} \in \mathbb{R}^{D_{\text{out}}\times D_{\text{in}}} : \|\mathbf{W}\|_2 = R > 0\}$ is actually not a manifold, due to the possible existence of multiple top singular values. This can be proved by the fact that the largest singular value is differentiable only at matrices where it is simple; see Lewis and Sendov (2005).

If we restrict the constraint to the subset where the top singular value is unique (and the first and second singular values have a small gap)

$$\tilde{\mathcal{M}}_S(R) = \{\mathbf{W} : \|\mathbf{W}\|_2 = R, \sigma_1(\mathbf{W}) \geq \sigma_2(\mathbf{W}) + \epsilon\}$$

then $\tilde{\mathcal{M}}_S(R)$ is shown to be a smooth embedded compact submanifold. A simple argument is that, on the closed set $\Omega = \{X : \sigma_1(X) \geq \sigma_2(X) + \epsilon\}$, the largest singular value $X \to \sigma_1(X)$ is Fréchet differentiable; again see Lewis and Sendov (2005). Also the spectral sphere constraint $\mathcal{M}_S(R) = \{\mathbf{W} \in \mathbb{R}^{D_{\text{out}} \times D_{\text{in}}} : \|\mathbf{W}\|_2 = R > 0\}$ is a regular level set, therefore their intersection is a smooth embedded submanifold by regular level set theorem; see Lee (2003). We can naturally consider this submanifold to apply Algorithm 1 and conduct a convergence analysis exactly the same as Theorem 1, due to the following reasons.

- The projection onto the tangent space of $\tilde{\mathcal{M}}_S(R)$ is exactly defined in Appendix B.1;
- It is hard globally compute the projectional retraction onto the set $\tilde{\mathcal{M}}_S(R)$ (The msign projection may not necessarily keep the top singular value unique). However we argue that this is not a problem in practice: due to stochasticity, the gradients G_t and Φ_t and the updates have unique top singular values with probability one, and msign could serve as the approximate projection in practice.

Therefore, despite the fact that $\mathcal{M}_S(R)$ is not a smooth manifold, the proposed Algorithm 1 and the convergence analysis in Theorem 1 can be applied for the spectral sphere constraint **as a manifold** almost surely.

C.2 Deriving the Radius Choices under Gaussian Initialization

In this section, we derive the radius bound when the weight direction is sampled independently from a normalized Gaussian distribution. At a layerwise-independent initialization, the incoming activation $\mathbf{X}$ depends only on the preceding layers and is independent of the current layer's Gaussian weight matrix, so the conditional formulation below applies directly.

For brevity, we still write

$$\begin{aligned} n &:= D_{\text{in}}, \qquad p := D_{\text{out}}, \\ r_{\mathbf{X}} &:= \frac{\|\mathbf{X}\|_F}{\sqrt{Tn}}, \qquad r_{\mathbf{Y}} := \frac{\|\mathbf{Y}\|_F}{\sqrt{Tp}}, \qquad \Sigma_{\mathbf{X}} := \frac{1}{T}\mathbf{X}^\top \mathbf{X}. \end{aligned} \tag{18}$$

All probabilities and expectations below are conditional on the realized input $\mathbf{X}$; equivalently, $\mathbf{X}$ may be any deterministic matrix.

Lemma 2 (Gaussian sketch and norm concentration)**.** *Let $\mathbf{G} \in \mathbb{R}^{p \times n}$ have i.i.d. $\mathcal{N}(0,1)$ entries and let $\mathbf{X} \in \mathbb{R}^{T \times n}$ be fixed. Define*

$$Q_{\mathbf{X}}(\mathbf{G}) := \|\mathbf{X}\mathbf{G}^\top\|_F^2.$$

For every $\varepsilon \in (0,1)$,

$$\mathbb{P}\big(Q_{\mathbf{X}}(\mathbf{G}) \leq (1-\varepsilon)p\|\mathbf{X}\|_F^2\big) \leq \exp\left(-\frac{p\varepsilon^2}{4}\right), \tag{19}$$

$$\mathbb{P}\big(Q_{\mathbf{X}}(\mathbf{G}) \geq (1+\varepsilon)p\|\mathbf{X}\|_F^2\big) \leq \exp\left(-\frac{p\varepsilon^2}{8}\right). \tag{20}$$

Moreover, for every $a > 0$ and $\delta \in (0,1)$,

$$\mathbb{P}\big(\|\mathbf{G}\|_2 > \sqrt{n} + (1+a)\sqrt{p}\big) \leq \exp\left(-\frac{a^2 p}{2}\right), \tag{21}$$

$$\mathbb{P}\big(\|\mathbf{G}\|_F^2 \notin [(1-\delta)pn, (1+\delta)pn]\big) \leq 2\exp\left(-\frac{pn\delta^2}{8}\right). \tag{22}$$

Proof. Let $\lambda_1, \ldots, \lambda_n \geq 0$ be the eigenvalues of $\mathbf{X}^\top\mathbf{X}$. By rotational invariance of a Gaussian matrix,

$$Q_{\mathbf{X}}(\mathbf{G}) \overset{d}{=} \sum_{j=1}^{n} \lambda_j \sum_{i=1}^{p} Z_{ij}^2, \qquad Z_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1), \tag{23}$$

with mean

$$\mu = p\sum_{j=1}^{n}\lambda_j = p\|\mathbf{X}\|_F^2.$$

If $\mathbf{X} = 0$, the first two claims are immediate, so suppose $\mu > 0$.

For the lower tail, for any $\theta > 0$,

$$\begin{aligned}
&\log \mathbb{E}\exp(-\theta(Q_{\mathbf{X}}(\mathbf{G}) - \mu)) \\
&\quad = \frac{p}{2}\sum_{j=1}^{n}(2\theta\lambda_j - \log(1+2\theta\lambda_j)) \\
&\quad \le p\theta^2\sum_{j=1}^{n}\lambda_j^2,
\end{aligned}$$

where we used

$$u - \log(1+u) \le \frac{u^2}{2}, \qquad u \ge 0.$$

Exponential Markov inequality and the choice

$$\theta = \frac{\varepsilon\sum_j \lambda_j}{2\sum_j \lambda_j^2}$$

give

$$\begin{aligned}
&\mathbb{P}(Q_{\mathbf{X}}(\mathbf{G}) \le (1-\varepsilon)\mu) \\
&\quad \le \exp\left(-\frac{p\varepsilon^2(\sum_j \lambda_j)^2}{4\sum_j \lambda_j^2}\right) \\
&\quad \le \exp\left(-\frac{p\varepsilon^2}{4}\right),
\end{aligned}$$

which proves equation 19.

For the upper tail, set

$$\theta = \frac{\varepsilon}{4\sum_j \lambda_j}.$$

Then $0 \le 2\theta\lambda_j \le \varepsilon/2 \le 1/2$, and

$$-\log(1-u) - u \le u^2, \qquad u \in [0, 1/2].$$

Hence

$$\begin{aligned}
&\log \mathbb{E}\exp(\theta(Q_{\mathbf{X}}(\mathbf{G}) - \mu)) \\
&\quad \le 2p\theta^2\sum_{j=1}^{n}\lambda_j^2 \\
&\quad \le 2p\theta^2\left(\sum_{j=1}^{n}\lambda_j\right)^2 = \frac{p\varepsilon^2}{8}.
\end{aligned}$$

Therefore, exponential Markov inequality gives

$$\begin{aligned}
&\mathbb{P}(Q_{\mathbf{X}}(\mathbf{G}) \ge (1+\varepsilon)\mu) \\
&\quad \le \exp\left(-\theta\varepsilon\mu + \frac{p\varepsilon^2}{8}\right) = \exp\left(-\frac{p\varepsilon^2}{8}\right),
\end{aligned}$$

which proves equation 20.

For equation 21, the classical Gaussian comparison bound gives

$$\mathbb{E}\|\mathbf{G}\|_2 \le \sqrt{n} + \sqrt{p},$$

and the map $\mathbf{G} \mapsto \|\mathbf{G}\|_2$ is one-Lipschitz with respect to the Frobenius norm. Gaussian concentration with deviation $a\sqrt{p}$ proves the claim.

Finally, $\|\mathbf{G}\|_F^2 \sim \chi^2_{pn}$. The standard Chernoff bounds for a chi-square variable give

$$\mathbb{P}\big(\|\mathbf{G}\|_F^2 \le (1-\delta)pn\big) \le e^{-pn\delta^2/4},$$

and

$$\mathbb{P}\big(\|\mathbf{G}\|_F^2 \ge (1+\delta)pn\big) \le e^{-pn\delta^2/8}.$$

Their union implies equation 22. □

Theorem 2 (Radius choices under normalized Gaussian initialization)**.** *Let $T, n, p \ge 1$. Fix any $\mathbf{X} \in \mathbb{R}^{T\times n}$, and let $\mathbf{G} \in \mathbb{R}^{p\times n}$ be independent of $\mathbf{X}$ with i.i.d. $\mathcal{N}(0,1)$ entries.*

(i) Spectral normalization. *Let*

$$\mathbf{W}_{\mathrm{spec}} := R_{\mathrm{spec}} \frac{\mathbf{G}}{\|\mathbf{G}\|_2}, \qquad R_{\mathrm{spec}} := r_{\mathrm{spec}} \sqrt{\frac{p}{n}}, \tag{24}$$

where $r_{\mathrm{spec}} > 0$, and set

$$r_{\mathbf{Y},\mathrm{spec}} := \frac{\|\mathbf{X}\mathbf{W}_{\mathrm{spec}}^\top\|_F}{\sqrt{Tp}}.$$

Then

$$r_{\mathbf{Y},\mathrm{spec}} \le r_{\mathrm{spec}} r_{\mathbf{X}} \qquad \textit{deterministically}. \tag{25}$$

For every $\varepsilon \in (0,1)$ and $a > 0$,

$$\mathbb{P}\left(\frac{r_{\mathrm{spec}}\sqrt{1-\varepsilon}}{\sqrt{n/p}+1+a}\, r_{\mathbf{X}} \le r_{\mathbf{Y},\mathrm{spec}} \le r_{\mathrm{spec}} r_{\mathbf{X}} \;\middle|\; \mathbf{X} \right) \ge 1 - e^{-p\varepsilon^2/4} - e^{-a^2 p/2}. \tag{26}$$

In particular, if $n/p \le C$ for a dimension-independent constant C, then $\sqrt{n/p}$ in equation 26 may be replaced by $\sqrt{C}$.

(ii) Frobenius normalization. *Let*

$$\mathbf{W}_{\mathrm{fro}} := R_{\mathrm{fro}} \frac{\mathbf{G}}{\|\mathbf{G}\|_F}, \qquad R_{\mathrm{fro}} := r_{\mathrm{fro}} \sqrt{p}, \tag{27}$$

where $r_{\mathrm{fro}} > 0$, and set

$$r_{\mathbf{Y},\mathrm{fro}} := \frac{\|\mathbf{X}\mathbf{W}_{\mathrm{fro}}^\top\|_F}{\sqrt{Tp}}.$$

For every finite n, p,

$$\mathbb{E}\big[r^2_{\mathbf{Y},\mathrm{fro}} \mid \mathbf{X}\big] = r^2_{\mathrm{fro}} r^2_{\mathbf{X}}. \tag{28}$$

More strongly, for every $\varepsilon, \delta \in (0,1)$,

$$\mathbb{P}\left(r_{\mathrm{fro}}\sqrt{\frac{1-\varepsilon}{1+\delta}}\, r_{\mathbf{X}} \le r_{\mathbf{Y},\mathrm{fro}} \le r_{\mathrm{fro}}\sqrt{\frac{1+\varepsilon}{1-\delta}}\, r_{\mathbf{X}} \;\middle|\; \mathbf{X} \right) \ge 1 - 2e^{-p\varepsilon^2/8} - 2e^{-pn\delta^2/8}. \tag{29}$$

Proof. **Spectral normalization.** For any deterministic pair $(\mathbf{X}, \mathbf{W})$,

$$\begin{aligned} \frac{\|\mathbf{X}\mathbf{W}^\top\|_F}{\sqrt{Tp}} &\le \frac{\|\mathbf{X}\|_F \|\mathbf{W}\|_2}{\sqrt{Tp}} \\ &= \sqrt{\frac{n}{p}}\, \|\mathbf{W}\|_2 r_{\mathbf{X}}. \end{aligned}$$

Substituting $\|\mathbf{W}_{\mathrm{spec}}\|_2 = R_{\mathrm{spec}}$ proves equation 25.

On the intersection of the events in equation 19 and equation 21,

$$\begin{aligned} r_{\mathbf{Y},\mathrm{spec}} &= \frac{R_{\mathrm{spec}}}{\|\mathbf{G}\|_2}\frac{\|\mathbf{X}\mathbf{G}^\top\|_F}{\sqrt{Tp}} \\ &\geq \frac{r_{\mathrm{spec}}\sqrt{p/n}}{\sqrt{n}+(1+a)\sqrt{p}}\,\frac{\sqrt{(1-\varepsilon)p}\,\|\mathbf{X}\|_F}{\sqrt{Tp}} \\ &= \frac{r_{\mathrm{spec}}\sqrt{1-\varepsilon}}{\sqrt{n/p}+1+a}\,r_{\mathbf{X}}. \end{aligned}$$

The union bound gives equation 26.

Frobenius normalization. Let

$$\mathbf{U} := \frac{\mathbf{G}}{\|\mathbf{G}\|_F}.$$

Vectorizing $\mathbf{U}$ gives a random vector uniformly distributed on the unit sphere in $\mathbb{R}^{pn}$. Rotational symmetry therefore implies

$$\mathbb{E}[U_{ij}U_{k\ell}] = \frac{1}{pn}\mathbf{1}\{(i,j)=(k,\ell)\},$$

and hence

$$\mathbb{E}\big[\mathbf{W}_{\mathrm{fro}}^\top\mathbf{W}_{\mathrm{fro}}\big] = \frac{R_{\mathrm{fro}}^2}{n}\mathbf{I}_n.$$

Consequently,

$$\begin{aligned} \mathbb{E}\big[r_{\mathbf{Y},\mathrm{fro}}^2 \mid \mathbf{X}\big] &= \frac{1}{Tp}\,\mathrm{Tr}\big(\mathbf{X}\,\mathbb{E}\big[\mathbf{W}_{\mathrm{fro}}^\top\mathbf{W}_{\mathrm{fro}}\big]\,\mathbf{X}^\top\big) \\ &= \frac{R_{\mathrm{fro}}^2}{Tpn}\|\mathbf{X}\|_F^2 \\ &= \frac{R_{\mathrm{fro}}^2}{p}r_{\mathbf{X}}^2 \\ &= r_{\mathrm{fro}}^2 r_{\mathbf{X}}^2, \end{aligned}$$

which proves equation 28.

For the high-probability statement, write

$$r_{\mathbf{Y},\mathrm{fro}}^2 = \frac{R_{\mathrm{fro}}^2}{Tp}\frac{Q_{\mathbf{X}}(\mathbf{G})}{\|\mathbf{G}\|_F^2}. \tag{30}$$

On the event where

$$(1-\varepsilon)p\|\mathbf{X}\|_F^2 \leq Q_{\mathbf{X}}(\mathbf{G}) \leq (1+\varepsilon)p\|\mathbf{X}\|_F^2$$

and

$$(1-\delta)pn \leq \|\mathbf{G}\|_F^2 \leq (1+\delta)pn,$$

identity equation 30, together with

$$R_{\mathrm{fro}}^2 = r_{\mathrm{fro}}^2 p, \qquad \|\mathbf{X}\|_F^2 = Tnr_{\mathbf{X}}^2,$$

gives

$$r_{\mathrm{fro}}^2\frac{1-\varepsilon}{1+\delta}r_{\mathbf{X}}^2 \leq r_{\mathbf{Y},\mathrm{fro}}^2 \leq r_{\mathrm{fro}}^2\frac{1+\varepsilon}{1-\delta}r_{\mathbf{X}}^2.$$

Taking square roots and applying Lemma 2 with a union bound proves equation 29. □

Remark 1 (Implication for radius selection)**.** *Theorem 2 directly implies the radius scalings used in the main text. In particular, under normalized Gaussian initialization, if $r_{\mathbf{X}} = \Theta(1)$ and $n/p = \mathcal{O}(1)$, then maintaining $r_{\mathbf{Y}} = \Theta(1)$ with high probability requires, up to dimension-independent constants,*

$$R_{\mathrm{spec}} = \Theta\left(\sqrt{\frac{p}{n}}\right), \qquad R_{\mathrm{fro}} = \Theta(\sqrt{p})\,.$$

Thus the Gaussian initialization analysis recovers exactly the spectral and Frobenius radius choices adopted in Proposition 1.

C.2.1 Practical Radius Selection

Combining Theorem 2 with the norm-alignment relations of Oblique Manifolds definition in Section 2 yields a single tunable hyperparameter $r = \Theta(1)$ that fixes all four radii simultaneously:

$$R_{\text{spec}} = r\sqrt{\tfrac{D_{\text{out}}}{D_{\text{in}}}}, \quad R_{\text{fro}} = r\sqrt{D_{\text{out}}}, \quad R_{\text{out}} = \tfrac{R_{\text{fro}}}{\sqrt{D_{\text{out}}}} = r, \quad R_{\text{in}} = \tfrac{R_{\text{fro}}}{\sqrt{D_{\text{in}}}} = r\sqrt{\tfrac{D_{\text{out}}}{D_{\text{in}}}}. \tag{31}$$

The mapping $R_{\text{out}} = \|\mathbf{W}\|_F/\sqrt{D_{\text{out}}}$ and $R_{\text{in}} = \|\mathbf{W}\|_F/\sqrt{D_{\text{in}}}$ follows directly from $\|\mathbf{W}\|_F^2 = D_{\text{out}}R_{\text{out}}^2 = D_{\text{in}}R_{\text{in}}^2$ on the corresponding manifolds; these are the choices summarized in Section 3.2.

Why the Radius Depends on the Norm Choice. The worst-case estimate $r_{\mathbf{Y}} \le \sqrt{n/p}\,\|\mathbf{W}\|_F\, r_{\mathbf{X}}$ suggests using the spectral radius scaling for the Frobenius constraint. However, this choice is too conservative at Gaussian initialization. For $\mathbf{W}_{\text{fro}} = R_{\text{fro}}\mathbf{G}/\|\mathbf{G}\|_F$, with $\mathbf{G}$ sampled independently of $\mathbf{X}$ as in Appendix C.2, the second-moment calculation gives

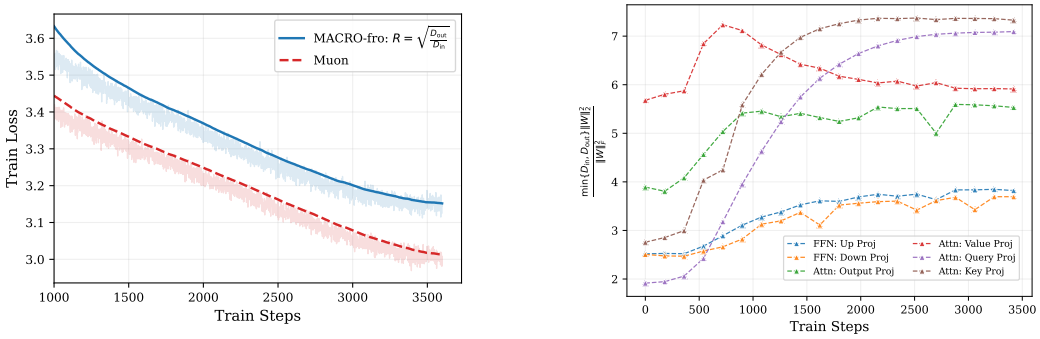
$$\mathbb{E}\big[r_{\mathbf{Y},\text{fro}}^2 \mid \mathbf{X}\big] = \frac{R_{\text{fro}}^2}{p}\, r_{\mathbf{X}}^2 = \begin{cases} \dfrac{r_{\text{spec}}^2}{n}\, r_{\mathbf{X}}^2, & R_{\text{fro}} = r_{\text{spec}}\sqrt{p/n}, \\ r_{\text{fro}}^2 r_{\mathbf{X}}^2, & R_{\text{fro}} = r_{\text{fro}}\sqrt{p}. \end{cases} \tag{32}$$

For order-one radius multipliers, directly using the spectral radius therefore introduces a width-dependent attenuation in the initial mean squared output RMS. In contrast, the proposed Frobenius radius keeps this quantity comparable to $r_{\mathbf{X}}^2$. Figure 6 (a) illustrates the practical issue: `MACRO`-fro with $R_{\text{fro}} = \sqrt{p/n}$ attains higher training loss than Muon in the reported experiment.

Consistency with prior initialization schemes. The radii in equation 31 agree with the initialization conventions of Yang et al. (2023) and Wen et al. (2026). If the model is initialized as $\mathbf{W}_0 \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = \min\Big\{\sqrt{D_{\text{out}}/D_{\text{in}}},\, 1\Big\}/\sqrt{D_{\text{in}}}$, then

$$\mathbb{E}[\|\mathbf{W}_0\|_F] \;\approx\; \sqrt{D_{\text{out}}D_{\text{in}}\,\text{Var}(\mathbf{W}_{ij})} \;=\; \min\Big\{\sqrt{\tfrac{D_{\text{out}}}{D_{\text{in}}}},\, 1\Big\}\sqrt{D_{\text{out}}}, \tag{33}$$

which matches our Frobenius radius $R_{\text{fro}} = r\sqrt{D_{\text{out}}}$ for $r = \min\{\sqrt{D_{\text{out}}/D_{\text{in}}}, 1\} = \Theta(1)$. Likewise, the data-dependent strategy of Wen et al. (2026) fixes the radius to the realized energy of an $\mathcal{N}(0, 1/D_{\text{out}})$ initialization, which corresponds to $r = 1$ in equation 31.



(a) Training Loss.

(b) Normalized inverse stable rank κ.

Figure 6: (a) Train loss for 120M QWEN3-like model under Frobenius Sphere with different radius. (b) Normalized inverse stable rank during training.

C.3 Rotational Equilibrium under Frobenius Sphere in Section 4.2

This subsection supplies a formal derivation of the Frobenius rotation identity stated in the main body (Section 4.2, paragraph on static rotational equilibrium under the Frobenius sphere) that does not rely on the heuristic Frobenius-norm-preservation argument used there. The argument is parametrized by the geometric *tangency cosine*

$$\alpha_t \;:=\; \frac{\langle \mathbf{W}_t,\, O_t\rangle_F}{R\,\|O_t\|_F} \;=\; \cos\angle(\mathbf{W}_t,\, O_t) \;\in\; [-1,\, 1], \tag{34}$$

which is independent of η_t because $O_t = \mathrm{msign}(\Phi_t)$ depends on the gradient geometry alone.

Setup. On the Frobenius sphere $\mathcal{M}_F(R) = \{\mathbf{W} : \|\mathbf{W}\|_F = R\}$, the projection of any non-zero matrix A is $\mathcal{P}_{\mathcal{M}_F}(A) = R\,A/\|A\|_F$. With the normalized update direction $\widetilde{\nabla}_t = (R/\|O_t\|_F)\,O_t$ used by MACRO,

$$\mathbf{W}_{t+1} \;=\; \mathcal{P}_{\mathcal{M}_F}\big(\mathbf{W}_t - \eta_t\widetilde{\nabla}_t\big) \;=\; \frac{R}{\|\mathbf{W}_t - \eta_t\widetilde{\nabla}_t\|_F}\big(\mathbf{W}_t - \eta_t\widetilde{\nabla}_t\big), \tag{35}$$

and $\|\mathbf{W}_t\|_F = \|\mathbf{W}_{t+1}\|_F = R$ by construction.

Exact projected angle. Direct computation yields

$$\langle \mathbf{W}_t - \eta_t\widetilde{\nabla}_t,\, \mathbf{W}_t\rangle_F \;=\; R^2 - \eta_t R^2\,\alpha_t \;=\; R^2\big(1 - \eta_t\,\alpha_t\big), \tag{36}$$

$$\|\mathbf{W}_t - \eta_t\widetilde{\nabla}_t\|_F^2 \;=\; R^2 - 2\eta_t R^2\,\alpha_t + \eta_t^2 R^2 \;=\; R^2\big(1 - 2\eta_t\,\alpha_t + \eta_t^2\big). \tag{37}$$

Combining equation 35–equation 37,

$$\cos\theta_t \;=\; \frac{\langle \mathbf{W}_{t+1}, \mathbf{W}_t\rangle_F}{\|\mathbf{W}_{t+1}\|_F\,\|\mathbf{W}_t\|_F} \;=\; \frac{1 - \eta_t\,\alpha_t}{\sqrt{1 - 2\eta_t\,\alpha_t + \eta_t^2}}. \tag{38}$$

Equation equation 38 is exact in η_t and α_t. No norm-preservation heuristic has been used; in particular, the inner product $\langle \mathbf{W}_t, O_t\rangle_F$ enters only through the η_t-independent quantity α_t.

Taylor expansion. Expanding the denominator of equation 38 via the binomial series $(1+x)^{-1/2} = 1 - \frac{x}{2} + \frac{3x^2}{8} + O(x^3)$ with $x = -2\eta_t\,\alpha_t + \eta_t^2$ and retaining terms through $O(\eta_t^2)$,

$$(1 - 2\eta_t\,\alpha_t + \eta_t^2)^{-1/2} \;=\; 1 + \eta_t\,\alpha_t - \tfrac{1}{2}\eta_t^2 + \tfrac{3}{2}\eta_t^2\,\alpha_t^2 + O(\eta_t^3).$$

Multiplying by the numerator $1 - \eta_t\,\alpha_t$, the $O(\eta_t)$ terms cancel and we obtain

$$\cos\theta_t \;=\; 1 \;-\; \tfrac{1}{2}\,\eta_t^2\,\big(1 - \alpha_t^2\big) + O(\eta_t^3). \tag{39}$$

Inverting via $\arccos(1 - z) \approx \sqrt{2z}$ for small z yields the rigorous Frobenius rotation identity

$$\boxed{\theta_t \;\approx\; \eta_t\,\sqrt{1 - \alpha_t^2} \;=\; \eta_t\,\sin\angle(\mathbf{W}_t,\, O_t).} \tag{40}$$

Across our experiments the tangent-space residual for $\mathcal{M}_F$ $\mathrm{vio}(\mathbf{W}_t) := |\langle \mathbf{W}_t, \widetilde{\nabla}_t\rangle|$ stabilizes at 10^{-2}–10^{-3} for every linear-layer type throughout training; see Figure 12 (Frobenius panel, right). Since $|\alpha_t| = \mathrm{vio}(W_t)/R^2$, the same regime gives $|\alpha_t| \approx 0$, so

$$\sqrt{1 - \alpha_t^2} \;=\; 1 - \tfrac{1}{2}\alpha_t^2 + O(\alpha_t^4) \;\approx\; 1,$$

and equation 40 reduces to

$$\theta_t \;\approx\; \eta_t, \tag{41}$$

which is the identity used in the main body and confirmed empirically by the flat curves of Figure 2.

Contrast with the spectral case. The same setup under the spectral norm requires Wedin's $\sin\Theta$ theorem and produces an *adaptive* rotation angle modulated by the training-dependent spectral gap; see Appendix C.4. The contrast is geometric: the Frobenius-side rigorous identity equation 40 depends only on η_t and a small empirically-measured residual α_t, whereas the spectral-side bound additionally depends on the iterate-dependent quantity $R - \sigma_2(\mathbf{W}_{t+1})$, producing the adaptive, anisotropic dynamics visualized in Figure 3.

C.4 Rotational Equilibrium under Spectral Sphere in Section 4.2

This section supplies the formal proof of the spectral rotation bound stated in the main body (Section 4.2, paragraph "Adaptive Rotational Equilibrium under Spectral Sphere"). Concretely, we show that under the normalized update rule of MACRO, the leading left and right singular subspaces of $\mathbf{W}_t$ rotate by an angle whose magnitude is governed by the local spectral gap of $\mathbf{W}_t$. The argument relies on (i) the standard equivalence between subspace distance and principal angles (Appendix C.4.2), and (ii) the Wedin $\sin\Theta$ theorem for singular subspaces (Appendix C.4.3). The two ingredients are then combined in Appendix C.4.4.

C.4.1 SETUP: NORMALIZED UPDATE ON THE SPECTRAL SPHERE

We start from the update step without projection step such as

$$\widetilde{\mathbf{W}}_{t+1} \;=\; \mathbf{W}_t \;-\; \eta_t \,\frac{R\,O_t}{\|O_t\|_{\mathcal{M}}}, \text{ and } \mathbf{W}_{t+1} = \mathcal{P}_{\mathcal{M}_S}(\widetilde{\mathbf{W}}_{t+1}) \tag{42}$$

where O_t is the (un-projected) update direction returned by the optimizer, R is the constraint radius, and $\|\cdot\|_{\mathcal{M}}$ denotes the manifold-specific norm. For the spectral sphere $\mathcal{M} = \mathcal{M}_{\mathrm{spec}}(R)$ we set $\|\cdot\|_{\mathcal{M}} = \|\cdot\|_2$, and we denote the spectral perturbation by

$$\mathbf{E}_t \;:=\; \widetilde{\mathbf{W}}_{t+1} - \mathbf{W}_t \;=\; -\,\eta_t\,\frac{R\,O_t}{\|O_t\|_2}, \qquad \|\mathbf{E}_t\|_2 \;=\; \eta_t\,R. \tag{43}$$

Below, Wedin's theorem is applied to the pair $(\mathbf{W}_t, \widetilde{\mathbf{W}}_{t+1})$, and then we transfer the resulting principal-angle bound from $(\mathbf{W}_t, \widetilde{\mathbf{W}}_{t+1})$ to $(\mathbf{W}_t, \mathbf{W}_{t+1})$.

Remark 2 (Update activation control)**.** *The same normalized update equation 42, combined with the input scale assumption* $\|\operatorname{vec}(\mathbf{X})\|_{\mathrm{RMS}} = \Theta(1)$*, also secures the second* μ*P desideratum (Figure 8) on the activation update,*

$$\|\operatorname{vec}(\Delta\mathbf{Y}_t)\|_{\mathrm{RMS}} \;=\; \eta_t \left\|\operatorname{vec}\Big(\tfrac{R\,O_t}{\|O_t\|_2}\,\mathbf{X}\Big)\right\|_{\mathrm{RMS}} \;=\; \Theta(1), \tag{44}$$

since $R\eta_t = \Theta(1)$ *and the bracketed factor obeys exactly the bound established for* $\mathbf{Y}_t$ *in Lemma* **??**.

C.4.2 SUBSPACE DISTANCE AND PRINCIPAL ANGLES

The Frobenius rotation angle of the main body is well defined because $\|\cdot\|_F$ is induced by an inner product. The spectral norm is not, and the natural object of study is instead the rotation of *singular subspaces*, measured through principal angles.

Definition 1 (Subspace distance and principal angles)**.** *Let* $X, Z \in \mathbb{R}^{m\times r}$ *have orthonormal columns* ($X^\top X = Z^\top Z = \mathbf{I}_r$)*. The* subspace distance *between* $\mathrm{range}(X)$ *and* $\mathrm{range}(Z)$ *is*

$$\mathrm{dist}(X, Z) \;:=\; \big\|XX^\top - ZZ^\top\big\|_2. \tag{45}$$

Let $X^\top Z = U\,\Sigma\,V^\top$ *be the singular value decomposition of* $X^\top Z$*, and write* $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$ *with* $1 \ge \sigma_1 \ge \cdots \ge \sigma_r \ge 0$*. The* principal angles $\theta_1, \ldots, \theta_r \in [0, \pi/2]$ *between* $\mathrm{range}(X)$ *and* $\mathrm{range}(Z)$ *are defined by* $\cos\theta_i = \sigma_i$*.*

Lemma 3 (Subspace distance and the largest principal angle)**.** *For* $X, Z \in \mathbb{R}^{m\times r}$ *with orthonormal columns,*

$$\mathrm{dist}(X, Z) \;=\; \max_{1\le i\le r} \sin\theta_i \;=\; \sin\theta_{\max}. \tag{46}$$

Proof. This is Stewart and Sun (1990, Theorem I.5.5). We give the short argument for completeness. Complete X and Z to orthogonal matrices $[X\ X_\perp]$ and $[Z\ Z_\perp]$ in $\mathbb{R}^{m\times m}$, so that

$$XX^\top - ZZ^\top \;=\; XX^\top(I - ZZ^\top) - (I - XX^\top)ZZ^\top \;=\; X(X^\top Z_\perp)Z_\perp^\top \;-\; X_\perp(X_\perp^\top Z)Z^\top. \tag{47}$$

The two terms have orthogonal column spaces and orthogonal row spaces, so

$$\|XX^\top - ZZ^\top\|_2 \;=\; \max\big\{\|X^\top Z_\perp\|_2,\ \|X_\perp^\top Z\|_2\big\}. \tag{48}$$

Each of these two operator norms equals $\sin\theta_{\max}$: indeed, the singular values of $X^\top Z$ are $\cos\theta_1, \ldots, \cos\theta_r$, hence those of $X^\top Z_\perp$ are $\sin\theta_1, \ldots, \sin\theta_r$ (e.g., Stewart and Sun, 1990, Thm. I.5.2), and likewise for $X_\perp^\top Z$. The maximum singular value is therefore $\sin\theta_{\max}$, which proves equation 46. □

C.4.3 WEDIN'S $\sin\Theta$ THEOREM

We use the singular-vector form of the Davis–Kahan / Wedin perturbation theorem.

Theorem 3 (Wedin $\sin\Theta$ theorem; Wedin, 1972, see also Davis and Kahan, 1970; Stewart and Sun, 1990)**.** *Let $A, \tilde{A} \in \mathbb{R}^{m\times n}$ with $\tilde{A} = A + E$, and let*

$$A = [U_0\ U_1]\begin{bmatrix}\Sigma_0 & \\ & \Sigma_1\end{bmatrix}\begin{bmatrix}V_0^\top \\ V_1^\top\end{bmatrix}, \qquad \tilde{A} = [\tilde{U}_0\ \tilde{U}_1]\begin{bmatrix}\tilde{\Sigma}_0 & \\ & \tilde{\Sigma}_1\end{bmatrix}\begin{bmatrix}\tilde{V}_0^\top \\ \tilde{V}_1^\top\end{bmatrix} \tag{49}$$

be partitioned SVDs in which $\Sigma_0, \tilde{\Sigma}_0$ collect the top r singular values and $U_0, V_0, \tilde{U}_0, \tilde{V}_0$ have r orthonormal columns. Define the spectral gap

$$\Delta \ := \ \min_{i\le r,\, j>r} \big|\sigma_i(A) - \sigma_j(\tilde{A})\big|. \tag{50}$$

If $\Delta > 0$, then

$$\max\big\{\mathrm{dist}(\tilde{U}_0, U_0),\ \mathrm{dist}(\tilde{V}_0, V_0)\big\} \ \le \ \frac{\max\big\{\|EV_0\|_2,\ \|E^\top U_0\|_2\big\}}{\Delta} \ \le \ \frac{\|E\|_2}{\Delta}. \tag{51}$$

We will apply Theorem 3 to $A = \mathbf{W}_t$, $\tilde{A} = \mathbf{W}_{t+1}$, and the perturbation $E = \mathbf{E}_t$ from equation 43.

C.4.4 Proof of the Spectral Rotational Equilibrium Bound

We now combine Lemma 3 and Theorem 3 to bound the principal rotation angles between consecutive iterates on the spectral sphere. Let

$$\mathbf{W}_t \ = \ [\mathbf{U}_t^0\ \mathbf{U}_t^1]\begin{bmatrix}\Sigma_t^0 & \\ & \Sigma_t^1\end{bmatrix}\begin{bmatrix}\mathbf{V}_t^{0\top} \\ \mathbf{V}_t^{1\top}\end{bmatrix}, \qquad \mathbf{W}_{t+1} \ = \ [\mathbf{U}_{t+1}^0\ \mathbf{U}_{t+1}^1]\begin{bmatrix}\Sigma_{t+1}^0 & \\ & \Sigma_{t+1}^1\end{bmatrix}\begin{bmatrix}\mathbf{V}_{t+1}^{0\top} \\ \mathbf{V}_{t+1}^{1\top}\end{bmatrix} \tag{52}$$

be partitioned SVDs in which $\mathbf{U}_\cdot^0, \mathbf{V}_\cdot^0 \in \mathbb{R}^{D_{\text{out}}\times r}, \mathbb{R}^{D_{\text{in}}\times r}$ span the top-r left and right singular subspaces, respectively. Set $\mathbf{E}_t$ as in equation 43, so that $\|\mathbf{E}_t\|_2 = \eta_t R$.

General-rank bound. Theorem 3 applied to $A = \mathbf{W}_t$, $\tilde{A} = \mathbf{W}_{t+1}$, $E = \mathbf{E}_t$, with the spectral gap

$$\Delta_r \ = \ \sigma_r(\mathbf{W}_t) - \sigma_{r+1}(\mathbf{W}_{t+1}) \tag{53}$$

(assumed strictly positive), yields

$$\max\big\{\mathrm{dist}(\mathbf{U}_{t+1}^0, \mathbf{U}_t^0),\ \mathrm{dist}(\mathbf{V}_{t+1}^0, \mathbf{V}_t^0)\big\} \ \le \ \frac{\|\mathbf{E}_t\|_2}{\Delta_r} \ = \ \frac{\eta_t\, R}{\Delta_r}. \tag{54}$$

By Lemma 3, the left-hand side equals the maximum sine of the principal angles, so

$$\max\big\{\sin\theta^u_{\max},\ \sin\theta^v_{\max}\big\} \ \le \ \frac{\eta_t\, R}{\Delta_r}, \tag{55}$$

where $\theta^u_{\max}$ (resp. $\theta^v_{\max}$) is the largest principal angle between the top-r left (resp. right) singular subspaces of $\mathbf{W}_t$ and $\mathbf{W}_{t+1}$.

Specialization to the leading direction. The constraint $\|\mathbf{W}_t\|_2 = R$ is active precisely on the leading singular value, so the strongest geometric restriction is on the top singular pair $(\mathbf{u}_t, \mathbf{v}_t)$. Setting $r = 1$ and using $\sigma_1(\mathbf{W}_t) = \sigma_1(\mathbf{W}_{t+1}) = R$, the spectral gap equation 53 becomes

$$\Delta_1 \ = \ R - \sigma_2(\mathbf{W}_{t+1}), \tag{56}$$

and equation 55 reduces to

$$\max\big\{\sin\theta_u,\ \sin\theta_v\big\} \ \le \ \frac{\eta_t\, R}{R - \sigma_2(\mathbf{W}_{t+1})}, \tag{57}$$

where $\theta_u = \arccos|\mathbf{u}_{t+1}^\top \mathbf{u}_t|$ and $\theta_v = \arccos|\mathbf{v}_{t+1}^\top \mathbf{v}_t|$ are the angles between the leading singular vectors of consecutive iterates.

Spectral effective learning rate. Defining the spectral effective learning rate along the primary direction as $\eta_{\mathrm{eff},\,t}^{\mathrm{spec}} := \max\{\theta_u, \theta_v\}$, the small-angle approximation $\sin\theta \approx \theta$ (valid in typical training where $\eta_t R \ll \Delta_1$) converts equation 57 into

$$\boxed{\eta_{\mathrm{eff},\,t}^{\mathrm{spec}} \lesssim \eta_t \left(\frac{R}{R - \sigma_2(\mathbf{W}_{t+1})}\right).} \tag{58}$$

The bound equation 58 is derived by applying Wedin's theorem to the pair $(\mathbf{W}_t, \widetilde{\mathbf{W}}_{t+1})$ rather than to $(\mathbf{W}_t, \mathbf{W}_{t+1})$. However, the projected iterate $\mathbf{W}_{t+1} = \mathcal{P}_{\mathcal{M}_S(R)}(\widetilde{\mathbf{W}}_{t+1})$ shares the singular subspaces of $\widetilde{\mathbf{W}}_{t+1}$ and preserves σ_j for every $j \geq 2$. Both the principal angles θ_u, θ_v and the spectral gap $R - \sigma_2(\mathbf{W}_{t+1})$ are therefore identical when $\widetilde{\mathbf{W}}_{t+1}$ is replaced by $\mathbf{W}_{t+1}$. The bound equation 58 consequently applies to the actual iterate produced by Algorithm 1, with no additional assumption.

This completes the proof of the spectral rotation bound stated in the main body. The contrast with the Frobenius case $\eta_{\mathrm{eff},\,t}^{\mathrm{frob}} \approx \eta_t$ is geometric: under the Frobenius constraint the rotation angle is a function of the learning rate alone, whereas under the spectral constraint it is modulated by the training-dependent spectral gap $R - \sigma_2(\mathbf{W}_{t+1})$, producing the adaptive, anisotropic dynamics empirically visualized in Figure 3.

C.5 Update Cancellation model in Section 4.3

This appendix explains the Gaussian model we use to analyze the update cancellation under bf16 master weight training, propose the Proposition 2 and its proof, and gives the learning-rate calculations under lr cosine decay schedule. Numerical comparisons are reported in Appendix D.3. As in Section 4.3, $\Delta\mathbf{W}_t$ denotes an additive optimizer increment before retraction and rounding; for Muon, it excludes weight decay.

C.5.1 Unit of Least Precision and increments cancellation under BF16 precision

Let w be a nonzero BF16 value, and let $w^- < w < w^+$ be its adjacent representable values. We consider finite normal values away from underflow and overflow. Under rounding to nearest, an increment is canceled when $w + \delta$ lies in the rounding interval of w (Zamirai et al., 2020). Ignoring midpoint ties, this condition is

$$Q_{\mathrm{BF16}}(w + \delta) = w \quad \Longleftrightarrow \quad -\frac{w - w^-}{2} < \delta < \frac{w^+ - w}{2}. \tag{59}$$

Midpoint ties are resolved by the ties-to-even rule.

BF16 has seven explicit fraction bits. Away from exponent boundaries, write $|w| = m2^e$ with $1 < m < 2$. Both neighboring spacings equal $h(w) = 2^{e-7}$, so the local relative cancellation threshold is

$$\tau_{\mathrm{loc}}(w) = \frac{h(w)}{2|w|} = \frac{2^{-8}}{m}, \qquad 2^{-9} < \tau_{\mathrm{loc}}(w) < 2^{-8}. \tag{60}$$

At $|w| = 2^e$, the spacing toward zero is 2^{e-8}, while the spacing away from zero is 2^{e-7}. The corresponding relative half-spacings are 2^{-9} and 2^{-8}. These formulas exclude the smallest-normal boundary, where the subnormal spacing must be considered separately.

For example, at $w = 1.5$, the spacing is 0.0078125, and the absolute half-spacing is 0.00390625. An increment $\delta = 0.002$ is canceled, whereas $\delta = 0.004$ produces a realized increment of 0.0078125. Thus, a surviving increment need not retain its proposed magnitude.

We approximate the varying cancellation intervals by the common threshold $|\delta| \leq \tau|w|$. For all BF16 numerical estimates in this paper, we use

$$\tau \approx 1.96 \times 10^{-3}. \tag{61}$$

This is a lower-end reference from the BF16 format, not a fitted loss-onset threshold or the exact threshold of every coordinate. The threshold model cancels small increments and leaves surviving proposed increments unchanged; it does not reproduce their full BF16 rounding.

C.5.2 FORMAL UPDATE CANCELLATION MODEL AND THE PROPOSITION

In this section, we formally present the model we use to explain the loss degradation during the late stage of training in Section 4.3. Fix a training step and flatten $\mathbf{W}_t$ and $\Delta\mathbf{W}_t$ into $N = pn$ coordinate pairs (w_i, δ_i). We define

$$D_i := \mathbf{1}\{|\delta_i| > \tau|w_i|\}.$$

Then the canceled-coordinate fraction and the retained fraction of proposed absolute-update magnitude are defined as:

$$\widehat{p}_{\text{drop}} = \frac{1}{N}\sum_{i=1}^{N}(1 - D_i), \qquad \widehat{\mathcal{R}} = \frac{\sum_{i=1}^{N}|\delta_i|D_i}{\sum_{i=1}^{N}|\delta_i|}. \tag{62}$$

The two statistics use different weights: each coordinate contributes equally to $\widehat{p}_{\text{drop}}$, whereas its contribution to $\widehat{\mathcal{R}}$ is proportional to $|\delta_i|$.

Let I be uniform on $\{1, \ldots, N\}$. Then

$$\begin{aligned} \widehat{p}_{\text{drop}} &= \Pr_I(|\delta_I| \le \tau|w_I|)\,, \\ \widehat{\mathcal{R}} &= \frac{\mathbb{E}_I[|\delta_I|D_I]}{\mathbb{E}_I|\delta_I|}. \end{aligned} \tag{63}$$

These are exact identities for the fixed arrays. The sampling only expresses proportions across coordinates as probabilities and expectations; it does not change the rounding procedure.

To simulate the update cancellation, we follow the observations that weight matrices can be modeled as zero-mean Gaussian random matrices with i.i.d. entries (Huang et al., 2024; Kedia et al., 2024; Babaoglu et al., 2026). To match the random Gaussian Model with the scale of the weights and updates we set second moments of the Gaussian Model by setting

$$\sigma_w^2 = \frac{1}{N}\sum_{i=1}^{N} w_i^2 = \|W_t\|_{\text{RMS}}^2, \qquad \sigma_s^2 = \frac{1}{N}\sum_{i=1}^{N}\delta_i^2 = \|\Delta W_t\|_{\text{RMS}}^2. \tag{64}$$

Then we have following proposition:

Proposition 2 (Update retention under a Gaussian threshold model)**.** *Let $w \sim \mathcal{N}(0, \|\mathbf{W}_t\|_{RMS})$ and $\delta \sim \mathcal{N}(0, \|\Delta\mathbf{W}_t\|_{RMS}^2)$ be independent. For a cancellation threshold $\tau > 0$, set $\rho = \sigma_s/\sigma_w$ and $\delta_\tau = \delta\,\mathbf{1}\{|\delta| > \tau|w|\}$. The canceled-coordinate probability and the retained fraction of absolute-update magnitude are*

$$p_{\text{drop}}(\rho) := \Pr(|\delta| \le \tau|w|) = \frac{2}{\pi}\arctan\frac{\tau}{\rho}, \quad \mathcal{R}(\rho) := \frac{\mathbb{E}|\delta_\tau|}{\mathbb{E}|\delta|} = \frac{1}{\sqrt{1 + (\tau/\rho)^2}}. \tag{65}$$

Proof of Proposition 2. **Discarded-coordinate probability.** Write $w = \sigma_w Z_w$ and $\delta = \sigma_s Z_s$, where Z_w and Z_s are independent standard Gaussian variables. The cancellation event becomes

$$|\delta| \le \tau|w| \quad \Longleftrightarrow \quad \left|\frac{Z_s}{Z_w}\right| \le \frac{\tau}{\rho}.$$

The ratio Z_s/Z_w has density

$$f(z) = \int_{-\infty}^{\infty}\frac{|y|}{2\pi}\exp\left(-\frac{(1+z^2)y^2}{2}\right)dy = \frac{1}{\pi(1+z^2)}.$$

Integrating this density gives

$$p_{\text{drop}}(\rho) = \int_{-\tau/\rho}^{\tau/\rho}\frac{dz}{\pi(1+z^2)} = \frac{2}{\pi}\arctan\frac{\tau}{\rho}.$$

Retained absolute-update magnitude. Condition on $w = y$ and put $c = \tau|y|$. Independence implies

$$\begin{aligned} \mathbb{E}[|\delta_\tau| \mid w = y] &= 2\int_c^{\infty} x\,\frac{\exp(-x^2/(2\sigma_s^2))}{\sigma_s\sqrt{2\pi}}\,dx \\ &= \sigma_s\sqrt{\frac{2}{\pi}}\,\exp\left(-\frac{\tau^2y^2}{2\sigma_s^2}\right). \end{aligned}$$

Since $\mathbb{E}|\delta| = \sigma_s\sqrt{2/\pi}$, averaging over w yields

$$\begin{aligned}\mathcal{R}(\rho) &= \mathbb{E}_w \exp\left(-\frac{\tau^2 w^2}{2\sigma_s^2}\right) \\ &= \frac{1}{\sigma_w\sqrt{2\pi}}\int_{-\infty}^{\infty}\exp\left[-\frac{1}{2}\left(\frac{1}{\sigma_w^2}+\frac{\tau^2}{\sigma_s^2}\right)y^2\right]\,dy \\ &= \left(1+\frac{\tau^2\sigma_w^2}{\sigma_s^2}\right)^{-1/2} = \frac{1}{\sqrt{1+(\tau/\rho)^2}}.\end{aligned}$$

□

Retained relative scale. The Gaussian means of the absolute values satisfy

$$\frac{\mathbb{E}|\delta|}{\mathbb{E}|w|} = \frac{\sigma_s}{\sigma_w} = \rho.$$

It follows that

$$\frac{\mathbb{E}|\delta_\tau|}{\mathbb{E}|w|} = \rho\,\mathcal{R}(\rho) = \frac{\rho^2}{\sqrt{\rho^2+\tau^2}}.$$

For the idealized Frobenius-normalized step, $\rho = \eta_t$, giving

$$\eta_{\mathrm{ret},t} := \frac{\mathbb{E}|\delta_\tau|}{\mathbb{E}|w|} = \eta_t\mathcal{R}(\eta_t) = \frac{\eta_t^2}{\sqrt{\eta_t^2+\tau^2}}. \tag{66}$$

This diagnostic uses mean absolute magnitudes, rather than RMS magnitudes of the thresholded update, and is not an effective learning rate for the realized BF16 trajectory.

For $\tau/\rho \to 0$, the binomial expansion gives

$$1-\mathcal{R}(\rho) = \frac{1}{2}\left(\frac{\tau}{\rho}\right)^2 + O\left(\left(\frac{\tau}{\rho}\right)^4\right).$$

For $\rho/\tau \to 0$,

$$\mathcal{R}(\rho) \sim \frac{\rho}{\tau}, \qquad \rho\,\mathcal{R}(\rho) \sim \frac{\rho^2}{\tau}.$$

These limits describe gradual changes in cancellation, not a binary transition of the entire matrix.

Relative-update scales for the three optimizers. For a nonzero direction O_t, omitting ϵ, the proposed increments give

$$\rho_t = \begin{cases}\widetilde{\eta}_t\|O_t\|_F/\|\mathbf{W}_t\|_F, & \text{Muon},\\ \eta_t R/\|\mathbf{W}_t\|_F, & \texttt{MACRO}\text{-fro},\\ \eta_t R\|O_t\|_F/(\|O_t\|_2\|\mathbf{W}_t\|_F), & \texttt{MACRO}\text{-spec}.\end{cases} \tag{67}$$

Each row uses its own optimizer direction and weight matrix. For an idealized full-rank matrix-sign direction, $\|O_t\|_2 = 1$ and $\|O_t\|_F = \sqrt{d}$, where $d = \min\{p,n\}$. Together with exact norm constraints, this gives the main-text comparison in Eq. equation 6. For an arbitrary nonzero spectral update direction, the exact norm-ratio identity is

$$\rho_t = \eta_t\frac{R}{\|\mathbf{W}_t\|_2}\sqrt{\frac{\kappa(\mathbf{W}_t)}{\kappa(O_t)}}, \qquad \kappa(A) = d\frac{\|A\|_2^2}{\|A\|_F^2}. \tag{68}$$

The full-rank matrix-sign idealization sets $\kappa(O_t) = 1$. Substituting the corresponding ρ_t into $\mathcal{R}(\rho_t)$ gives the method-specific retention expressions used in the main text.

D Numerical Experiments Supplementary Materials

This appendix provides supplementary experimental details for the RMSNorm study in Section 4.1, the weight-decay and dynamics study in Section 4.2, the precision study in Section 4.3, and the standard pre-training experiments in Section 5. The configuration for each experiment group is reported in its corresponding subsection. Computational-efficiency results are reported in Table 17.

D.1 Numerical Experiments Details in Section 4.1

Experimental Setting. We test whether MACRO can train a 330M-parameter Transformer to convergence with all learnable RMSNorm layers removed. The model is a QWEN3-like architecture with RoPE, GQA, and Norm-Gated SwiGLU, augmented by the parameter-free normalizations introduced in Section 4.1: a parameter-free RMSNorm after the attention block, QK-norm, and the Norm-Gated SwiGLU activation. Table 5 reports the full model and training configuration.

Category	Parameter	330M
Model Configuration	dimension	1024
	number of layers	24
	heads	16
	kv-heads	8
	sequence length	1024
	vocabulary size	50304
	Embedding Tie	True
Train Configuration	batch size	64
	gradient accumulation step	2
	lr schedule	linear warmup + cosine decay
	warmup steps	850
	cosine decay lr	$10^{-3} \times lr$
	Train Steps	8500
	# GPUs	8
	# Tokens	8.9B

Table 5: Model architecture and training configuration for the normalization-free 330M QWEN3-like experiment in Section 4.1.

Hyperparameter Sweep and Selection. For each of Muon, MACRO-spec, and MACRO-fro, we sweep the learning rate η_t over $\{3 \times 10^{-3}, 5 \times 10^{-3}, 7 \times 10^{-3}, 1 \times 10^{-2}, 3 \times 10^{-2}\}$. For the two constrained variants, we additionally sweep the radius hyperparameter r over $\{0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5\}$. Additionally, we also search qkv-split for MACRO-spec following Xie et al. (2026). Table 6 summarizes the search ranges, and Table 7 reports the per-optimizer hyperparameters that produce the bolded entries of Table 2 in the main text.

Optimizer	learning rate η_t	radius r	qkv-split
Muon	$[3 \times 10^{-3}, 3 \times 10^{-2}]$	-	-
MACRO-spec	$[3 \times 10^{-3}, 3 \times 10^{-2}]$	$[0.7, 1.5]$	[True, False]
MACRO-fro	$[3 \times 10^{-3}, 3 \times 10^{-2}]$	$[0.7, 1.5]$	-

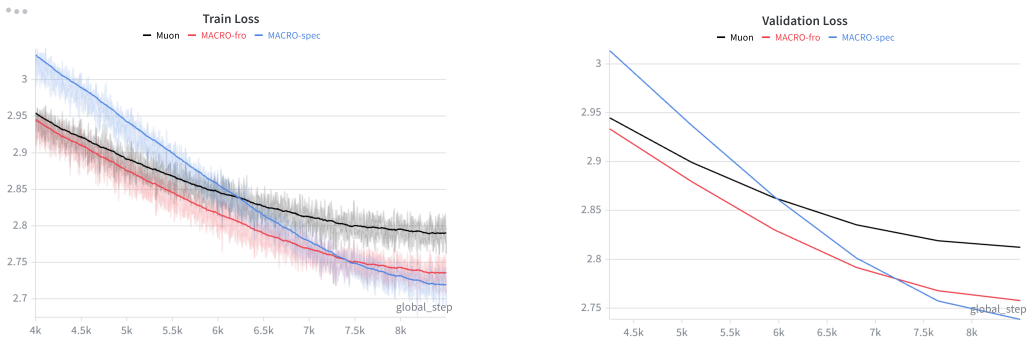
Table 6: Hyperparameter search ranges for each optimizer in the normalization-free setting.

Numerical Result Plots. Figure 7 plots the training and validation loss across the 8500 training steps for all three optimizers at the chosen learning rate $\eta_t = 10^{-2}$, using the radii reported in Table 7. These curves give the per-step picture behind the best result in Table 2.

Ablation Study. To validate the architecture design in Section 4.1, we run MACRO-spec while toggling three architectural choices: the activation (standard SwiGLU, Norm-Gated SwiGLU, or ReLU), the parameter-free RMSNorm after the attention block (Att-Norm), and the QK-norm. Each configuration is trained at three constraint radii $r \in \{1.1, 1.2, 1.3\}$. Table 8 reports the resulting

Parameter	Muon	MACRO-spec	MACRO-fro
learning rate η_t	0.01	0.01	0.01
weight decay	0.0	0.0	0.0
radius r	-	1.2	0.8
alignment c	-	1.0	1.0
qkv-split	-	False	-

Table 7: Selected hyperparameters per optimizer used to produce the bolded entries of Table 2.



(a) Training Loss.

(b) Validation Loss.

Figure 7: Training and validation loss for Muon, MACRO-fro, and MACRO-spec on the 330M QWEN3-like model with all learnable RMSNorm layers removed, at the chosen learning rate $\eta_t = 10^{-2}$ (Table 7).

validation loss. Standard SwiGLU diverges to `NaN` at every tested radius, even with both Att-Norm and QK-norm enabled. The Norm-Gated SwiGLU activation is the component that recovers stability, and adding Att-Norm and QK-norm on top yields the lowest validation loss across all three radii. ReLU is stable but underperforms Norm-Gated SwiGLU with both Att-Norm and QK-norm enabled.

Activation	**Att-Norm**	**QK-norm**	$r = 1.1$	$r = 1.2$	$r = 1.3$
SwiGLU	True	True	`NaN`	`NaN`	`NaN`
Norm-Gated SwiGLU	False	False	2.76	2.762	2.77
Norm-Gated SwiGLU	True	False	2.747	2.746	2.749
Norm-Gated SwiGLU	True	True	2.744	2.742	2.743
ReLU	True	True	2.762	2.755	2.757

Table 8: Ablation Study on RMSNorm-free training

D.2 Numerical Experiments Details in Section 4.2

This subsection supplies the empirical evidence for the three claims of Section 4.2. Manifold constraints subsume the regularization role of decoupled weight decay. Moreover, the locked update-weight-ratio enables zero-shot Maximal Update Parametrization (μP) transfer across model widths. The gradient-norm, 330M weight-decay ablation, and μP experiments reuse the QWEN3-like configurations summarized in Table 9.

Category	**Parameter**	120M	330M
Model Config	dimension	768	1024
	number of layers	12	24
	heads	6	16
	kv_heads	3	8
	sequence length	1024	1024
	vocabulary size	50304	50304
	Embedding Tie	True	True
Train Config	batch size	128	64
	gradient accumulation step	1	2
	warmup steps	360	850
	cosine decay lr	$10^{-3} \times lr$	$10^{-3} \times lr$
	Train Steps	3600	8500
	# GPUs	8	8
	# Tokens	3.7B	8.9B

Table 9: Training configuration referenced by the 120M and 330M dynamics experiments in Section 4.1 and Section 4.2.

Weight Decay Ablation. Section 4.2 argues that manifold constraints subsume the regularization effect of decoupled weight decay. To verify this empirically, we train the 330M QWEN3-like model with each of MuonH-spec, MuonH-fro, `MACRO`-spec, and `MACRO`-fro under two settings: $\lambda = 0.0$ (no weight decay) and $\lambda = 0.1$. We sweep the learning rate over $\eta_t \in \{0.007, 0.01, 0.03\}$ and hold every other hyperparameter fixed at the values in Table 18. Table 10 reports the resulting validation loss. For every optimizer and every learning rate, the two λ settings agree within 0.001. Adding weight decay therefore has no measurable effect once a manifold constraint is active, which confirms the substitution claim.

Optimizer	$\eta_t = 0.007$	$\eta_t = 0.01$	$\eta_t = 0.03$
MuonH-spec ($\lambda = 0.0$)	2.728	2.715	2.766
MuonH-spec ($\lambda = 0.1$)	2.729	2.715	2.766
MuonH-fro ($\lambda = 0.0$)	2.740	2.722	2.717
MuonH-fro ($\lambda = 0.1$)	2.740	2.722	2.717
MACRO-spec ($\lambda = 0.0$)	2.728	2.715	2.783
MACRO-spec ($\lambda = 0.1$)	2.727	2.715	2.783
MACRO-fro ($\lambda = 0.0$)	2.742	2.730	2.718
MACRO-fro ($\lambda = 0.1$)	2.742	2.729	2.718

Table 10: Weight-decay ablation on the 330M QWEN3-like model. Each manifold-constrained optimizer is trained with weight-decay coefficient $\lambda \in \{0.0, 0.1\}$ at three learning rates η_t. Across every cell, the two λ settings differ by at most 0.001, which confirms that manifold constraints subsume the regularization role of decoupled weight decay. This phenomenon contradicts the Muon which benefits from weight decay widely reported in the literature.

Joint Learning-Rate and Weight-Decay Sweep for Muon. We next ask whether jointly tuning Muon's learning rate and weight decay closes the performance gap with MACRO without weight decay on the constrained matrices. Table 11 reports a separate sweep on the 240M QWEN3-like model with the model config in Table 16, using learning rates $\{0.008, 0.016, 0.032\}$ and weight-decay coefficients $\{0.01, 0.05, 0.10, 0.50\}$. Within the tested grid, Muon achieves a best validation PPL of 20.583, essentially matching 20.579 for MACRO-fro, while MACRO-spec achieves a lower PPL of 20.502.

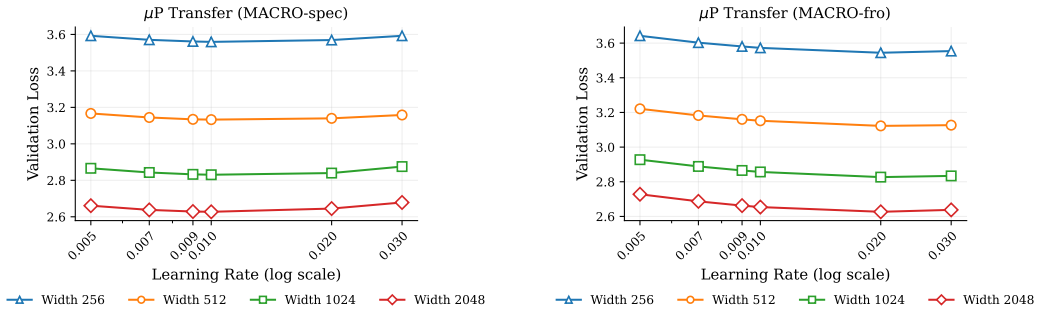
Table 11: Joint learning-rate and weight-decay sweep for Muon on the 240M QWEN3-like model, with MACRO variants as references.

Optimizer	**Learning Rate**	**Metric**	**Weight Decay**			
			0.01	0.05	0.10	0.50
Muon	0.008	Train	3.0731	3.0579	**3.0499**	3.0679
		Val PPL	21.348	21.020	**20.845**	21.168
	0.016	Train	3.0553	**3.0420**	3.0427	3.1022
		Val PPL	20.972	20.680	**20.676**	21.874
	0.032	Train	3.0420	**3.0382**	3.0524	3.1686
		Val PPL	20.691	**20.583**	20.848	23.345
MACRO-fro	0.016	Train	3.0347 (weight decay = 0.0)			
		Val PPL	20.579 (weight decay = 0.0)			
MACRO-spec	0.008	Train	**3.0308** (weight decay = 0.0)			
		Val PPL	**20.502** (weight decay = 0.0)			

The MACRO-fro and MACRO-spec reference results use weight decay 0.0. Their result cells are merged because weight decay is not swept for these configurations.

μP Transfer. The locked relative learning rate also enables zero-shot μP transfer across model widths. To make the experiment compatible with μP (Yang et al., 2023), we initialize the linear-layer weights from $\mathcal{N}(0, \sigma^2)$ with $\sigma = \min\left\{\sqrt{D_{\text{out}}/D_{\text{in}}}, 1\right\}/\sqrt{D_{\text{in}}}$ and set the Frobenius radius to $R = r \min\left\{\sqrt{D_{\text{out}}/D_{\text{in}}}, 1\right\}\sqrt{D_{\text{out}}}$, which absorbs the architecture-dependent $\Theta(1)$ constant into the tunable hyperparameter r (see Section 3.2). For Spectral Sphere $\mathcal{M}_R$ the radius

is same with the radii in Section 3.2. Holding the rest of the 330M training configuration of Table 9 fixed, we vary the model width across $\{256, 512, 1024, 2048\}$ and the learning rate across $\{0.005, 0.007, 0.009, 0.01, 0.02, 0.03\}$. Figure 8 reports the resulting validation loss for both MACRO-spec and MACRO-fro. Within each width, the validation-loss curve is nearly flat across the learning-rate sweep, and the minimum sits at the same learning rate ($\eta_t \approx 0.01$ for MACRO-spec and $\eta_t \approx 0.02$ for MACRO-fro) across all four widths. The locked $\eta_{\text{rel},t}$ therefore delivers zero-shot μP transfer in practice.



(a) MACRO-spec. (b) MACRO-fro.

Figure 8: Validation loss versus learning rate at four model widths $\{256, 512, 1024, 2048\}$ for the 330M QWEN3-like configuration of Table 9. Each width is trained with the same per-width radius r and the same alignment constant c. The optimum learning rate is consistent across widths, which empirically verifies zero-shot μP transfer for both MACRO-spec and MACRO-fro.

Learning-Rate Warmup. Table 12 compares Muon and the two MACRO variants using warmup ratios $\{0\%, 5\%, 10\%, 15\%\}$ at 240M and $\{0\%, 5\%, 15\%\}$ at 600M, reporting training loss averaged over the last 5% of training and final validation perplexity. Both 240M model and 600M model follow the experimental setting in Table 16. Across the tested 240M and 600M configurations, both MACRO variants achieve their lowest validation perplexity without warmup. Muon benefits from 5% warmup at 240M; at 600M, its 0% and 5% results are close (validation PPL 15.312 and 15.309).

Table 12: Effect of warmup ratio on the last-5% training loss and final validation perplexity for unconstrained Muon and constrained MACRO variants.

Optimizer	Warmup	240M		600M	
		Train	Val PPL	Train	Val PPL
Muon	0%	3.0484	20.868	2.7418	15.312
	5%	**3.0379**	**20.650**	**2.7416**	**15.309**
	10%	3.0396	20.679	–	–
	15%	3.0413	20.712	2.7479	15.402
MACRO-fro (Ours)	**0%**	**3.0328**	**20.543**	**2.7347**	**15.227**
	5%	3.0336	20.560	2.7363	15.252
	10%	3.0357	20.604	–	–
	15%	3.0393	20.674	2.7421	15.340
MACRO-spec (Ours)	**0%**	**3.0334**	**20.559**	**2.7343**	**15.216**
	5%	3.0370	20.632	2.7366	15.253
	10%	3.0404	20.700	–	–
	15%	3.0430	20.752	2.7435	15.354

D.3 Numerical Experiments for Section 4.3

Experimental setup. We compare Muon, MACRO-fro, and MACRO-spec on the 240M and 600M QWEN3-like models (model setting is in 16) using TorchTitan (Liang et al., 2024). The precision comparison changes `training.dtype` from `float32` to `bfloat16`. The BF16 setting reduces the storage precision of parameters, gradients, and optimizer states without an additional FP32 master-weight copy; it does not require every intermediate operation to use BF16. We retain the implementation's existing higher-precision operations and add no stochastic rounding, compensated accumulation, or error-feedback buffers. All runs use seed 1. The comparisons use the same learning-rate schedule in both precisions for each optimizer, with minimum lr decay factor $f = 0.01$.

Performance with BF16 master weight training. Table 13 preserves the full training-loss and validation-perplexity comparison. Training loss is averaged over the last 5% of each run; validation perplexity is computed from the final validation loss. The training-gap column retains the convention $\Delta_{\text{Train}} = L_{\text{train}}^{\text{FP32}} - L_{\text{train}}^{\text{BF16}}$.

Table 13: Performance under the FP32 and BF16 training configurations, with the BF16 learning-rate floors specified below. Training loss is averaged over the last 5% of training.

Model	Optimizer	FP32 Train	BF16 Train	Δ_{Train}	FP32 PPL	BF16 PPL
240M	Muon	3.0373	4.1813	-1.1440	20.593	64.845
	MACRO-fro	3.0311	3.0435	-0.0124	20.494	20.748
	MACRO-spec	3.0432	3.0706	-0.0274	20.732	21.324
600M	Muon	2.7446	3.6801	-0.9356	15.349	39.269
	MACRO-fro	2.7392	2.7611	-0.0219	15.296	15.665
	MACRO-spec	2.7405	2.7859	-0.0453	15.310	16.056

D.3.1 Update Cancellation Estimates under cosine decay LR schedule

To analyze the performance gap between precisions shown in the Fig 4, we substitute the ρ_t from Eq. equation 6 into the Proposition 2:

$$\mathcal{R}_t^{\text{Muon}} = \left[1 + \left(\frac{\tau_0 \|\mathbf{W}_t\|_F}{\widetilde{\eta}_t \sqrt{d}}\right)^2\right]^{-1/2}, \mathcal{R}_t^{\text{fro}} = \left[1 + \left(\frac{\tau_0}{\eta_t}\right)^2\right]^{-1/2}. \tag{69}$$

Figure 10 illustrates that the parameter norm of Muon under BF16 grows to approximately 129, far above the norm under FP32 precision. Such growth drives $\mathcal{R}_t^{\text{Muon}}$ up to 0.7 when $\widetilde{\eta}_t = 0.008$. In contrast, MACRO-fro's retention depends directly on η_t, without the additional weight-norm factor.

Under the cosine decay LR scheduler, consider the continuous cosine schedule

$$\eta_t = \eta_{\max}\left[f + \frac{1-f}{2}\left(1 + \cos\left(\pi \frac{t - T_w}{T - T_w}\right)\right)\right], \qquad T_w \le t \le T, \tag{70}$$

where $0 \le f < 1$ and $0 \le T_w < T$. The schedule decreases from $\eta_{\max}$ to $f\eta_{\max}$. For $\eta_\star \in (f\eta_{\max}, \eta_{\max})$, solving $\eta_{t_\star} = \eta_\star$ gives

$$t_\star = T_w + \frac{T - T_w}{\pi} \arccos\left(\frac{2(\eta_\star/\eta_{\max} - f)}{1-f} - 1\right). \tag{71}$$

The numerical step estimates use this continuous interpolation and are approximate.

For the idealized Frobenius-normalized step, $\rho_t = \eta_t$. Two markers follow from Proposition 2:

$$\eta_t = \tau \quad \Longrightarrow \quad p_{\text{drop}} = \frac{1}{2}, \qquad \mathcal{R} = \frac{1}{\sqrt{2}},$$
$$\eta_t = \frac{\tau}{\sqrt{3}} \quad \Longrightarrow \quad \mathcal{R} = \frac{1}{2}.$$

We denote the second marker by $t_{1/2}$. More generally, a retained fraction $h \in (0, 1)$ corresponds to

$$\mathcal{R}(\eta_t) = h \quad \Longleftrightarrow \quad \eta_t = \frac{\tau h}{\sqrt{1-h^2}}.$$

A marker is reached during decay only if its corresponding learning rate lies within the schedule range. These markers specify cancellation levels in the model, not the first step with a nonzero validation-loss gap. The 240M numerical estimates and observed gaps are reported in Appendix D.3.1.

For the 240M Frobenius comparison, set

$$\eta_{\max} = 0.016, \qquad T_w = 920, \qquad T = 9200, \qquad \tau = \tau_0 = 2^{-9}.$$

The threshold is the same format-based reference used in the main text, not a parameter fitted to the loss curves. With $f = 0.01$, Eq. equation 71 gives $t_{1/2} \approx 7884$, or 85.7% of training. Table 14 separates computed model quantities from measured validation gaps. The predicted retention decreases during late decay, while the observed gap grows after being small in the middle of training. An early precision gap also exists; the comparison concerns its renewed late-stage growth, not its first occurrence.

Step	η_t	Model $\mathcal{R}(\eta_t)$	Measured G_{val}
6800	3.223×10^{-3}	0.8552	0.000710
8200	7.233×10^{-4}	0.3473	0.007316
9200	1.600×10^{-4}	0.0816	0.017157

Table 14: Model retention estimates and observed validation-loss gaps for the 240M MACRO-fro comparison. Both precisions use `min_lr_factor`$= 0.01$. Retention estimates use $\tau_0 = 2^{-9}$ and are not measurements of realized BF16 updates.

With the BF16 floor increased to $f = 0.05$, the same model gives $t_{1/2} \approx 8423$, or 91.6% of training. Table 15 compares the two BF16 floors against the same FP32 reference at $f = 0.01$. The final model retention increases from 0.0816 to 0.3790, while the measured final gap decreases from 0.017157 to 0.012312. The model marker is a reference cancellation level, not a predicted validation-loss changepoint.

BF16 floor f	Model $t_{1/2}$	Model $\mathcal{R}(\eta_T)$	Measured final G_{val}
0.01	7884	0.0816	0.017157
0.05	8423	0.3790	0.012312

Table 15: Learning-rate-floor comparison for 240M MACRO-fro. The FP32 reference uses $f = 0.01$ in both rows. Model quantities use $\tau_0 = 2^{-9}$ and the continuous cosine-schedule interpolation. The validation gap is $G_{\text{val}} = L_{\text{val}}^{\text{BF16}} - L_{\text{val}}^{\text{FP32}}$.

Supplementary training-loss trajectories. Figure 9 contains the training-loss comparisons moved from the main-text performance figure. The plotted gap uses the original sign convention $L_{\text{train}}^{\text{FP32}} - L_{\text{train}}^{\text{BF16}}$.

D.4 Numerical Experiments Details in Section 5

Experimental Setup. Section 5 evaluates MACRO on QWEN3-like architectures at the 240M, 600M, and 1.2B scales using the TorchTitan (Liang et al., 2024) training pipeline. Table 16 reports the resolved configurations of the three representative MACRO-fro runs. All three runs use C4 sequences of length 2048 and a global batch size of 512.

Algorithm Efficiency Comparison Table To compare training efficiency across optimization algorithms and manifold constraints, we report per-GPU token throughput (TPS), per-GPU compute throughput (TFLOPS), model FLOPs utilization (MFU), and end-to-end (E2E) step time. All metrics are computed over the training steps and averaged across five random seeds. The results are summarized in the following table. Higher TPS, TFLOPS, and MFU indicate better efficiency, whereas lower E2E step time is better.

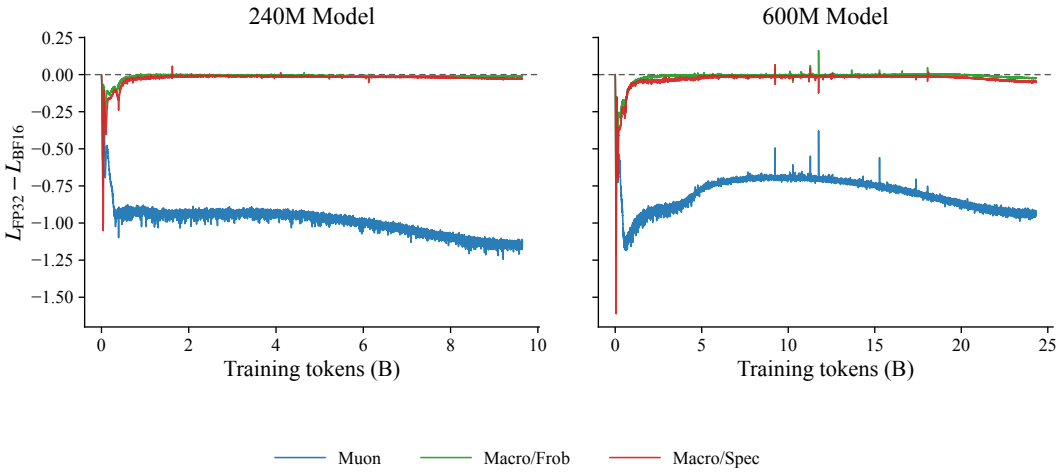


Figure 9: Training-loss differences between the FP32 and BF16 configurations over cumulative training tokens for the 240M and 600M models. Negative values of $L_{\text{train}}^{\text{FP32}} - L_{\text{train}}^{\text{BF16}}$ indicate higher training loss in BF16.

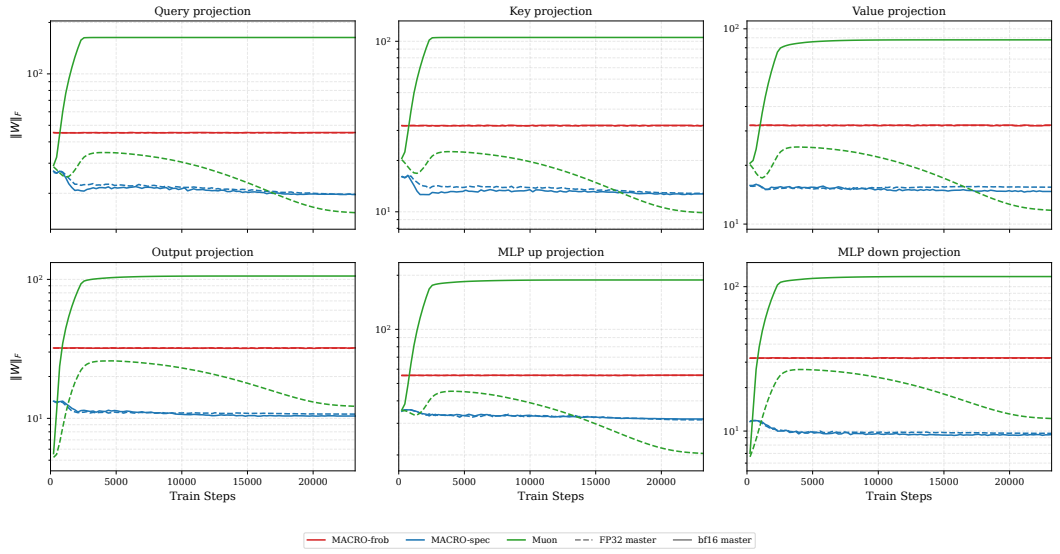


Figure 10: Parameter norm differences between the FP32 and BF16 configurations over cumulative training steps. The norm dynamics under Muon will below up at the early stage of Muon.

Category	Parameter	240M	600M	1.2B
Model Config	model flavor	`qwen3_120m_split`	`qwen3_0_6b_split`	`qwen3_1_2b_split`
	dimension	768	1024	1536
	number of layers	12	28	28
	heads	12	16	12
	kv_heads	4	8	4
	sequence length	2048	2048	2048
	vocabulary size	151936	151936	151936
	Embedding Tie	True	True	True
Train Config	training framework	TorchTitan	TorchTitan	TorchTitan
	training dataset	C4	C4	C4
	global batch size	512	512	512
	local batch size	64	32	32
	warmup steps (ratio)	920 (10%)	2320 (10%)	4600 (10%)
	minimum LR factor	0.01	0.01	0.01
	completed steps	9200	23200	46000
	total accelerators	8 × H100	16 × H100	16 × H100
	tokens seen	9.647B	24.327B	48.234B

Table 16: model and training configuration for the representative 240M, 600M, and 1.2B.

- MACRO vs. Muon: MACRO-fro shows only a small efficiency gap compared with Muon. Its TPS is 204.4K, compared with 207.7K for Muon (-1.6%), while its MFU decreases by only 0.62 percentage points. The E2E step time increases from 0.631 s to 0.641 s (+1.6%, or about 10 ms).
- Constraint Comparison: The Frobenius constraint is consistently more efficient than the spectral constraint, which is consistent with the algorithm design. The spectral sphere requires power iteration for tangent-space computation and retraction, making it more expensive than Frobenius norm computation.
- Algorithm Comparison: Under the Frobenius constraint, the efficiency ranking is MuonH-Frob $\approx$ MACRO-fro > FSO. Under the spectral constraint, the ranking is MuonH-Spec $\approx$ MACRO-Spec > SSO. This can be explained by the algorithm design. SSO and FSO use an inner loop to solve the tangent-space projection. With a maximum of 20 inner-loop iterations, they evaluate the Msign function up to 20 times, whereas Muon, MuonH, and MACRO evaluate it only once. Therefore, SSO and FSO are less efficient than the other manifold-constrained Muon variants under our setting.

Table 17: Efficiency comparison across different optimizers. E2E denotes the end-to-end time per training step.

Optimizer	E2E (s) ↓	TPS/GPU ↑	TFLOPS/GPU ↑	Avg. MFU ↑
Muon	0.6311	207,693	392.40	39.68%
`MACRO`-spec	0.7124	183,982	347.60	35.15%
`MACRO`-fro	0.6411	204,440	386.26	39.06%
MuonH-spec	0.6864	190,944	360.76	36.48%
MuonH-fro	**0.6156**	**212,910**	**402.26**	**40.67%**
SSO	1.7452	75,104	141.90	14.35%
FSO	1.2086	108,454	204.91	20.72%

We clarify that this efficiency comparison is based on plain implementations without specialized kernel optimizations when training a 240M-parameter model. The infrastructure design using Megatron in the SSO paper could narrow the efficiency gap.

Hyperparameter Sweep and Selection. We fix $\beta_1 = 0.9$ for every optimizer. For the SSO and FSO double-loop solvers, we set the subproblem tolerance to 10^{-3} and cap the inner-loop iterations at 8. For Muon, MACRO-spec, MACRO-fro, MuonH-spec, and MuonH-fro, we sweep the learning rate η_t over $\{2.0 \times 10^{-3}, 4.0 \times 10^{-3}, 8.0 \times 10^{-3}, 1.6 \times 10^{-2}, 3.2 \times 10^{-2}\}$. Table 18 reports the chosen hyperparameters that produce the bold entries of Table 4 in the main text.

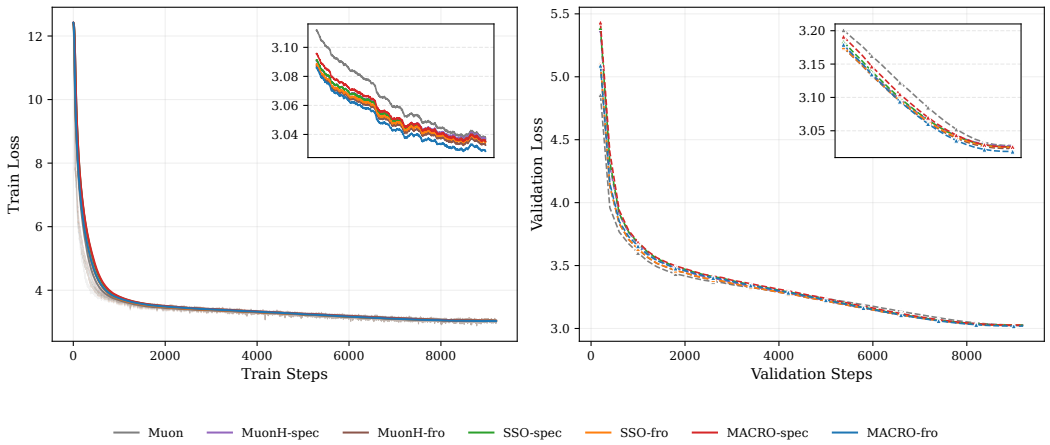
Model	Optimizer	learning rate	weight decay	r
240M	Muon	0.016	0.1	-
	MuonH-spec	0.008	0.0	2.0
	MuonH-fro	0.016	0.0	1.0
	SSO	0.008	0.0	2.0
	FSO	0.016	0.0	1.0
	MACRO-spec	0.008	0.0	2.0
	MACRO-fro	0.016	0.0	1.0
600M	Muon	0.008	0.1	-
	MuonH-spec	0.008	0.0	2.0
	MuonH-fro	0.008	0.0	1.0
	SSO	0.008	0.0	2.0
	FSO	0.008	0.0	2.0
	MACRO-spec	0.008	0.0	2.0
	MACRO-fro	0.008	0.0	1.0
1.2B	Muon	0.008	0.1	-
	MuonH-spec	0.004	0.0	2.0
	MuonH-fro	0.008	0.0	1.0
	MACRO-spec	0.004	0.0	2.0
	MACRO-fro	0.008	0.0	1.0

Table 18: Selected hyperparameters per optimizer per model scale, used to produce the bold entries of Table 4 in the main text.

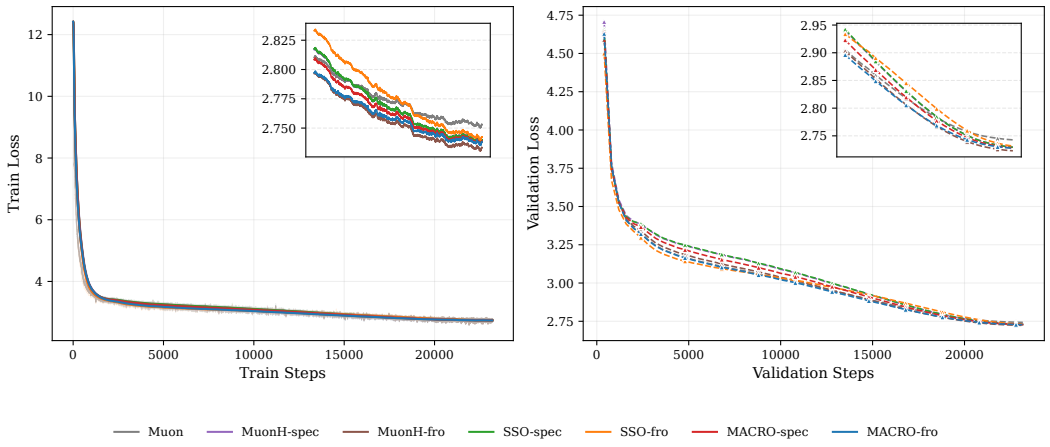
Per-Step Training and Validation Loss Curves. Figure 11 plots the training and validation loss across the full training schedule for each model scale. The curves give the per-step trajectory behind the final numbers reported in Table 4. All manifold-constrained optimizers (MACRO-spec, MACRO-fro, MuonH-spec, MuonH-fro, SSO, FSO) track the unconstrained Muon baseline closely throughout training, and they are superior than Muon in the late stage of training. Across the three scales, MACRO-spec and MACRO-fro reach validation losses comparable to the heuristic MuonH counterparts and to the exact double-loop SSO and FSO baselines.

Tangent Space Violation Ablation. MACRO replaces the inner bisection of SSO and FSO with a single-step tangent space projection (Line 5 of Algorithm 1). To quantify the resulting trade-off, we measure the per-step tangent space violation, defined as the residual norm $\mathrm{vio}(\mathbf{W}_t) := |\langle \Gamma_t, \widetilde{\nabla}_t \rangle|$ between the proposed update direction $\widetilde{\nabla}_t$ and the normal vector to the constraint surface Γ_t defined in Appendix B.1. Figure 12 reports the average of $\mathrm{vio}(\mathbf{W}_t)$ across the full 8500-step training schedule of the 330M model, separated by linear-layer type (FFN-Up, Attn-Out, Attn-Value, Attn-Query, Attn-Key) and by manifold ($\mathcal{M}_S$ on the left, $\mathcal{M}_F$ on the right). Solid curves report MACRO; dashed curves report the corresponding exact double-loop solver (SSO for $\mathcal{M}_S$, FSO for $\mathcal{M}_F$). Two conclusions follow.

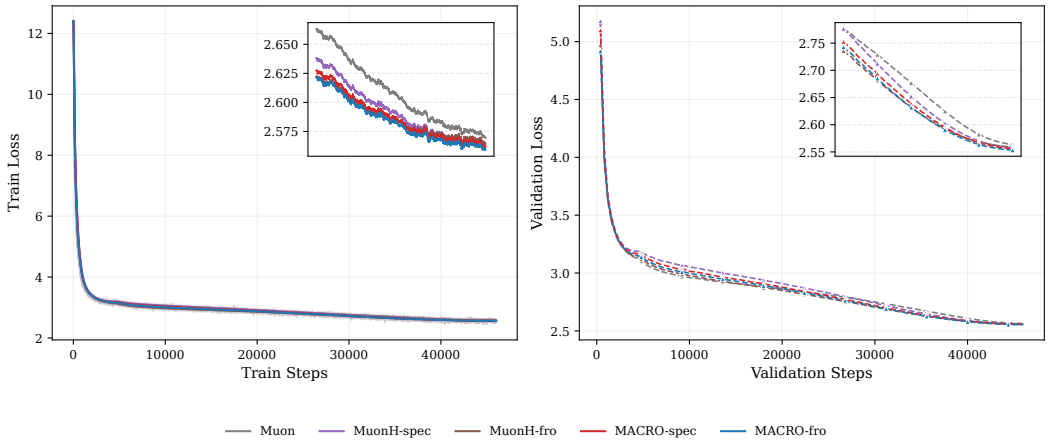
First, MACRO keeps the absolute tangent space violation small. The solid curves stabilize at roughly 10^{-3} to 10^{-2} across all five layer types and both manifolds. The single-step projection therefore lands near, but not exactly on, the tangent space. The double-loop dashed curves stabilize at roughly 10^{-4}, one to two orders of magnitude lower. Second, this larger residual does not translate into worse



(a) Train and Validation Loss for 240M training.



(b) Train and Validation Loss for 600M training.



(c) Train and Validation Loss for 1.2B training. SSO and FSO are omitted at this scale because of their double-loop computational cost (see Table 4).

Figure 11: Training (left) and validation (right) loss across the full training schedule for the 120M, 330M, and 1B QWEN3-like models. Each subfigure compares Muon, MuonH-spec, MuonH-fro, SSO, FSO, MACRO-spec, and MACRO-fro under the chosen hyperparameters from Table 18. These curves are the per-step trajectories underlying Table 4.

validation loss. Table 4 shows that MACRO-spec and MACRO-fro reach validation losses comparable to SSO and FSO on 120M model and 330M model. Removing the inner bisection therefore eliminates a large per-step computation overhead at no measurable cost in validation loss, and MACRO attains a favorable balance between tangent space residual and per-step computation.

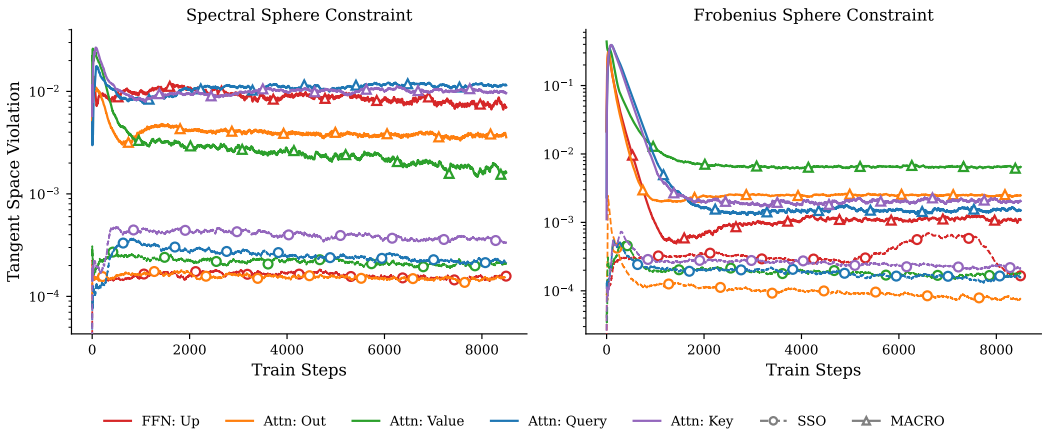


Figure 12: The average of Per-step tangent space violation $\mathrm{vio}(\mathbf{W}_t) := |\langle \Gamma_t, \widetilde{\nabla}_t \rangle|$, where $\widetilde{\nabla}_t$ is the update direction and Γ_t is the normal vector to the constraint surface, across the 8500-step training schedule of the 330M QWEN3-like model. The left panel reports the spectral sphere ($\mathcal{M} = \mathcal{M}_S$); the right panel reports the Frobenius sphere ($\mathcal{M} = \mathcal{M}_F$). Each colored curve corresponds to a linear-layer type (FFN-Up, Attn-Out, Attn-Value, Attn-Query, Attn-Key); solid curves report MACRO (single-step projection) and dashed curves report the corresponding exact double-loop solver (SSO for $\mathcal{M}_S$, FSO for $\mathcal{M}_F$). Across all five layer types and both manifolds, MACRO stabilizes at residuals on the order of 10^{-3} to 10^{-2}, one to two orders of magnitude above the double-loop baseline. Table 4 shows that this larger residual does not degrade validation loss.

D.5 Empirical Equivalence of Oblique and Frobenius Constraints in Section 2

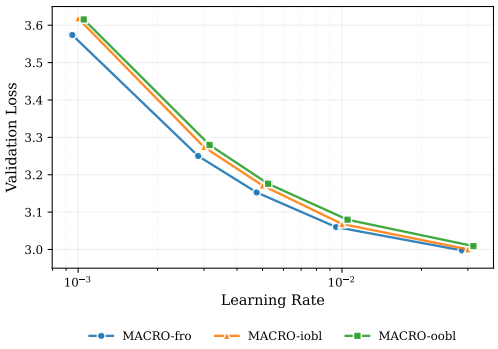
In Section 2, we introduced Oblique manifolds as candidates for row-wise or column-wise constraints. In this appendix, we provide the detailed empirical justification for omitting these Oblique constraints in favor of the Frobenius Sphere.

Generalization Performance Advantage. We first compare the generalization performance of the global Frobenius constraint ($\mathcal{M}_F$) against the Input and Output Oblique constraints ($\mathcal{M}_{O_{\text{in}}}$ and $\mathcal{M}_{O_{\text{out}}}$). Figure 13 (a) plots the validation loss across various learning rates. The results show that the Frobenius constraint consistently achieves lower validation loss than both Oblique constraints. This clear performance gap provides the primary motivation for adopting the Frobenius sphere.

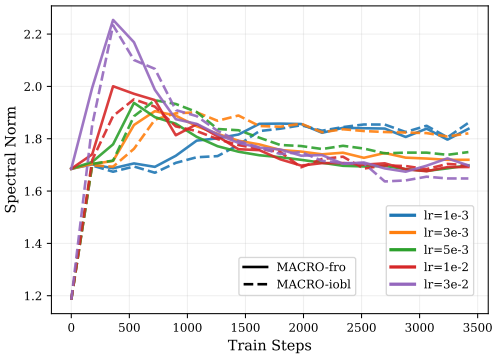
Equivalence of Internal Training Dynamics. To understand why Oblique constraints do not improve performance, we analyze the internal optimization dynamics. We specifically compare the global Frobenius constraint against the Input Oblique constraint. Figure 13 (b) and Figure 13 (c) track the spectral norm and the maximum row norm ($2 \to \infty$ norm), respectively. For most learning rates, the trajectories under the Frobenius constraint closely match the trajectories under the explicitly normalized Input Oblique constraint. This matching behavior indicates that a Frobenius naturally regulates individual feature dimensions. The neural network implicitly maintains a uniform norm distribution without requiring strict row-wise or column-wise limitations.

Adaptive Capacity at Extreme Learning Rates. While the dynamics are generally similar, Figure Figure 13 (d) illustrates the maximum column norm ($1 \to \infty$ norm) dynamics. At lower and standard learning rates, both constraints produce similar flat trajectories. However, at the highest learning rate ($\eta = 3 \times 10^{-2}$), the trajectories separate. The Input Oblique constraint strictly limits the column norm, keeping the trajectory flat. In contrast, the Frobenius constraint allows this specific norm to

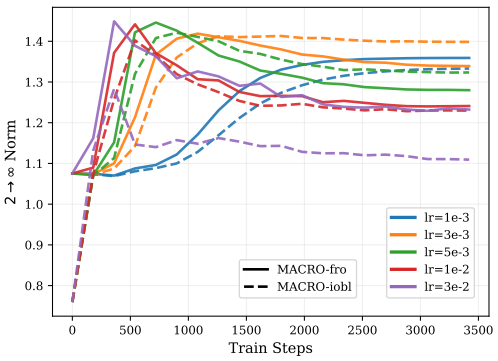
adaptively increase. This flexibility suggests that the Frobenius sphere provides a better adaptive balance. It regularizes the overall weight matrix but allows specific feature dimensions to adapt when necessary. Because Oblique constraints offer no performance benefits and restrict this adaptive capacity, we omit them from our core theoretical and empirical analyses.



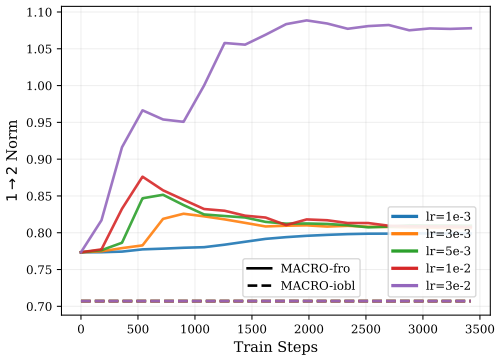
(a) **Validation loss across different learning rates**. We compare the performance of MACRO-fro, MACRO-iobl, and MACRO-oobl. The x-axis represents the learning rate on a logarithmic scale. The global Frobenius constraint (MACRO-fro) consistently achieves the lowest validation loss across all evaluated learning rates compared to the Oblique constraints.



(b) **Spectral norm dynamics during training.** We track the spectral norm for MACRO-fro (solid lines) and MACRO-iobl (dashed lines) across five different learning rates. Despite the architectural differences between the constraints, the spectral norm trajectories remain highly similar throughout the training process.



(c) **Maximum $2 \to \infty$ Norm dynamics.** We compare the $2 \to \infty$ norm under the MACRO-fro and MACRO-iobl constraints. Similar to the spectral norm behavior, both constraints produce closely matching trajectories across the entire training period.



(d) **Maximum $1 \to 2$ Norm dynamics.** We analyze the $1 \to 2$ norm under the two constraints. For most learning rates, the trajectories match closely. However, at the learning rate $\eta = 3 \times 10^{-2}$, the Frobenius constraint allows the norm to increase, whereas the Input Oblique constraint strictly limits this growth.

Figure 13: Ablation study on geometric constraints.