Demystifying Manifold Constraints in LLM Pre-Training

Shiqian Ma

Data Science and AI Institute, Department of Applied Mathematics and Statistics
Johns Hopkins University

ECE Seminar,University of Rochester
September 30, 2026

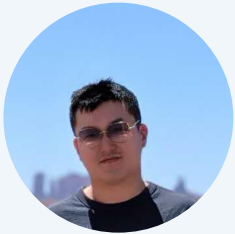
Research Supported by ONR and NSF (DMS, CCF, ECCS)







Joint Work With



Kang An

Rice University



Jiaxiang Li

Meta → Virginia Tech



Donald Goldfarb

Columbia University

Paper: Demystifying Manifold Constraints in LLM Pretraining. arxiv:2605.04418

Code: https://github.com/ErstinAn/titanprecond

Outline

- Muon
- Manifold Muon
- Comparison of Manifolds
- RMSNorm
- Weight Decay
- Numerical Experiments
- Conclusions

Outline

Muon

Manifold Muon

Comparison of Manifolds

RMSNorm

Weight Decay

Numerical Experiments

Conclusions

Improve the training stability by utilizating matrix structures

$$\min_{\mathbf{W}} \ \mathcal{L}(\mathbf{W}) := \mathbb{E}_{\xi}\ell(\mathbf{W}, \xi)$$

- LLM pre-training is highly nonconvex
- No clear rule how to make training algorithms convergent or stable

Heuristics to improve the training stability:

- Careful initialization of weights
- Residual connection: avoid vanishing gradients
- Activation normalization: LayerNorm, RMSNorm, and QK-Norm
- Learning-rate warmup and decay schedules
- Gradient clipping: control the gradient size
- Weight decay: avoid weights becoming too large

Neural-Network Weights Are Matrices

A linear layer is naturally a linear mapping:

$$\mathbf{x} \longmapsto \mathbf{W}\mathbf{x}, \qquad \mathbf{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}.$$

Adam: $\underbrace{\text{vec}(\mathbf{W})}_{\text{coordinatewise view}}$ versus $\underbrace{\mathbf{W} : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^{d_{\text{out}}}}_{\text{matrix/operator view}}$.

Method	**Matrix structure used by the optimizer**
K-FAC	Kronecker-factored curvature across layer inputs and outputs.
Shampoo	Left and right preconditioners from gradient statistics.
SOAP	Applies Adam in Shampoo's evolving matrix eigenbasis.
ASGO	One-sided preconditioner.
Muon	Orthogonalizes matrix momentum.

Martens and Grosse (2015); Gupta et al. (2018); Vyas et al. (2024); An et al. (2025).

Muon: control the size of the update

$$\min_{\mathbf{W}\in\mathbb{R}^{m\times n}} \mathcal{L}(\mathbf{W}) := \mathbb{E}_\xi \ell(\mathbf{W}, \xi)$$

Orthogonalize the momentum / gradient:

$$\begin{aligned}
\mathbf{G}_t &= \nabla_{\mathbf{W}} \ell(\mathbf{W}_t, \xi_t), \\
\mathbf{M}_t &= \beta \mathbf{M}_{t-1} + (1-\beta)\mathbf{G}_t, \\
\mathbf{O}_t &= \mathrm{argmin}_{\mathbf{O}\in\mathbb{R}^{m\times n}} \{\|\mathbf{O} - \mathbf{M}_t\|_F^2 \mid \mathbf{O}^\top \mathbf{O} = I_{n\times n} \text{ or } \mathbf{O}\mathbf{O}^\top = I_{m\times m}\} \\
\mathbf{W}_{t+1} &= \mathbf{W}_t - \eta_t \mathbf{O}_t.
\end{aligned}$$

Orthogonalization admits a closed-form solution:

$$\mathbf{O}_t = UV^\top = \mathrm{msign}(\mathbf{M}_t), \text{ where } U, V \text{ are from the SVD of } \mathbf{M}_t = U\Sigma V^\top$$

Muon: MomentUm Orthogonalized by Newton–Schulz[1]

- SVD is expensive!
- However, the msign function can be computed iteratively by Newton-Schulz.

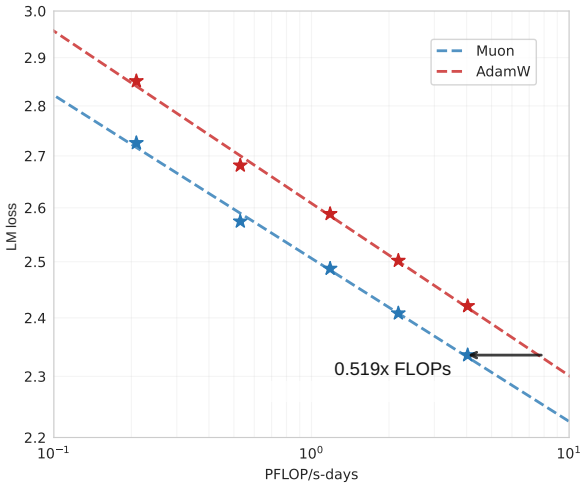
$$\text{msign}(M) = UV^\top = M(M^\top M)^{-1/2}$$

Muon Update

$$\begin{aligned}
\mathbf{G}_t &= \nabla_{\mathbf{W}} \ell(\mathbf{W}_t, \xi_t), \\
\mathbf{M}_t &= \beta \mathbf{M}_{t-1} + (1-\beta)\mathbf{G}_t, \\
\mathbf{O}_t &= \text{NS}_5(\mathbf{M}_t) \approx \text{msign}(\mathbf{M}_t), \\
\mathbf{W}_{t+1} &= \mathbf{W}_t - \eta_t \mathbf{O}_t.
\end{aligned}$$

[1] Jordan et al., "Muon: An Optimizer for Hidden Layers in Neural Networks," 2024.

Muon is more efficient

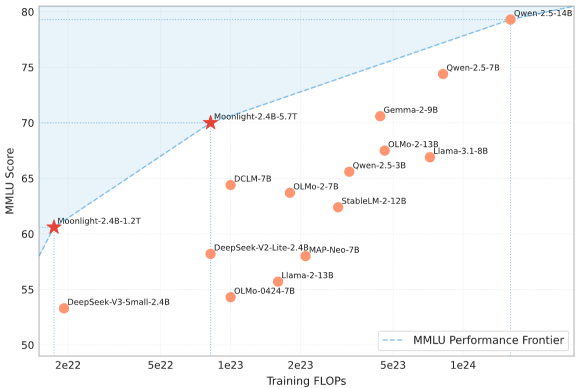


Empirical message

In the reported scaling-law fit, Muon is about $2x$ more computational efficient than Adam with compute optimal training

Source: Liu et al., *Muon is Scalable for LLM Training*, arXiv:2502.16982, 2025.

Muon advances the Pareto frontier



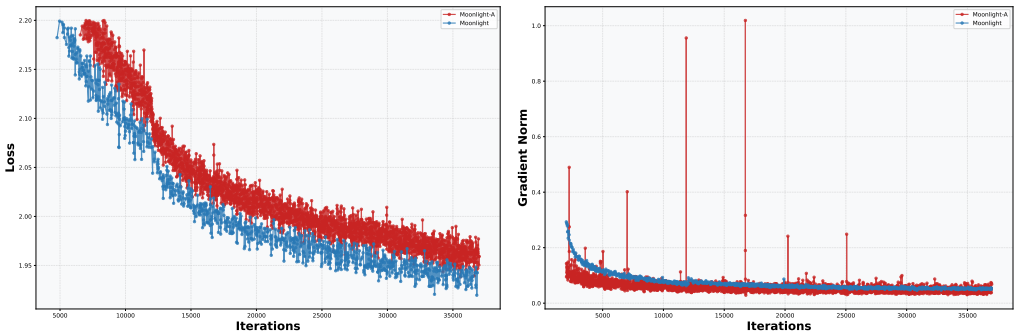
Empirical message

Moonlight optimized by Muon advances the Pareto frontier of performance vs training FLOPs.

Source: Liu et al., *Muon is Scalable for LLM Training*, arXiv:2502.16982, 2025.

Muon is stable

Moonlight: Muon **Moonlight-A: AdamW**



In the reported large-scale run, Moonlight has a smooth loss curve and **no loss or gradient-norm spikes**.

Source: Liu et al., *Muon is Scalable for LLM Training*, arXiv:2502.16982, 2025.

Muon Was Used to Train Frontier Models

2.4. Muon Optimizer

We employ the Muon (Jordan et al., 2024; Liu et al., 2025) optimizer for the majority of modules in DeepSeek-V4 series due to its faster convergence and improved training stability. The full algorithm of our Muon optimization is summarized in Algorithm 1.

Source: DeepSeek-AI, *DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence*, Sec. 2.4, p. 14, arXiv:2606.19348, 2026.

2.5 Per-Head Muon

Following Kimi K2, Kimi K3 adopts Muon [54] as the optimizer for its matrix parameters.

Source: Kimi Team, *Kimi K3: Open Frontier Intelligence*, Sec. 2.5, p. 10, arXiv:2607.24653v2, 2026.

Muon is Spectral Steepest Gradient Descent

Euclidean Gradient Descent:

$$\begin{cases} D_t = \operatorname{argmin}_D \langle \mathbf{G}_t, D \rangle + \dfrac{1}{2\eta} \|D\|_F^2 \\ \mathbf{W}_{t+1} = \mathbf{W}_t + D_t \end{cases}$$

Equivalent to

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \eta \mathbf{G}_t.$$

Muon replaces the Euclidean distance $\|D\|_F^2$ by spectral norm $\|D\|_{\text{spec}}$:

$$\begin{cases} D_t = \operatorname{argmin}_D \langle \mathbf{G}_t, D \rangle, \text{ s. t., } \|D\|_{\text{spec}} \leq \eta \\ \mathbf{W}_{t+1} = \mathbf{W}_t + D_t \end{cases}$$

Equivalent to

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \eta \operatorname{msign}(\mathbf{G}_t).$$

Outline

Muon

Manifold Muon

Comparison of Manifolds

RMSNorm

Weight Decay

Numerical Experiments

Conclusions

Manifold Muon

- The success of Muon suggests that normalization/orthogonalization is important

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \eta_t \mathbf{M}_t \qquad \text{SGD},$$

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \eta_t \,\mathrm{msign}(\mathbf{M}_t) \qquad \text{Muon}.$$

- Besides normalize/orthogonalize the momentum, can also normalize the weights
- This means adding constraints to the weights

$$\min \ \mathcal{L}(\mathbf{W}), \ \text{s.t.}, \ \mathbf{W} \in \mathcal{M}$$

- Riemannian/Manifold Optimization
- Questions:
  - Which manifold to use?
  - How to solve it?
  - What roles does the manifold constraint play?

Manifold Muon over Stiefel Manifold

- Muon orthogonalizes the momentum
- Naturally, manifold muon should orthogonalize the weights

$$\min \ \mathcal{L}(\mathbf{W}) \quad \text{s.t.} \quad \mathbf{W} \in \mathrm{St}(d_{\mathrm{out}}, d_{\mathrm{in}}) = \{\mathbf{W} : \mathbf{W}^\top \mathbf{W} = I\}.$$

- Jeremy Bernstein (2024-2025), Franz Louis Cesista (2025), Jianlin Su (2025)
- Riemannian gradient descent method ($\mathbf{G} = \nabla \mathcal{L}(\mathbf{W})$):

$$\mathbf{A}^\star = \underset{\mathbf{A} \in T_{\mathbf{W}}\mathrm{St}(m,n)}{\operatorname{argmin}} \left\{ \langle \mathbf{G}, \mathbf{A} \rangle + \frac{1}{2\eta} \|\mathbf{A}\|_F^2 \right\}$$

$$\mathbf{W}^+ = \mathrm{Retr}_{\mathbf{W}}(\eta \mathbf{A}^\star)$$

- $T_{\mathbf{W}}\mathrm{St}(m,n) := \{\mathbf{A} \mid \mathbf{W}^\top \mathbf{A} + \mathbf{A}^\top \mathbf{W} = 0\}$ is the tangent space.
- Manifold Muon replaces Frobenius norm by spectral norm:

$$\min_{\mathbf{A}} \ \langle \mathbf{G}, \mathbf{A} \rangle, \ \text{s.t.}, \quad \|\mathbf{A}\|_{\mathrm{spec}} \le \eta, \quad \mathbf{A}^\top \mathbf{W} + \mathbf{W}^\top \mathbf{A} = 0. \quad \text{Not easy!}$$

J. Bernstein, Thinking Machines Lab, *Modular Manifolds*, September 2025.

Yang and Lai. Manifold constrained steepest descent for smooth and closed-set optimization. arXiv:2601.21487

Manifold Muon over Spectral Sphere

The **Spectral Sphere Optimizer (SSO)** restricts every hidden weight matrix to spectral sphere:

$$\mathcal{S}_R := \{\mathbf{W} : \|\mathbf{W}\|_{\text{spec}} = R\}.$$

SSO computes the spectral steepest-descent direction inside the tangent space:

$$\begin{aligned} \max_{\Phi} \quad & \langle \mathbf{M}_t, \Phi \rangle_F \\ \text{s.t.} \quad & \|\Phi\|_{\text{spec}} = 1, \\ & \langle \Theta_t, \Phi \rangle_F = 0. \end{aligned}$$

Here $\Theta_t = u_1 v_1^\top$, and u_1 and v_1 are the leading singular vectors of $\mathbf{W}_t$. Use λ to denote the Lagrange multiplier of $\langle \Theta_t, \Phi \rangle_F = 0$. This subproblem can be solved by bisection.

Xie et al. Controlled LLM Training on Spectral Sphere. arXiv:2601.08393

Spectral Sphere Optimizer

Tangent-space correction

Define

$$\widehat{\mathbf{M}}_t = \frac{\mathbf{M}_t}{\|\mathbf{M}_t\|_F}, \qquad h_t(\lambda) := \big\langle \Theta_t, \mathrm{msign}\big(\mathbf{M}_t + \lambda\Theta_t\big)\big\rangle_F .$$

Use the bisection method to find

$$\lambda_t^\star \qquad \text{such that} \qquad h_t(\lambda_t^\star) = 0.$$

Main SSO Update

1. **Tangent direction**

$$\Phi_t = \mathrm{msign}\big(\widehat{\mathbf{M}}_t + \lambda_t^\star \Theta_t\big).$$

2. **Trial step**

$$\widetilde{\mathbf{W}}_{t+1} = \mathbf{W}_t - \eta_t R \Phi_t.$$

3. **Spectral retraction**

$$\boxed{\mathbf{W}_{t+1} = \mathrm{Retr}_{\mathcal{S}_R}\big(\widetilde{\mathbf{W}}_{t+1}\big) := R\,\widetilde{\mathbf{W}}_{t+1}/\sigma_1(\widetilde{\mathbf{W}}_{t+1}).}$$

MuonH: Muon over Hyperball

Define

$$\mathbf{U}_t = \mathrm{NS}_5(\mathbf{M}_t) \approx \mathrm{msign}(\mathbf{M}_t), \qquad \mathrm{Normalize}(\mathbf{X}) := \frac{\mathbf{X}}{\|\mathbf{X}\|_F}, \qquad R := \|\mathbf{W}_0\|_F.$$

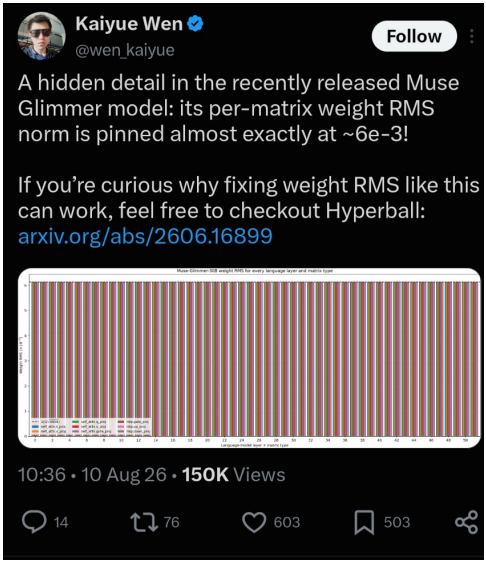
MuonH Update

$$\boxed{\mathbf{W}_{t+1} = R \cdot \mathrm{Normalize}\left(\mathbf{W}_t - \eta_t R \cdot \mathrm{Normalize}(\mathbf{U}_t)\right)}$$

- Muon determines the update direction $\mathbf{U}_t$.
- The update is normalized before taking the step.
- The new weight is normalized back to the sphere $\|\mathbf{W}_{t+1}\|_F = R$.

Source: Wen, Dang, Lyu, Ma, and Liang, *Fantastic Pretraining Optimizers and Where to Find Them II: Hyperball Optimization*, Dec 2025 (blog); arXiv:2606.16899, 2026.

Idea of Manifold Muon was used in Meta's Muse-Glimmer-30B



Muse Glimmer observation

Across language layers and matrix types, the per-matrix weight RMS is concentrated near

$$\|\mathbf{W}\|_F \approx 6 \times 10^{-3} \sqrt{d_{\text{out}}\, d_{\text{in}}}.$$

This corresponds to the hypersphere or Frobenius sphere: $\|\mathbf{W}\|_F = R$, a smooth manifold.

Takeaway: fixed-norm weight geometry is becoming relevant in frontier-scale model training, making its interaction with optimization and normalization practically important.

Source: Meta Superintelligence Lab, *Muse Glimmer-30B Model Card*, Meta, Aug 2026.

MuonH Is Used to Train Marin 535B-A23B



The public Marin training configuration confirms that **MuonH is used for the 535B-A23B hero run**.

Sources: Percy Liang, X, August 21, 2026; Marin public training configuration on github: `marin/experiments/grug/moe_hero_ep/launch_scaling_ladder.py`

Issues in Existing Works

Issues:

- Manifold Muon over Stiefel manifold doesn't scale
- Spectral sphere optimizer is too slow
- MuonH doesn't follow the Riemannian optimization principle and has no convergence guarantee

Questions:

- What's the difference between spectral sphere and Frobenius sphere?
- What roles does manifold constraints play?
- Can we design provably convergent algorithm based on Riemannian optimization theory?

Our Algorithm: Msign-Aligned Constrained Riemannian Optimizer (MACRO)

- MACRO: a combination of Riemannian SGD and Muon for solving

$$\min \ \mathcal{L}(\mathbf{W}) \quad \text{s.t.} \quad \mathbf{W} \in \mathcal{M}.$$

- Main iterates of MACRO:

$$\begin{aligned}
G_t &= \nabla \ell(\mathbf{W}_t, \xi_t), && \text{Euclidean gradient} \\
M_t &= \beta M_{t-1} + (1-\beta) G_t, && \text{Momentum} \\
\Phi_t &= \mathrm{proj}_{T_{\mathbf{W}_t}\mathcal{M}}(M_t), && \text{Riemannian gradient} \\
O_t &= \mathrm{msign}(\Phi_t), && \text{spectral steepest descent} \\
\widetilde{\nabla}_t &= R \, \frac{O_t}{\|O_t\|_{\mathcal{M}} + \epsilon}, && \text{scale normalization} \\
\mathbf{W}_{t+1} &= \mathcal{P}_{\mathcal{M}}\left(\mathbf{W}_t - \eta_t \widetilde{\nabla}_t\right). && \text{retraction}
\end{aligned}$$

Convergence Guarantee of MACRO

Assumptions.

- $\mathcal{M}$ is a compact C^3 manifold.
- ℓ is L-smooth
- $\mathbb{E}_\xi \nabla \ell(\mathbf{W}, \xi) = \nabla \mathcal{L}(\mathbf{W})$, $\mathbb{E}_\xi \|\nabla \ell(\mathbf{W}, \xi) - \nabla \mathcal{L}(\mathbf{W})\|_F^2 \le \sigma^2$.

Theorem

Denote $\Delta = \mathcal{L}(\mathbf{W}_0) - \mathcal{L}^*$. *By setting* $\beta = 1 - 1/\sqrt{T}$, $\eta_t = \eta = \Omega(\sqrt{\Delta/(LT^{3/2})})$, *the sequence* $\{\mathbf{W}_t\}$ *generated by MACRO satisfies*

$$\min_{t=0,\ldots,T-1} \mathbb{E}\|\nabla_{\mathcal{M}} \mathcal{L}(\mathbf{W}_t)\|_* \le \mathcal{O}\left(\sqrt{L\Delta}\sigma T^{-1/4}\right).$$

Matches the $O(T^{-1/4})$ rate for general nonconvex smooth stochastic optimization.

Outline

Muon

Manifold Muon

Comparison of Manifolds

RMSNorm

Weight Decay

Numerical Experiments

Conclusions

Which Manifold?

$$\min \ \ell(\mathbf{W}) \quad \text{s.t.} \quad \mathbf{W} \in \mathcal{M}.$$

1. Spectral sphere
$$\mathcal{M}_S(R) = \{\mathbf{W} : \|\mathbf{W}\|_{\text{spec}} = R\}$$
2. Frobenius sphere (hypersphere / hyperball)
$$\mathcal{M}_F(R) = \{\mathbf{W} : \|\mathbf{W}\|_F = R\}$$

Effects of Manifold Constraints: Activation Scale Control

Consider a single linear layer for a sequence of length T (# of tokens in one input sequence):

$$\mathbf{Y} = \mathbf{X}\mathbf{W}^\top, \qquad \mathbf{X} \in \mathbb{R}^{T \times d_{\text{in}}}, \quad \mathbf{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}.$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_T^\top \end{bmatrix} \in \mathbb{R}^{T \times d_{\text{in}}}, \qquad \mathbf{Y} = \mathbf{X}\mathbf{W}^\top = \begin{bmatrix} (\mathbf{W}\mathbf{x}_1)^\top \\ (\mathbf{W}\mathbf{x}_2)^\top \\ \vdots \\ (\mathbf{W}\mathbf{x}_T)^\top \end{bmatrix} \in \mathbb{R}^{T \times d_{\text{out}}},$$

where $\mathbf{x}_t \in \mathbb{R}^{d_{\text{in}}}$ is the activation vector for the t-th token.

We then have

$$\text{vec}(\mathbf{Y}) \;=\; (\mathbf{W} \otimes I_T)\, \text{vec}(\mathbf{X}).$$

Effects of Manifold Constraints: Activation Scale Control

We want the forward pass to maintain a constant RMS norm:

$$\|\operatorname{vec}(\mathbf{Y})\|_{\mathrm{RMS}} = \|\operatorname{vec}(\mathbf{X})\|_{\mathrm{RMS}} \;=\; \Theta(1),$$

where RMS norm is defined as

$$\|z\|_{\mathrm{RMS}} := \frac{\|z\|_2}{\sqrt{d}}, \text{ for } z \in \mathbb{R}^d$$

Easy to show:

$$\left\|\mathbf{W} \otimes I_T\right\|_{\mathrm{spec}} \;=\; \|\mathbf{W}\|_{\mathrm{spec}}, \qquad \left\|\mathbf{W} \otimes I_T\right\|_F \;=\; \sqrt{T}\,\|\mathbf{W}\|_F.$$

Therefore, the two constraints lead to different activation-control mechanisms

Effects of Manifold Constraints: Activation Scale Control

Assump. The weight matrix $\mathbf{W}$ is full rank with probability 1 throughout training.

This assumption is well supported in modern LLM pre-training:

- (i) at initialization, standard Gaussian schemes produce a full-rank matrix a.s.
- (ii) during training, the next-token prediction objective penalizes representational collapse, since a low-rank $\mathbf{W}$ would force a degenerate output distribution and an immediate spike in the training loss

Theorem. If

$$\|\mathbf{W}\|_{\text{spec}} = \Theta\left(\sqrt{\frac{d_{\text{out}}}{d_{\text{in}}}}\right),$$

then the RMS norm of activation is preserved as $\Theta(1)$.

Proof. From $\|\mathbf{Y}\|_F \leq \|\mathbf{W}\|_{\text{spec}}\|\mathbf{X}\|_F$, we get

$$\|\operatorname{vec}(\mathbf{Y})\|_{\text{RMS}} \leq \sqrt{\frac{d_{\text{in}}}{d_{\text{out}}}}\|\mathbf{W}\|_{\text{spec}}\|\operatorname{vec}(\mathbf{X})\|_{\text{RMS}}$$

The lower bound can be provded easily using the assumption.

The Frobenius sphere radius is larger

For Frobenius norm, the inequality

$$\|\mathbf{X}\mathbf{W}^\top\|_F \le \|\mathbf{X}\|_F \|\mathbf{W}\|_F$$

is too loose.

Average-case Analysis at Initialization

At initialization, the rows x_i of $\mathbf{X}$ are well modeled as zero-mean isotropic random vectors satisfying

$$\mathbb{E}[x_i x_i^\top] = \sigma_x^2 I_{D_{\text{in}}}.$$

A direct expansion gives

$$\begin{aligned}
\mathbb{E}\big[\|\mathbf{Y}\|_F^2\big] &= \mathbb{E}\Big[\text{Trace}\left(\mathbf{X}\mathbf{W}^\top\mathbf{W}\mathbf{X}^\top\right)\Big] \\
&= \text{Trace}\left(\sum_{i=1}^{T} \mathbb{E}[x_i x_i^\top]\,\mathbf{W}^\top\mathbf{W}\right) \\
&= T\sigma_x^2\|\mathbf{W}\|_F^2.
\end{aligned}$$

The Frobenius sphere radius is larger

Hence,

$$\left(\mathbb{E}\left[\|\operatorname{vec}(\mathbf{Y})\|_{\mathrm{RMS}}^2\right]\right)^{1/2} = \sqrt{\frac{\mathbb{E}\left[\|\mathbf{Y}\|_F^2\right]}{T d_{\mathrm{out}}}} = \frac{\sigma_x}{\sqrt{d_{\mathrm{out}}}}\|\mathbf{W}\|_F.$$

Therefore, we should require

$$\boxed{\|\mathbf{W}\|_F = \Theta\left(\frac{\sqrt{d_{\mathrm{out}}}}{\sigma_x}\right) = \Theta\left(\sqrt{d_{\mathrm{out}}}\right)} \qquad \text{when } \sigma_x = \Theta(1).$$

The Frobenius sphere radius is larger

Beyond Initialization: Covariance and Stable Rank.

Assump.

There exists a constant $\kappa = \Theta(1)$ such that

$$\|\mathbf{W}\|_{\text{spec}}^2 \;\le\; \frac{\kappa}{\min\{d_{\text{in}}, d_{\text{out}}\}}\,\|\mathbf{W}\|_F^2,$$

i.e., the stable rank $\mathrm{srk}(\mathbf{W}) := \|\mathbf{W}\|_F^2/\|\mathbf{W}\|_{\text{spec}}^2$ satisfies $\mathrm{srk}(\mathbf{W}) \ge \min\{d_{\text{in}}, d_{\text{out}}\}/\kappa$.

Theorem. If

$$\|\mathbf{W}\|_F \;=\; \Theta\Big(\sqrt{d_{\text{out}}}\Big),$$

then $\|\operatorname{vec}(\mathbf{Y})\|_{\text{RMS}} = \Theta(1)$.

Summary of Practical Constraint Radii

We introduce a tunable hyperparameter $r = \Theta(1)$.

Manifold	**Radius Setting**
Spectral Sphere	$r\sqrt{d_{\text{out}}/d_{\text{in}}}$
Frobenius Sphere	$r\sqrt{d_{\text{out}}}$

Later in the numerical experiments, we will show that our MACRO algorithm is not sensitive to this parameter little r.

Outline

Muon

Manifold Muon

Comparison of Manifolds

RMSNorm

Weight Decay

Numerical Experiments

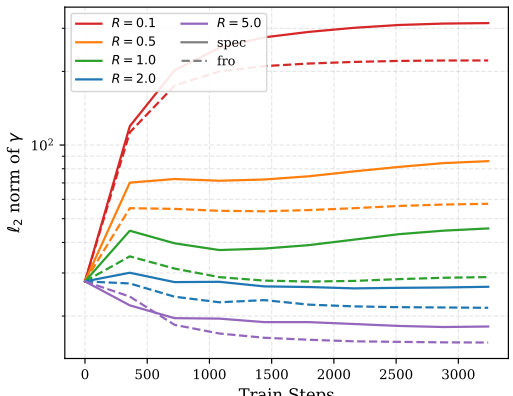
Conclusions

Manifold constraints vs. normalization layers

- In standard LLM architectures, normalization layers explicitly control the forward activation scale
- The RMSNorm layer uses a learnable affine parameter, γ, to scale activations

$$\text{RMSNorm}_\gamma(x) = \gamma \odot \frac{x}{\text{RMS}(x)}$$

The Model Compensates: $R \uparrow,\ \|\gamma\|_2 \downarrow$



Empirical observation

In the standard architecture, increasing the constraint radius is accompanied by a smaller learned RMSNorm affine scale.

Interpretation: the two mechanisms are functionally overlapping in activation-scale control.

Remove learnable RMS normalization

We remove all standard learnable Pre-RMSNorm layers,

$$\mathrm{RMSNorm}_{\gamma}(\mathbf{X}) = \gamma \odot \frac{\mathbf{X}}{\|\mathbf{X}\|_{\mathrm{RMS}}},$$

including the normalizations preceding the attention and MLP sublayers.

To handle architectural nonlinearities, we retain only targeted **parameter-free** normalizations:

- after the embedding and attention blocks;
- QK-Norm for controlling pre-softmax logits;
- Norm-Gated SwiGLU for controlling the multiplicative activation:

$$\mathbf{Y} = \left(\mathrm{Swish}(\mathbf{X}\mathbf{W}_1^\top)/\|\,\mathrm{Swish}(\mathbf{X}\mathbf{W}_1^\top)\|_{\mathrm{RMS}}\right) \odot (\mathbf{X}\mathbf{W}_2^\top).$$

Manifold constraint vs. normalization layers

Validation loss for QWEN3-like 330M pre-training without learnable normalization 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

Interplay with RMSNorm

1. In the standard architecture, both the manifold constraint and trainable RMSNorm control activation scale.
2. RMSNorm adapts when the constraint radius changes.
3. We remove the trainable RMSNorm layers and ask whether manifold constraint alone can stabilize the training.

Outline

Muon

Manifold Muon

Comparison of Manifolds

RMSNorm

Weight Decay

Numerical Experiments

Conclusions

Weight Decay as Geometric Regulation

Decoupled weight-decay update

$$\mathbf{W}_{t+1} = (1 - \eta_t \lambda)\mathbf{W}_t - \eta_t \mathbf{D}_t$$

Effect on weight dynamics

1. **Controls weight scale**: The shrinkage factor $1 - \eta_t \lambda$ discourages excessive growth of $\|\mathbf{W}_t\|_F$.
2. **Controls the relative update**:

$$\eta_{\mathrm{rel},t} := \frac{\|\Delta \mathbf{W}_t\|_F}{\|\mathbf{W}_t\|_F}.$$

   At scale equilibrium, weight decay keeps this ratio approximately controlled.

3. **Induces rotational equilibrium**: Late stage of training,

$$\mathbf{W}_{t+1} \approx \mathrm{rotation}(\mathbf{W}_t).$$

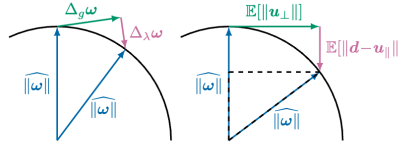


Figure 2: Two views of equilibrium where the weight norm $\|\omega\|$ is preserved because the gradient and weight decay components balance out on average. **Left:** Standard optimizer update eq. (7). **Right:** The total update contributions over the course of training, $\boldsymbol{u}$ and $\boldsymbol{d}$, derived from the gradient and weight decay of a given timestep, respectfully.

Weight decay achieves these geometric properties only heuristically.

Source: Kosson, Messmer, and Jaggi, *Rotational Equilibrium: How Weight Decay Balances Learning Across Neural Networks*, 2023.

Manifold constraints offer a principled alternative

MACRO constructs

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

Ignore ϵ, we have

$$\|\mathbf{W}_t\|_{\mathcal{M}} = R, \qquad \|\widetilde{\nabla}_t\|_{\mathcal{M}} = R,$$

hence the relative step ratio is

$$\eta_{\mathrm{rel,t}} := \frac{\|\eta_t \widetilde{\nabla}_t\|_{\mathcal{M}}}{\|\mathbf{W}_t\|_{\mathcal{M}}} = \eta_t,$$

which is controlled, because η_t is controlled (usually decaying).
The rotational equilibrium

$$\mathbf{W}_{t+1} \approx \mathrm{rotation}(\mathbf{W}_t)$$

is naturally achieved throughout the training because $\|\mathbf{W}_{t+1}\| = \|\mathbf{W}_t\| = R$.

Outline

Muon

Manifold Muon

Comparison of Manifolds

RMSNorm

Weight Decay

Numerical Experiments

Conclusions

Qwen3-like Models

A dense, decoder-only causal Transformer trained from scratch, following the main architectural design of Qwen3.

Attention	Grouped-Query Attention (GQA)
Position encoding	Rotary Positional Embeddings (RoPE)
Feed-forward block	SwiGLU gated MLP
Normalization	Pre-RMSNorm before attention and MLP
Training objective	Causal next-token prediction

Qwen3-like $\neq$ an official Qwen3 checkpoint

The models do not use pretrained Qwen3 weights or reproduce the complete Qwen3 tokenizer, dataset, context length, and training recipe. Their width and depth are scaled to produce our 240M, 600M, and 1.2B experimental models.

Standard Pre-training Results

QWEN3-like models: training loss and validation perplexity (**lower is better**).

Optimizer	**240M**		**600M**		**1.2B**	
	Train	Val. PPL	Train	Val. PPL	Train	Val. PPL
Muon	3.042	20.72 ± 0.010	2.745	15.37 ± 0.006	2.577	12.99 ± 0.004
MuonH-fro	3.033	20.55 ± 0.007	**2.735**	$\mathbf{15.23 \pm 0.006}$	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	3.034	20.58 ± 0.011	2.739	15.29 ± 0.002	**2.563**	$\mathbf{12.85 \pm 0.006}$
MACRO-spec	**3.031**	$\mathbf{20.50 \pm 0.011}$	2.740	15.30 ± 0.005	2.568	12.90 ± 0.009

- Manifold constraints improve validation loss
- Spectral and Frobenius sphere perform similarly
- MACRO and MuonH perform similarly

Learning-Rate Warmup

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	**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	**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

Warmup is the fraction of total training steps used for learning-rate ramp-up. Bold indicates the best warmup ratio for each optimizer.

Joint learning-rate and weight-decay sweep

Table: Muon learning-rate–weight-decay sweep on the 240M QWEN3-like model.

Optimizer	LR	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 (WD = 0)			
		Val. PPL	20.579 (WD = 0)			
MACRO-spec	0.008	Train	**3.0308** (WD = 0)			
		Val. PPL	**20.502** (WD = 0)			

Radius Sensitivity

Final validation perplexity; lower is better. Bold indicates the best radius for each learning rate.

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

Optimizer Efficiency

Optimizer	E2E (s/step) ↓	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%

Takeaway

MACRO-fro incurs only about 1.6% additional E2E time relative to Muon, while SSO and FSO are substantially more expensive.

Low-Precision Master Weights

Last-5% training loss and final validation perplexity; lower is better.

Model	Optimizer	Training Loss			Validation PPL	
		FP32	BF16	Δ_{tr}	FP32	BF16
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

$\Delta_{\mathrm{tr}} = L_{\mathrm{train}}^{\mathrm{FP32}} - L_{\mathrm{train}}^{\mathrm{BF16}}$.
A more negative value indicates greater degradation under BF16 master weights. All experiments use seed 1.

FP32 uses `min_lr_factor`= 0.01. For the 240M BF16 experiments, MACRO-fro and MACRO-spec use their best observed value, 0.05; all other BF16 experiments use 0.01.

Outline

Muon

Manifold Muon

Comparison of Manifolds

RMSNorm

Weight Decay

Numerical Experiments

Conclusions

Conclusions

Muon
Controls the scale of the **update direction**

Manifold constraint
Controls the scale of the **weights**.

RMSNorm
Functionally overlaps with manifold constraints.

Weight decay
is not needed when manifold constraints are imposed.

- MACRO algorithm combines Riemannian SGD and Muon
- MACRO is scalable, stable, and provably convergent
- MACRO is comparable with MuonH
- Ideas of manifold muon have already been by Meta and Marin
- RMSNorm (architecture) overlaps with manifold constraints (optimizer)
- Weight decay is not needed when manifold constraints are imposed

Thank You!