RODE: A RADIAL-ORTHOGONAL DECOUPLED ENGINE FOR OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Matrix-aware optimizers such as Muon exploit the structure of weight matrices to improve neural network training. However, their additive updates couple changes in weight norm and direction: a directional step can also increase the weight norm, and a larger norm reduces the angular effect of subsequent steps of the same size. This coupling makes it difficult to control weight scale and directional learning independently. We introduce RODE, a Radial-Orthogonal Decoupled Engine that separates each weight matrix into a scalar norm and a unit-norm direction. This separation is nontrivial for Muon-style optimization: as the direction changes, accumulated momentum may no longer lie in the current tangent space, while Newton–Schulz conditioning does not preserve tangency. RODE controls the norm with a scalar radial update and optimizes the direction through tangent-projected Newton–Schulz conditioning and a spherical update, using separate learning rates for the two components. This design allows directional learning without unintended norm growth from additive stepping. Compared with both evaluated Muon variants, RODE achieves lower mean validation loss in GPT-2 and Qwen2-style 114M language modeling and higher mean accuracy on CIFAR-100 and ImageNet-1K using learning rates transferred from GPT-2. RODE also achieves lower validation loss than Muon RMS in Qwen2-style 1.5B pretraining from scratch, and the highest mean accuracy on all four downstream tasks after Qwen3.5-9B fine-tuning.

1 INTRODUCTION

An optimizer determines how gradients are used to update model parameters, affecting both training efficiency and the performance of the trained model. AdamW uses past gradients to adjust the update size for each parameter (Kingma & Ba, 2015; Loshchilov & Hutter, 2019). Muon instead approximately orthogonalizes the momentum matrix through Newton–Schulz iterations and uses the result to update the weight matrix. This approach has shown strong performance in neural network training (Jordan et al., 2024; Liu et al., 2025a).

However, Muon still applies the resulting update by addition. A step that changes the weight's direction can therefore also increase its Frobenius norm. Once the Frobenius norm grows, an update of the same absolute size produces a smaller change in direction. The effective directional learning rate can thus decrease even when the prescribed step size stays fixed. This is the problem we address: learning the direction also changes the Frobenius norm, and that Frobenius norm change feeds back into how quickly the direction can move.

To see why this happens, separate a weight matrix into its Frobenius norm and direction:

$$W_t = \rho_t U_t, \qquad \rho_t = \|W_t\|_{\mathrm{F}}, \qquad \|U_t\|_{\mathrm{F}} = 1. \tag{1}$$

Here, ρ_t is the weight's Frobenius norm and U_t is its direction with Frobenius norm one. For an additive update $W_{t+1} = W_t - \eta Q_t$,

$$\|W_t - \eta Q_t\|_{\mathrm{F}}^2 = \rho_t^2 - 2\eta\rho_t\langle U_t, Q_t\rangle_{\mathrm{F}} + \eta^2\|Q_t\|_{\mathrm{F}}^2. \tag{2}$$

If Q_t is nonzero and perpendicular to U_t, the middle term vanishes, but the last term is positive. The Frobenius norm therefore grows even when the update has no radial component. For small steps, the angular change is $\eta\|Q_t^{\perp}\|_{\mathrm{F}}/\rho_t + \mathcal{O}(\eta^2)$, where $Q_t^{\perp}$ is the component perpendicular to U_t. The factor $1/\rho_t$ explains why a larger Frobenius norm reduces the angular effect of a given tangent update.

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.

Weight decay can counteract Frobenius norm growth, but the resulting Frobenius norm depends on the balance between weight decay and the optimizer's updates (Kosson et al., 2024; Wang & Aitchison, 2025). It therefore controls the Frobenius norm indirectly. This matters even in scale-invariant layers: changing the Frobenius norm may leave the layer's function unchanged while still changing its effective directional learning rate (van Laarhoven, 2017; Hoffer et al., 2018; Heo et al., 2021; Kosson et al., 2024). These observations motivate controlling the Frobenius norm and direction with separate update rules and learning rates.

We propose **RODE**, a **R**adial-**O**rthogonal **D**ecoupled **E**ngine. RODE updates the scalar Frobenius norm ρ_t using the radial gradient. To update the direction U_t, it projects momentum onto the tangent space, applies Newton–Schulz iterations with tangent reprojection, and moves along the unit sphere. This spherical step changes U_t while preserving its unit Frobenius norm, so the Frobenius norm of the reconstructed weight is set by the radial update. Separate learning rates control the radial and directional steps. Setting the radial learning rate to zero also allows the weight's Frobenius norm to remain fixed. RODE thus retains Muon's use of Newton–Schulz iterations while giving Frobenius norm and direction separate controls.

Our contributions are:

- RODE, a matrix optimizer that updates the weight's Frobenius norm and direction separately, combining a scalar radial update with tangent-projected Newton–Schulz iterations and a spherical directional step.
- Controlled experiments and ablations that show the benefits of Frobenius norm control and the directional update, together with a convergence-neighborhood analysis under stated regularity and alignment assumptions (Appendix D). For a first-order approximation of the RODE update, under assumptions including radial and tangential gradient alignment and the Polyak–Łojasiewicz condition, we show that the loss approaches its optimum at a linear rate, up to an error determined by the step size (Appendix E).
- Evaluations on language modeling, image classification, and downstream tasks after language-model fine-tuning, showing improvements over the evaluated Muon variants and other baselines.

2 Related Work

From coordinate adaptation to matrix orthogonalization. While standard optimizers such as Adam (Kingma & Ba, 2015) and AdamW (Loshchilov & Hutter, 2019) adjust updates coordinate-wise, recent matrix-aware methods exploit parameter geometry more directly. Shampoo (Gupta et al., 2018) and SOAP (Vyas et al., 2025) precondition updates via matrix covariances and their eigenspaces, whereas Muon (Jordan et al., 2024) and MARS-M (Yuan et al., 2025; Liu et al., 2025b) employ Newton–Schulz iterations to orthogonalize momentum. Despite these structural gains, these optimizers apply updates additively. Consequently, any step that alters the weight direction can inadvertently perturb its Frobenius norm, which subsequently dampens the effective angular change of future updates.

Scale and direction in optimization. The tension between parameter magnitude and directional movement has inspired extensive research. Architectural solutions like Weight Normalization (Salimans & Kingma, 2016) decouple scale from direction, while AdamP (Heo et al., 2021) removes norm-increasing radial momentum during optimization. Rotational-equilibrium studies (Kosson et al., 2024; Roburin et al., 2022) and norm-scaled methods (e.g., LARS/LAMB, Fromage, Nero) (You et al., 2017; 2020; Bernstein et al., 2020; Liu et al., 2021) explicitly regulate relative or angular step sizes. Complementing these heuristic controls, Riemannian optimization (Absil et al., 2008; Bonnabel, 2013; Bécigneul & Ganea, 2019) provides principled tools for updating parameters along manifold tangent spaces. However, these techniques are predominantly formulated for vector or coordinate-level updates rather than matrix orthogonalization.

Radial–directional matrix methods. Closest to our motivation are recent efforts that bring magnitude–direction separation to weight matrices. Hyperball (Wen et al., 2026) normalizes updates from a base optimizer and projects the result back to the initial Frobenius-norm sphere, while

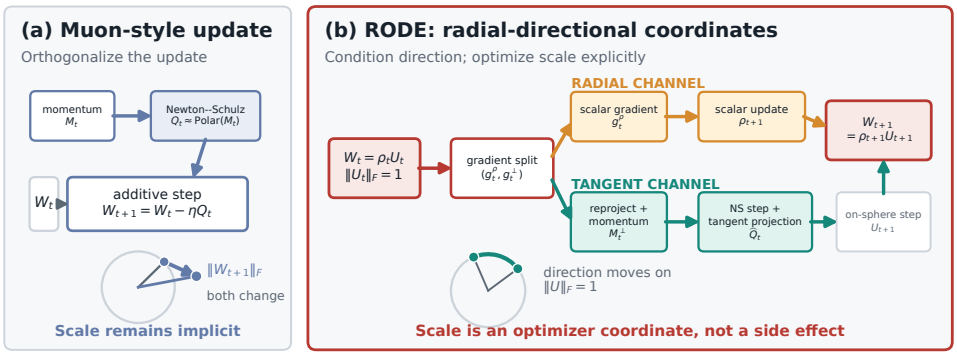

Figure 1: Muon conditions momentum and adds the resulting matrix step directly to W_t, so its effects on norm and direction are jointly determined. RODE exposes a scalar radial signal and a tangent-projected directional signal, assigns them separate step sizes, and reconstructs the managed matrix from the resulting ρ_{t+1} and U_{t+1}.

Magnitude–Direction Decoupling (Hägele et al., 2026) maintains a unit-norm directional factor alongside learnable row and column gains. RODE takes a different approach: rather than relying on post-hoc projection or extra gain parameters, it embeds tangent projection directly into each Newton–Schulz iteration before taking a spherical step, while governing matrix scale through an explicit, independently tuned scalar radial update.

3 Method

RODE decomposes each managed weight matrix as $W_t = \rho_t U_t$, where $\rho_t = \|W_t\|_{\mathrm{F}}$ is its Frobenius norm and $U_t = W_t/\rho_t$ is its direction with $\|U_t\|_{\mathrm{F}} = 1$. As shown in Figure 1, RODE updates the matrix norm and direction through two separately controlled channels: a scalar radial channel for ρ_t and a matrix directional channel for U_t. Below, we describe these channels for a single matrix; parameter routing across model layers is specified in Section 5.1.

3.1 Radial Control and Tangent Momentum

To separate norm from direction, we first decompose the stochastic gradient $g_t \approx \nabla f(W_t)$. Using the Frobenius inner product $\langle A, B\rangle_{\mathrm{F}} = \mathrm{tr}(A^\top B)$, we define the tangent projection $\mathcal{P}_t$, which removes the component aligned with U_t:

$$\mathcal{P}_t(X) = X - \langle X, U_t\rangle_{\mathrm{F}} U_t, \qquad g_t^\rho = \langle g_t, U_t\rangle_{\mathrm{F}}, \qquad g_t^\perp = \mathcal{P}_t(g_t). \tag{3}$$

Here, g_t^ρ is the scalar radial gradient that changes the matrix norm, while $g_t^\perp$ is the tangent component orthogonal to U_t that changes its direction.

As U_t rotates, accumulating historical momentum without projection would mix radial and tangent signals. RODE therefore projects the previous raw momentum buffer B_{t-1} onto the current tangent space before adding the new gradient:

$$\begin{aligned} \widetilde{B}_t &= \beta \mathcal{P}_t(B_{t-1}) + g_t, & s_t &= \langle \widetilde{B}_t, U_t\rangle_{\mathrm{F}} = g_t^\rho, \\ \rho_{t+1} &= \max\{\rho_t - \eta_\rho s_t, \kappa_{\mathrm{floor}}\rho_t\}, & M_t^\perp &= \mathcal{P}_t(\widetilde{B}_t). \end{aligned} \tag{4}$$

Because historical momentum is projected onto the tangent space, only the current radial gradient g_t^ρ drives the norm update. The separate radial learning rate η_ρ controls this update, and the positive lower bound $\kappa_{\mathrm{floor}}\rho_t$ prevents the radius from crossing zero. Setting $\eta_\rho = 0$ keeps the norm fixed. We store $B_t = \widetilde{B}_t$ for the next step. The stored radial component can contribute to future tangent updates as U_t rotates; Appendix D bounds this contribution.

Under $W = \rho U$, the directional-coordinate gradient is $\rho_t g_t^{\perp}$. RODE omits the explicit factor ρ_t when forming its directional momentum. This avoids explicitly scaling the directional update by the current norm; the gradient and momentum history may still depend on the weight scale.

3.2 Conditioning and spherical realization

Next, RODE conditions the tangent momentum $M_t^{\perp}$ and updates the direction along the unit sphere. For efficient Newton–Schulz iterations, if $M_t^{\perp}$ has more columns than rows, we transpose both $M_t^{\perp}$ and U_t and use the corresponding tangent projector in the transposed coordinates. For nonzero $M_t^{\perp}$, initialize $X_0 = M_t^{\perp}/\|M_t^{\perp}\|_{\mathrm{F}}$. For $k = 0, \ldots, K-1$, compute

$$\begin{aligned} A_k &= X_k^{\top} X_k, \\ X_{k+1} &= \mathcal{P}_t\big[3.4445 X_k + X_k\{-4.7750 A_k + 2.0315 A_k^2\}\big]. \end{aligned} \tag{5}$$

The Newton–Schulz polynomial approximately orthogonalizes the momentum matrix (Higham, 2008). Since this operation does not generally preserve tangency, RODE reapplies $\mathcal{P}_t$ after every iteration to keep the conditioned matrix orthogonal to U_t.

After undoing any transpose, let $\widehat{Q}_t = X_K$ be the final conditioned matrix and $q_t = \|\widehat{Q}_t\|_{\mathrm{F}}$ its norm. When $q_t > 0$, normalize it to a unit update direction and take a spherical step:

$$\begin{aligned} D_t^{\perp} &= \widehat{Q}_t / q_t, & \theta_t &= \eta_{\mathrm{dir}} q_t, \\ U_{t+1} &= U_t \cos\theta_t - D_t^{\perp} \sin\theta_t, & W_{t+1} &= \rho_{t+1} U_{t+1}. \end{aligned} \tag{6}$$

Because $D_t^{\perp}$ is orthogonal to U_t, this formula rotates U_t along the unit sphere by θ_t, with directional learning rate η_{dir}. In exact arithmetic, the rotation preserves $\|U_{t+1}\|_{\mathrm{F}} = 1$, so the directional step does not change the matrix norm.

In practice, we renormalize U_{t+1} to guard against floating-point drift and handle zero norms separately (Algorithm 1). By default, RODE uses momentum $\beta = 0.9$ and $K = 5$ Newton–Schulz steps without damping. The optional RODE-SNR variant multiplies θ_t by a bounded scalar damping factor based on the gradient signal-to-noise ratio (Appendix A.2).

4 Theoretical Guarantees

We analyze conditions under which RODE converges. The analysis considers steps where the norm update does not activate the lower-bound safeguard and the directional update is nonzero. Along with smoothness, boundedness, and local regularity assumptions, we require bounded rotation angles and positive alignment between the tangent momentum and the update produced by Newton–Schulz iterations with tangent projection. The precise assumptions are given in Appendix D.

Write $G_t^{\rho} = \langle G_t, U_t \rangle_{\mathrm{F}}$ and $G_t^{\perp} = \mathcal{P}_t(G_t)$ for the radial and tangent components of the full gradient. A displacement expansion separates radial descent, tangent descent, geodesic curvature, and cross-step momentum leakage. Combining objective value with tangent-momentum tracking error in a Lyapunov function yields

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[(G_t^{\rho})^2] \le \epsilon_{\rho}, \qquad \limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[\|G_t^{\perp}\|_{\mathrm{F}}^2] \le \epsilon_{\perp}. \tag{7}$$

Theorem 1 gives the dependence on step sizes, noise, and regularity constants. Appendix E analyzes a first-order approximation of the RODE update under its stated assumptions, including radial and tangential gradient alignment and the Polyak–Łojasiewicz condition (Karimi et al., 2016). It establishes linear convergence of the loss to an $\mathcal{O}(\eta)$ neighborhood of its optimal value.

5 Experiments

5.1 Setup

We evaluate standard GPT-2 and a randomly initialized Qwen2-style 114M decoder on WikiText-103 for 8000 and 4000 steps, respectively (Radford et al., 2019; Merity et al., 2017; Yang et al., 2024);

ResNet-50 on CIFAR-100 and ImageNet-1K (He et al., 2016; Krizhevsky, 2009; Deng et al., 2009); a Qwen2-style 1.5B decoder on FineWeb for 20000 steps (Penedo et al., 2024); and Qwen3.5-9B full-parameter fine-tuning (Qwen Team, 2026). The Qwen2-style models are locally instantiated architectures, not official checkpoints.

All optimization-quality comparisons report three-seed means and sample standard deviations. Within-task formal runs share model, data, horizon, seed set, and checkpoint rule. Rates are selected by Optuna or transferred as marked: vision uses GPT-2-selected rates, and 1.5B includes both transferred rates and equal-budget task-specific tuning. Appendix B details search spaces, selection criteria, and schedules.

Muon Original and Muon RMS use the same momentum and Newton–Schulz procedure but scale the update differently. For an $m \times n$ matrix and matrix learning rate η, their effective learning rates are

$$\eta_{\text{Original}} = \eta\sqrt{\max(1, m/n)}, \qquad \eta_{\text{RMS}} = 0.2\eta\sqrt{\max(m, n)}.$$

RODE works alongside AdamW to update model parameters. In language-model pretraining, RODE updates matrix parameters other than embeddings and the output head, while AdamW updates the remaining parameters. In vision tasks, multi-dimensional weights are reshaped into matrices and updated by RODE; the remaining parameters use AdamW. The 9B full-parameter fine-tuning experiments use a different assignment: embedding and output-head matrices are also updated by the matrix optimizer.

The reported full-model norm includes all trainable parameters, including those updated by AdamW. Thus, fixing the norms of RODE-managed matrices does not necessarily keep the full-model norm constant. Appendix B.3 gives the precise definition.

5.2 Optimization quality and norm behavior

Table 1: Cross-task optimization performance on language modeling and image classification benchmarks. Results are reported as mean ± sample standard deviation over three seeds. Vision reports best accuracy on the official CIFAR-100 test set and the ImageNet-1K validation set. ‡Rates transferred from GPT-2. Best available mean per column is bold.

Optimizer	GPT-2 loss ↓	Qwen2-style LM loss ↓	CIFAR acc. ↑	ImageNet acc. ↑
AdamW	3.1886 ± 0.0120	3.0503 ± 0.0071	$68.66 \pm 0.29^{\ddagger}$	$62.49 \pm 0.46^{\ddagger}$
Muon Original	3.0295 ± 0.0001	2.8185 ± 0.0065	$71.49 \pm 0.67^{\ddagger}$	$64.98 \pm 0.05^{\ddagger}$
Muon RMS	3.0265 ± 0.0027	3.0205 ± 0.0103	$71.11 \pm 0.14^{\ddagger}$	$64.48 \pm 0.14^{\ddagger}$
SGD	–	–	72.77 ± 0.24	70.25 ± 0.08
RODE	2.9774 ± 0.0011	2.7731 ± 0.0013	$\mathbf{73.75 \pm 0.29}^{\ddagger}$	$69.74 \pm 0.47^{\ddagger}$
RODE-SNR	2.9724 ± 0.0557	$\mathbf{2.7493 \pm 0.0011}$	$73.64 \pm 0.56^{\ddagger}$	$\mathbf{70.49 \pm 0.09}^{\ddagger}$
SOAP	$\mathbf{2.9567 \pm 0.0023}$	2.7684 ± 0.0008	$72.21 \pm 0.09^{\ddagger}$	–
MARS-M	2.9699 ± 0.0016	2.8885 ± 0.0040	$66.57 \pm 1.37^{\ddagger}$	–

RODE achieves lower mean validation loss on the two language-modeling tasks and higher mean accuracy on the two image-classification tasks than both Muon variants (Table 1). Other baselines lead on some tasks: SOAP and MARS-M have lower GPT-2 loss than either RODE variant, SOAP has lower Qwen2-style LM loss than default RODE, and SGD has higher ImageNet-1K accuracy than default RODE. Compared with Muon, the lower GPT-2 loss is accompanied by reduced norm growth (Figure 2). The vision experiments evaluate hyperparameter transfer. We further examine the contributions of norm control and directional updating.

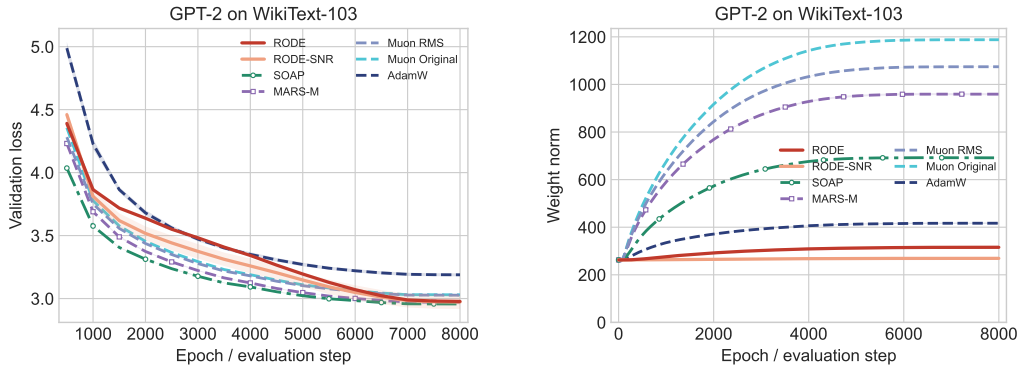

Figure 2: Standard GPT-2/WikiText-103 over 8000 steps (three-seed mean ± sample SD): validation loss (left) and observed full-model weight norm (right). The norm aggregates all trainable parameters, including those delegated to AdamW; both RODE variants show substantially less growth than the Muon runs.

5.3 What contributes to the improvement?

Table 2: Sequential GPT-2 norm–direction intervention. Three-seed values are mean ± sample SD; Δ is relative to Muon RMS. Muon and RODE use their respective selected learning rates.

Condition	Direction update	Radius policy	Val. loss ↓	Δ vs. Muon
Muon RMS	additive Muon	implicit	3.027 ± 0.003	–
Muon + NormProj	additive Muon	fixed by projection	3.008 ± 0.001	-0.019
RODE, fixed ρ	RODE tangent stack	fixed	2.977 ± 0.002	-0.050
Full RODE	RODE tangent stack	learned scalar	2.977 ± 0.001	-0.050

Norm control alone improves Muon, but still falls short of RODE (Table 2). Ablations show that Newton–Schulz conditioning is an important component of RODE, whereas fixing the radius or replacing the exact spherical rotation with normalized retraction has little effect on the results (Table 3).

Table 3: Three-seed 114M component study (mean ± sample SD). Structural ablations inherit RODE's learning rates without retuning; RODE-SNR is tuned independently. Δ is relative to full RODE.

Variant	Intervention	Final val. ↓	Δ
Full RODE	Full engine	2.7731 ± 0.0013	–
RODE-SNR	Add tangent-gradient ratio damping	2.7493 ± 0.0011	-0.0238
w/o Newton–Schulz	Remove projected NS conditioning	3.1249 ± 0.0113	$+0.3518$
w/o geodesic	Use normalized retraction	2.7736 ± 0.0016	$+0.0005$
RODE, radius fixed	Hold radius fixed	2.7745 ± 0.0017	$+0.0014$
Radial-only control	Hold direction fixed	3.8715 ± 0.0023	$+1.0984$

5.4 SCALING TO 1.5B PARAMETERS

Table 4: Task-specific tuning at 1.5B scale. All losses are measured on the document-disjoint held-out cache; values are three-seed mean ± sample SD.

Optimizer	Best held-out loss ↓	Final held-out loss ↓	Final global norm ↓
Muon RMS	3.09277 ± 0.00200	3.09278 ± 0.00199	2910.6 ± 2.2
RODE	3.02803 ± 0.00182	3.02807 ± 0.00182	1163.1 ± 2.8
RODE, fixed ρ	$\mathbf{3.01019 \pm 0.00099}$	$\mathbf{3.01021 \pm 0.00103}$	$\mathbf{950.1 \pm 2.2}$

With equal-budget task-specific tuning, RODE's advantage over Muon RMS extends to 1.5B parameters (Table 4). Fixed-radius RODE performs better still. Transferred learning rates yield the same ordering (Appendices C.4 and B.6.1).

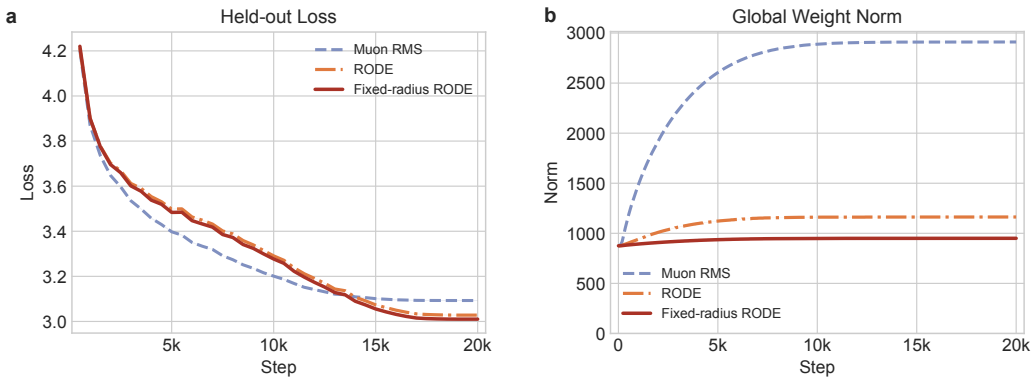

Figure 3: Qwen2-style 1.5B/FineWeb training after task-specific tuning (three-seed mean ± sample SD): unsmoothed held-out loss and observed global weight norm. Fixed-radius RODE freezes managed matrix radii; parameters delegated to AdamW can still change the global norm.

5.5 FULL-PARAMETER FINE-TUNING AT 9B SCALE

Table 5: Qwen3.5-9B full-parameter fine-tuning and task evaluation (three-seed mean ± sample SD). All task columns report accuracy in percent. Each optimizer is independently tuned, and downstream evaluation uses the minimum-validation-loss checkpoint from each run. The best mean per column among these five optimizers is bold.

Optimizer	Best val. loss ↓	GSM8K ↑	MATH-500 ↑	MMLU-STEM ↑	MMLU-Pro Math ↑
AdamW	0.3490 ± 0.0005	79.58 ± 0.52	44.67 ± 0.70	75.53 ± 0.36	40.07 ± 0.65
Muon Original	0.3497 ± 0.0005	84.23 ± 0.80	48.60 ± 3.47	75.77 ± 0.91	39.97 ± 1.35
Muon RMS	0.3455 ± 0.0005	84.03 ± 0.98	53.40 ± 1.51	76.24 ± 0.91	41.77 ± 1.98
SOAP	$\mathbf{0.3434 \pm 0.0011}$	81.68 ± 1.98	48.13 ± 0.83	76.22 ± 0.54	41.40 ± 1.04
RODE	0.3492 ± 0.0022	$\mathbf{85.47 \pm 0.46}$	$\mathbf{54.53 \pm 0.95}$	$\mathbf{76.75 \pm 1.14}$	$\mathbf{41.87 \pm 1.26}$

Among AdamW, Muon Original, Muon RMS, SOAP, and RODE, RODE achieves the highest mean accuracy on all four downstream tasks after Qwen3.5-9B full-parameter fine-tuning (Table 5). The validation-loss ranking differs: SOAP has the lowest mean best validation loss, and Muon RMS and AdamW also have lower values than RODE. These accuracy comparisons concern three-seed means; the MMLU-Pro Math margin over Muon RMS is only 0.10 percentage points, smaller than either method's sample standard deviation. All optimizers are independently tuned and evaluated using their minimum-validation-loss checkpoints (Appendix B).

5.6 COMPUTATION COST

RODE's gains come with a throughput cost relative to Muon RMS, but no additional matrix-sized optimizer state. Shape-batched fusion reduces this overhead, while optimizer-state storage remains below AdamW's. Table 6 reports the measurements, and Appendix A.3 describes the fused implementation.

Table 6: Compute and memory benchmark. Measurements use a single RTX 5090 D v2 with `bf16` autocast: the Qwen2-style LM uses micro-batch 16 and sequence length 512, while ResNet-50 uses batch 64 and 224×224 inputs. Each of three runs has 8 warm-up and 25 timed steps; Step and Opt. are medians of the three per-run medians. Peak is PyTorch peak allocated memory, State is exact optimizer-state tensor storage, Opt. includes `step()` and gradient clearing, and throughput is relative to AdamW.

Task	Optimizer	State (MB)	Peak (GB)	Step (ms)	Opt. (ms)	Throughput (vs. AdamW)
Qwen2-style LM	AdamW	913.0	10.92	102.0	9.0	$1.00\times$
	Muon Original	611.0	10.62	131.3	37.7	$0.78\times$
	Muon RMS	611.0	10.62	132.4	39.1	$0.77\times$
	SOAP	2643.2	12.67	137.3	42.1	$0.74\times$
	MARS-M	913.0	10.96	158.9	68.1	$0.64\times$
	RODE (loop)	611.0	10.62	172.3	81.2	$0.59\times$
	RODE (fused)	611.0	10.63	142.2	47.9	$0.72\times$
ResNet-50	AdamW	204.5	3.27	41.3	2.5	$1.00\times$
	Muon Original	102.4	3.26	59.0	20.2	$0.70\times$
	Muon RMS	102.4	3.26	58.9	20.2	$0.70\times$
	SOAP	630.3	3.71	74.0	34.7	$0.56\times$
	MARS-M	204.5	3.28	201.6	161.8	$0.20\times$
	RODE (loop)	102.4	3.17	89.5	50.4	$0.46\times$
	RODE (fused)	102.4	3.17	64.8	24.7	$0.64\times$

6 LIMITATIONS

In the compute benchmarks, RODE's training throughput remains below that of Muon RMS. The fused implementation narrows this gap without eliminating the overhead of the additional projections and spherical update.

RODE's performance gains vary across tasks and metrics. Our experiments use three random seeds; additional runs would help assess the robustness of smaller performance differences.

The vision experiments evaluate hyperparameter transfer from language modeling; performance after independent tuning on each vision task remains untested. Finally, the pretraining experiments use limited training horizons, leaving RODE's advantage over longer training runs to be verified.

7 CONCLUSION

Matrix optimizers such as Muon typically use additive matrix updates, which can cause the weight matrix's Frobenius norm to grow during training. This norm growth can reduce the effectiveness of subsequent directional updates. To address this, we propose RODE, which explicitly controls the matrix norm and updates the direction through preconditioning in the tangent space. Across the evaluated pretraining and image-classification tasks, RODE improves mean task performance over the compared Muon baselines. In the five-optimizer 9B comparison, it achieves the highest mean accuracy on all four downstream tasks, while other optimizers achieve lower validation losses. We also find that, under some experimental conditions, fixing the radius matches or outperforms learning it. These results show that decoupling the parameter matrix norm from orthogonalized directional updates provides a practical approach to matrix optimization.

AI USE STATEMENT

We used generative AI tools for language editing and LaTeX formatting of this manuscript. We did not use generative AI tools to generate research ideas, experimental results, or theoretical proofs. All technical claims, equations, and reported numbers were written and verified by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts.

ETHICS STATEMENT

This work studies optimization algorithms for training neural networks on publicly available language and vision datasets. It does not involve human subjects, private data collection, or the release of new datasets. We do not foresee immediate harmful applications beyond those already associated with training and fine-tuning large models.

REPRODUCIBILITY STATEMENT

The complete per-matrix update is given in Algorithm 1, and the fused implementation is described in Appendix A.3. Optimizer routing, search spaces, schedules, and selected hyperparameters are specified in Appendix B. Metric definitions and additional diagnostics appear in Appendix B.3 and Appendix C. Convergence assumptions and proofs are collected in Appendices D and E. All tabulated results use raw logs over three seeds unless otherwise stated.

REFERENCES

P.-A. Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

Gary Bécigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. In *International Conference on Learning Representations*, 2019.

Jeremy Bernstein, Arash Vahdat, Yisong Yue, and Ming-Yu Liu. On the distance between two neural networks and the stability of learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Silvère Bonnabel. Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013. doi: 10.1109/TAC.2013.2254619.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, volume 80, pp. 1842–1850, 2018.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021b. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

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. In *International Conference on Learning Representations*, 2021.

Nicholas J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, 2008. doi: 10.1137/1.9780898717778.

Elad Hoffer, Ron Banner, Itay Golan, and Daniel Soudry. Norm matters: Efficient and accurate normalization schemes in deep networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

HuggingFaceFW. FineWeb dataset card, 2025. URL https://huggingface.co/datasets/HuggingFaceFW/fineweb. Version 1.2.0; accessed July 31, 2026.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. https://kellerjordan.github.io/posts/muon/, 2024. Accessed August 17, 2026.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak–łojasiewicz condition. In *European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 795–811, 2016. doi: 10.1007/978-3-319-46128-1_50.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Atli Kosson, Bettina Messmer, and Martin Jaggi. Rotational equilibrium: How weight decay balances learning across neural networks. In *International Conference on Machine Learning*, volume 235, pp. 25333–25369, 2024.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *Technical Report*, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for LLM training. *arXiv preprint arXiv:2502.16982*, 2025a. URL https://arxiv.org/abs/2502.16982.

Yang Liu, Jeremy Bernstein, Markus Meister, and Yisong Yue. Learning by turning: Neural architecture aware optimisation. In *International Conference on Machine Learning*, volume 139, pp. 6748–6758, 2021.

Yifeng Liu, Angela Yuan, and Quanquan Gu. MARS-M: When variance reduction meets matrices, 2025b. URL https://arxiv.org/abs/2510.21800.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0970.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5. Accessed August 21, 2026.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Technical Report*, 2019. URL https://cdn.openai.com/better-language-models/language-models.pdf.

Simon Roburin, Yann de Mont-Marin, Andrei Bursuc, Renaud Marlet, Patrick Pérez, and Mathieu Aubry. Spherical perspective on learning with normalization layers. *Neurocomputing*, 487:66–74, 2022. doi: 10.1016/j.neucom.2022.02.021.

Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

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

Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham M. Kakade. SOAP: Improving and stabilizing Shampoo using Adam for language modeling. In *International Conference on Learning Representations*, 2025.

Xi Wang and Laurence Aitchison. How to set AdamW's weight decay as you scale model and dataset size. In *International Conference on Machine Learning*, volume 267, pp. 62222–62250, 2025.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3018. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/ad236edc564f3e3156e1b2feafb99a24-Abstract-Datasets_and_Benchmarks_Track.html.

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

An Yang et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. URL https://arxiv.org/abs/2407.10671.

Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks, 2017. URL https://arxiv.org/abs/1708.03888.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training BERT in 76 minutes. In *International Conference on Learning Representations*, 2020.

Huizhuo Yuan, Yifeng Liu, Shuang Wu, Xun Zhou, and Quanquan Gu. MARS: Unleashing the power of variance reduction for training large models. In *International Conference on Machine Learning*, volume 267, pp. 73553–73587, 2025.

Appendix A gives the algorithm and implementation details. Appendices B and C collect the experimental setup and additional results. Appendices D and E provide the analyses of convergence properties and convergence rates, respectively.

A Algorithm and implementation details

A.1 Default RODE update

Input: parameter W_t, gradient g_t, raw buffer B_{t-1}, learning rates $\eta_\rho, \eta_{\mathrm{dir}}$, momentum β, floor κ_{floor}, NS steps K, and safeguards $\epsilon, \epsilon_{\mathrm{NS}}$.

1. If $\|W_t\|_{\mathrm{F}} > \epsilon$, form $\rho_t = \|W_t\|_{\mathrm{F}}$, $U_t = W_t/\rho_t$, and $\mathcal{P}_t(X) = X - \langle X, U_t\rangle_{\mathrm{F}} U_t$.
2. Set $\widetilde{B}_t = \beta \mathcal{P}_t(B_{t-1}) + g_t$.
3. Set $s_t = \langle \widetilde{B}_t, U_t\rangle_{\mathrm{F}}$ and $\rho_{t+1} = \max\{\rho_t - \eta_\rho s_t, \kappa_{\mathrm{floor}}\rho_t\}$.
4. Set $M_t^\perp = \mathcal{P}_t(\widetilde{B}_t) = \widetilde{B}_t - s_t U_t$.
5. Transpose $M_t^\perp$ and U_t if needed so rows $\geq$ columns; initialize $X_0 = M_t^\perp/\|M_t^\perp\|_{\mathrm{F}}$ in the nonzero branch.
6. For $k = 0, \ldots, K-1$, set
$$X_{k+1} = \mathcal{P}_t\big[3.4445 X_k + X_k\{-4.7750 X_k^\top X_k + 2.0315 (X_k^\top X_k)^2\}\big] .$$
Undo any transpose and set $\widehat{Q}_t = X_K$.
7. For $q_t = \|\widehat{Q}_t\|_{\mathrm{F}} > 0$, set $D_t^\perp = \widehat{Q}_t/q_t$, $\theta_t = \eta_{\mathrm{dir}} q_t$, and $U_{t+1} = U_t \cos\theta_t - D_t^\perp \sin\theta_t$. Renormalize U_{t+1}.
8. Write $W_{t+1} = \rho_{t+1} U_{t+1}$ and store $B_t = \widetilde{B}_t$.

Algorithm 1: Default RODE update for one matrix parameter

Zero parameter norms and zero projected kernels take safeguarded no-direction branches. RODE-SNR changes θ_t to $\eta_{\mathrm{dir}}\gamma_t q_t$.

A.2 Optional RODE-SNR variant

RODE-SNR optionally damps the directional step using a running estimate of the tangent-gradient scale. We maintain a scalar state v_t that accumulates the recent magnitude of $g_t^\perp$:

$$v_t = \beta v_{t-1} + \|g_t^\perp\|_{\mathrm{F}} + \epsilon. \tag{8}$$

The damping factor compares the current tangent-momentum norm with this reference scale:

$$\gamma_t = \mathrm{clip}\left(\frac{\|M_t^\perp\|_{\mathrm{F}}}{v_t}, \gamma_{\min}, 1\right). \tag{9}$$

RODE-SNR then uses

$$\theta_t = \eta_{\mathrm{dir}}\gamma_t q_t, \tag{10}$$

so γ_t changes only the directional step magnitude and leaves $D_t^\perp$ unchanged. Default RODE sets $\gamma_t = 1$; we report RODE-SNR as an optional, separately tuned variant.

A.3 Batched implementation

A direct implementation of Algorithm 1 updates one matrix at a time in a Python loop. Each matrix requires several GPU kernel launches for norm computation, tangent projection, Newton–Schulz iterations, and the spherical update. We reduce this overhead by processing matrices of the same shape together.

Within each parameter group, we stack B matrices of shape $m \times n$ into a tensor $\mathcal{W} \in \mathbb{R}^{B\times m\times n}$, together with their gradients and momentum buffers. Norms and inner products are computed over the last two dimensions, so each matrix retains its own radius, tangent projection, and rotation angle.

In particular, the radii and angles have shape $B \times 1 \times 1$ and broadcast over the corresponding matrices. Parameters with different shapes are processed in separate groups.

The batch follows the same update sequence as Algorithm 1. We project the stored raw momentum buffers, add the gradients, update the radii, and extract the tangent momentum. Newton–Schulz matrix products use batched multiplication (`bmm`), with tangent reprojection after every iteration. When $m < n$, all matrices in the batch are transposed for these iterations and transposed back afterward. The spherical update is then evaluated for each matrix, and the weights are written back using `torch._foreach_copy_`. As in the single-matrix algorithm, we store the raw momentum buffers for projection at the next step.

This implementation batches the operations while preserving the per-matrix update rule. A five-step check found a numerical difference of 9×10^{-4} between the loop and batched implementations, within the reported `bf16` tolerance. On Qwen2-style-LM, batching increases throughput from $0.59\times$ to $0.72\times$ that of AdamW (Table 6, RODE loop and fused rows).

B Experimental setup and hyperparameters

B.1 Optimizer definitions and parameter routing

This appendix gives the exact optimizer routing, sweep budgets, and selected hyperparameters used in the reported experiments. We write $\eta_{2\mathrm{d/dir}}$ for the matrix-parameter learning rate; for RODE-family optimizers this is the angular learning rate η_{dir}, while for non-RODE matrix optimizers it is the ordinary matrix-parameter learning rate. We write η_ρ for RODE's radial learning rate and $\eta_{1\mathrm{d}}$ for the non-matrix learning rate. A dash denotes an unused parameter. Unless stated otherwise, language runs use two-dimensional non-embedding, non-head parameters as the matrix set, with all remaining parameters routed to the listed non-matrix optimizer. Vision runs route all parameters with `ndim >= 2` to the matrix optimizer; for Muon baselines, convolutional tensors are reshaped to matrices before the Muon step.

Table 7: Operational baseline definitions used in the code. Matrix weight decay is the task-level `--wd`; non-matrix AdamW uses weight decay 9×10^{-4} unless the row states otherwise.

Optimizer	Matrix-parameter update	Non-matrix update	Fixed optimizer constants
AdamW	AdamW on matrix parameters	AdamW on all remaining parameters	betas $(0.9, 0.95)$
Muon Original	Muon, original LR scaling	AdamW	momentum 0.95; matrix wd 0
Muon RMS	Muon, RMS-matched LR scaling	AdamW	momentum 0.95; matrix wd 0
RODE	Radial update + tangent Newton–Schulz geodesic step	AdamW	momentum 0.9; NS steps 5; $\kappa_{\mathrm{floor}} = 10^{-6}$; $\epsilon = 10^{-12}$
RODE-SNR	RODE Add tangent-gradient ratio damping	AdamW	RODE constants; ratio clamp $[0.01, 1]$
SOAP	SOAP on matrix parameters	AdamW	betas $(0.95, 0.95)$, precondition frequency 10, max precondition dim 10000
MARS-M	MARS-M on matrix parameters	AdamW	momentum 0.95, NS steps 5, $\gamma = 0.025$, approximate mode
SGD	SGD on matrix parameters	SGD on all remaining parameters	momentum 0.9

B.2 HYPERPARAMETER SEARCH AND SELECTION

For each sweep group in Table 8, all optimizers in that group use the listed number of Optuna trials, training horizon, data source, model configuration, evaluation protocol, and log-uniform search spaces. Qwen2-style sweeps select hyperparameters by validation loss at the end of each trial. On standard GPT-2, AdamW, Muon, and RODE variants use the mean training loss over the final 30 steps of a 300-step trial; SOAP and MARS-M use held-out loss at step 300. The selected hyperparameters remain fixed across formal multi-seed runs.

Configurations marked as transferred reuse hyperparameters selected on the source task without a new search on the target task. CIFAR-100 and ImageNet-1K use this setting. The 1.5B study reports both transfer and task-specific tuning results; its data splits are described in Appendix B.6.2. The 9B search protocol is given in Appendix B.7.

Table 8: Learning-rate sweep budgets and search spaces. All sweep ranges are log-uniform. The selected values from these sweeps are reported in the task-specific hyperparameter tables below. SOAP and MARS-M are swept separately on the Qwen2-style and standard-GPT-2 benchmarks; their standard-GPT-2-selected rates are transferred unchanged to CIFAR-100. The 1.5B sweeps use a constant learning rate during each 500-step trial.

Source	Optimizer	Budget	Search space
GPT-2	AdamW	40×300	$\eta : [10^{-4}, 10^{-2}]$; weight decay $[10^{-4}, 10^{-1}]$
GPT-2	Muon Original	40×300	$\eta_{\mathrm{2d/dir}} : [10^{-4}, 2\times10^{-1}]$; $\eta_{\mathrm{1d}} : [10^{-5}, 10^{-2}]$
GPT-2	Muon RMS	40×300	$\eta_{\mathrm{2d/dir}} : [10^{-5}, 5\times10^{-2}]$; $\eta_{\mathrm{1d}} : [10^{-5}, 10^{-2}]$
GPT-2	RODE/RODE-SNR	40×300	$\eta_{\mathrm{2d/dir}} : [10^{-4}, 5\times10^{-1}]$; $\eta_\rho : [10^{-5}, 10^{-1}]$; $\eta_{\mathrm{1d}} : [10^{-5}, 10^{-2}]$
GPT-2	SOAP	40×300	$\eta_{\mathrm{2d/dir}} : [3\times10^{-5}, 10^{-2}]$; $\eta_{\mathrm{1d}} : [3\times10^{-5}, 3\times10^{-3}]$
GPT-2	MARS-M	40×300	$\eta_{\mathrm{2d/dir}} : [3\times10^{-5}, 3\times10^{-2}]$; $\eta_{\mathrm{1d}} : [3\times10^{-5}, 3\times10^{-3}]$
Qwen2-style	AdamW	30×500	$\eta_{\mathrm{2d/dir}}, \eta_{\mathrm{1d}} : [3\times10^{-5}, 3\times10^{-3}]$
Qwen2-style	Muon Original/Muon RMS	30×500	$\eta_{\mathrm{2d/dir}} : [5\times10^{-3}, 3\times10^{-1}]$; $\eta_{\mathrm{1d}} : [3\times10^{-5}, 3\times10^{-3}]$
Qwen2-style	RODE/RODE-SNR	30×500	$\eta_{\mathrm{2d/dir}} : [10^{-4}, 5\times10^{-2}]$; $\eta_\rho : [10^{-6}, 5\times10^{-3}]$; $\eta_{\mathrm{1d}} : [3\times10^{-5}, 3\times10^{-3}]$
Qwen2-style	SOAP	30×500	$\eta_{\mathrm{2d/dir}} : [3\times10^{-5}, 10^{-2}]$; $\eta_{\mathrm{1d}} : [3\times10^{-5}, 3\times10^{-3}]$
Qwen2-style	MARS-M	30×500	$\eta_{\mathrm{2d/dir}} : [3\times10^{-5}, 3\times10^{-2}]$; $\eta_{\mathrm{1d}} : [3\times10^{-5}, 3\times10^{-3}]$
1.5B FineWeb	Muon RMS	20×500	$\eta_{\mathrm{2d/dir}} : [5\times10^{-4}, 5\times10^{-2}]$; $\eta_{\mathrm{1d}} : [10^{-4}, 10^{-2}]$
1.5B FineWeb	RODE	20×500	$\eta_{\mathrm{2d/dir}} : [10^{-4}, 10^{-2}]$; $\eta_\rho : [10^{-5}, 10^{-2}]$; $\eta_{\mathrm{1d}} : [10^{-4}, 10^{-2}]$
1.5B FineWeb	RODE, fixed ρ	20×500	$\eta_{\mathrm{2d/dir}} : [10^{-4}, 10^{-2}]$; $\eta_{\mathrm{1d}} : [10^{-4}, 10^{-2}]$

B.3 METRICS AND CURVE PROCESSING

Diagnostics include every trainable parameter, including parameters delegated to AdamW. For the diagnostic parameter set $\mathcal{P}_{\mathrm{diag}}$,

$$\|W_t\|_{\mathrm{global}} = \left(\sum_{p \in \mathcal{P}_{\mathrm{diag}}} \|p_t\|_{\mathrm{F}}^2 \right)^{1/2}, \tag{11}$$

$$\|\Delta W_t\|_{\mathrm{global}} = \left(\sum_{p \in \mathcal{P}_{\mathrm{diag}}} \|p_{t+1} - p_t\|_{\mathrm{F}}^2 \right)^{1/2}. \tag{12}$$

The update ratio divides the latter by $\|W_t\|_{\mathrm{global}} + 10^{-12}$. Gradient norm is the analogous root-sum-square immediately before the optimizer step. Only high-frequency training, gradient, and update diagnostics are EMA-smoothed for visualization; validation, accuracy, weight norm, schedules, and tabulated values use raw logs.

B.4 SMALL-MODEL LANGUAGE MODELING

Table 9: Standard GPT-2/WikiText-103 hyperparameters for the 8000-step main benchmark. Runs use sequence length 512, effective batch size 128, micro-batch size 16, `bf16` autocast, evaluation every 500 steps, 100 warmup steps, and cosine decay.

Optimizer	$\eta_{\mathrm{2d/dir}}$	η_ρ	η_{1d}	Matrix wd	1D wd	Source
AdamW	4.483×10^{-4}	–	4.483×10^{-4}	9.077×10^{-4}	9.077×10^{-4}	GPT-2 sweep
Muon Original	1.264×10^{-2}	–	1.314×10^{-5}	0	9×10^{-4}	GPT-2 sweep
Muon RMS	1.806×10^{-3}	–	9.270×10^{-4}	0	9×10^{-4}	GPT-2 sweep
RODE	6.430×10^{-4}	1.502×10^{-2}	5.012×10^{-3}	0	9×10^{-4}	GPT-2 sweep
RODE-SNR	2.413×10^{-3}	1.288×10^{-4}	2.310×10^{-4}	0	9×10^{-4}	GPT-2 sweep
SOAP	1.214×10^{-3}	–	2.799×10^{-3}	10^{-2}	9×10^{-4}	GPT-2 sweep
MARS-M	1.593×10^{-3}	–	1.868×10^{-3}	10^{-2}	9×10^{-4}	GPT-2 sweep

Table 10: Qwen2-style LM/WikiText-103 hyperparameters for the aligned 4000-step main benchmark and component ablation. The model has 114M parameters. All runs use sequence length 512, effective batch size 128, micro-batch size 16, `bf16` autocast, the official validation split, evaluation every 200 steps, 200 warmup steps, and cosine decay. This locally instantiated architecture uses a separate protocol from Table 9.

Optimizer	$\eta_{\mathrm{2d/dir}}$	η_ρ	η_{1d}	Matrix wd	1D wd	Source
AdamW	1.544×10^{-4}	–	2.988×10^{-3}	10^{-2}	9×10^{-4}	Qwen sweep
Muon Original	2.257×10^{-2}	–	2.897×10^{-3}	0	9×10^{-4}	Qwen sweep
Muon RMS	5.160×10^{-3}	–	2.913×10^{-3}	0	9×10^{-4}	Qwen sweep
RODE	1.206×10^{-3}	2.438×10^{-3}	2.915×10^{-3}	0	9×10^{-4}	Qwen sweep
RODE-SNR	4.544×10^{-3}	3.698×10^{-5}	2.974×10^{-3}	0	9×10^{-4}	Qwen sweep
SOAP	2.900×10^{-3}	–	2.591×10^{-3}	10^{-2}	9×10^{-4}	Qwen sweep
MARS-M	4.097×10^{-3}	–	2.693×10^{-3}	10^{-2}	9×10^{-4}	Qwen sweep

B.5 IMAGE CLASSIFICATION

Table 11: ResNet-50/CIFAR-100 hyperparameters. Runs use 100 epochs, image size 224, effective batch size 128, three distinct seeds, and epoch-wise cosine annealing. The implemented scheduler is `CosineAnnealingLR(T_max=epochs)` with no warmup stage. We report best test accuracy. No CIFAR-specific sweep is used: all optimizer-specific rates are transferred from their corresponding standard-GPT-2 sweeps.

Optimizer	$\eta_{\mathrm{2d/dir}}$	η_ρ	η_{1d}	Matrix wd	1D wd	Source
AdamW	4.480×10^{-4}	–	4.480×10^{-4}	10^{-2}	9×10^{-4}	GPT-2 transfer
Muon Original	1.264×10^{-2}	–	1.314×10^{-5}	0	9×10^{-4}	GPT-2 transfer
Muon RMS	1.806×10^{-3}	–	9.270×10^{-4}	0	9×10^{-4}	GPT-2 transfer
RODE	6.430×10^{-4}	1.502×10^{-2}	5.012×10^{-3}	0	9×10^{-4}	GPT-2 transfer
RODE-SNR	2.413×10^{-3}	1.288×10^{-4}	2.310×10^{-4}	0	9×10^{-4}	GPT-2 transfer
SOAP	1.214×10^{-3}	–	2.799×10^{-3}	10^{-2}	9×10^{-4}	GPT-2 transfer
MARS-M	1.593×10^{-3}	–	1.868×10^{-3}	10^{-2}	9×10^{-4}	GPT-2 transfer
SGD	1.0×10^{-1}	–	1.0×10^{-1}	5×10^{-4}	5×10^{-4}	standard

Table 12: ResNet-50/ImageNet-1K hyperparameters. Runs use 100 epochs, image size 224, effective batch size 256, three distinct seeds, and epoch-wise cosine annealing without an implemented warmup stage. Table 1 reports, for each seed, the best validation accuracy observed over 100 epochs. The added SOAP/MARS-M baselines are not run on ImageNet-1K. For RODE, the default-seed run used $\eta_{\mathrm{dir}} = 6.430\times10^{-4}$; the listed value rounds the full-precision sweep setting used for the other two seeds.

Optimizer	$\eta_{\mathrm{2d/dir}}$	η_ρ	η_{1d}	Matrix wd	1D wd	Source
AdamW	4.480×10^{-4}	–	4.480×10^{-4}	10^{-2}	9×10^{-4}	GPT-2 transfer
Muon Original	1.264×10^{-2}	–	1.314×10^{-5}	0	9×10^{-4}	GPT-2 transfer
Muon RMS	1.806×10^{-3}	–	9.270×10^{-4}	0	9×10^{-4}	GPT-2 transfer
RODE	6.432×10^{-4}	1.502×10^{-2}	5.012×10^{-3}	0	9×10^{-4}	GPT-2 transfer
RODE-SNR	2.413×10^{-3}	1.288×10^{-4}	2.310×10^{-4}	0	9×10^{-4}	GPT-2 transfer
SGD	1.0×10^{-1}	–	1.0×10^{-1}	5×10^{-4}	5×10^{-4}	standard

B.6 1.5B LANGUAGE MODELING

Both formal protocols use streaming FineWeb `sample-100BT`, sequence length 1024, effective batch size 128, micro-batch size 2 per GPU, 64 accumulation steps, `bf16` parameters/autocast, and `torch.compile(default)`. Within each protocol, all three methods use the same three distinct seeds for 20000 steps, with 200 warmup steps, cosine decay, and evaluation every 500 steps. Checkpoints are saved every 100 steps for transfer and every 500 steps for task-specific tuning; the latter retains the checkpoint with the lowest held-out loss. Matrix weight decay is zero and non-matrix AdamW weight decay is 9×10^{-4}. RODE uses the fused implementation with chunk size 2; fixed-radius RODE holds each managed matrix's radius at initialization. Muon RMS uses the PyTorch implementation.

The transfer evaluation cache contains 1024 length-1024 sequences from FineWeb `sample-10BT` and is identical across optimizers. Document-level exclusion from the `sample-100BT` training stream is not enforced (HuggingFaceFW, 2025). The separate held-out protocol for task-specific tuning is described in Appendix B.6.2.

B.6.1 LEARNING-RATE TRANSFER

We directly transfer the learning rates selected on the Qwen2-style 114M model in Appendix B.4 without a new search on the 1.5B model. RODE and Muon RMS each retain their respective settings. Fixed-radius RODE uses the same directional and non-matrix learning rates as RODE, with the radial learning rate set to zero. The transferred values are listed in Table 13, and the results are reported in Appendix C.4.

Table 13: Learning rates transferred from Qwen2-style 114M to the 1.5B/FineWeb experiment.

Optimizer	$\eta_{\mathrm{2d/dir}}$	η_ρ	η_{1d}	Source
RODE, fixed ρ	1.206×10^{-3}	0	2.915×10^{-3}	Qwen2-style 114M transfer
RODE	1.206×10^{-3}	2.438×10^{-3}	2.915×10^{-3}	Qwen2-style 114M transfer
Muon RMS	5.160×10^{-3}	–	2.913×10^{-3}	Qwen2-style 114M transfer

B.6.2 TASK-SPECIFIC TUNING

We tune Muon RMS, RODE, and fixed-radius RODE independently using 20 Optuna trials of 500 steps each, with constant learning rates and the search spaces in Table 8. Learning rates are selected by the final loss on the transfer study's evaluation cache and fixed for three-seed, 20,000-step formal runs. Formal training uses FineWeb shards 000–004 and evaluates on a separate cache of 1,024 length-1,024 sequences from shard 005, with no document overlap with training.

Table 14: Selected learning rates for Qwen2-style 1.5B/FineWeb.

Optimizer	$\eta_{\mathrm{2d/dir}}$	η_ρ	η_{1d}
Muon RMS	1.119×10^{-3}	–	1.117×10^{-3}
RODE	2.871×10^{-4}	5.305×10^{-5}	1.246×10^{-3}
RODE, fixed ρ	3.185×10^{-4}	0	6.984×10^{-4}

B.7 QWEN3.5-9B FINE-TUNING AND DOWNSTREAM EVALUATION

All methods fine-tune the same Qwen3.5-9B checkpoint (non-Base) on a fixed GSM8K split of 7,099 training and 374 validation examples (Cobbe et al., 2021). Each optimizer receives 20 Optuna trials of 800 steps, including four preset candidates for each of Muon Original and SOAP. Learning rates are selected by the lowest validation loss across evaluations every 100 steps.

Formal runs use three seeds and 3,000 steps, with sequence length 1,024, micro-batch size 1, gradient accumulation 8, BF16 full-parameter training, gradient checkpointing, gradient clipping at 1.0, zero

weight decay, and 3% linear warmup followed by cosine decay. Evaluation occurs every 100 steps; the checkpoint with the lowest validation loss is used for downstream evaluation.

Unlike the default routing in Appendix B.1, the matrix optimizers update all trainable two-dimensional tensors, including embeddings and output heads; remaining tensors use AdamW with betas $(0.9, 0.95)$. The AdamW baseline updates all parameters with one optimizer. SOAP uses betas $(0.9, 0.95)$, precondition frequency 100, and maximum precondition dimension 1,024, with dimension merging, one-dimensional preconditioning, and gradient normalization disabled.

GSM8K (1,319 examples) and MATH-500 (500) (Hendrycks et al., 2021b) use greedy decoding with at most 512 new tokens. Answers are normalized and matched by string equality or numerical comparison with relative and absolute tolerances of 10^{-6}. MMLU-STEM (3,153) and MMLU-Pro Math (1,351) (Hendrycks et al., 2021a; Wang et al., 2024) use fixed zero-shot prompts and select the option letter with the highest summed token log-likelihood. Accuracy is computed over each complete evaluation set.

Table 15: Qwen3.5-9B learning-rate sampling ranges and selected values. Sampling is log-uniform; "Same" denotes the shared AdamW learning rate.

Optimizer	Sampling range	Selected $\eta_{\mathrm{2d/dir}}$	Selected η_ρ	Selected η_{1d}
AdamW	$\eta : [3\times10^{-6}, 8\times10^{-5}]$	2.402×10^{-5}	–	Same
Muon Original	$\eta_{\mathrm{2d/dir}} : [10^{-4}, 5\times10^{-2}]$; $\eta_{\mathrm{1d}} : [2\times10^{-6}, 10^{-4}]$	2.869×10^{-4}	–	5.846×10^{-6}
Muon RMS	$\eta_{\mathrm{2d/dir}} : [3\times10^{-6}, 10^{-4}]$; $\eta_{\mathrm{1d}} = 2\times10^{-5}$ fixed	1.971×10^{-5}	–	2.000×10^{-5}
SOAP	$\eta_{\mathrm{2d/dir}} : [10^{-5}, 8\times10^{-3}]$; $\eta_{\mathrm{1d}} : [2\times10^{-6}, 2\times10^{-4}]$	2.118×10^{-5}	–	3.020×10^{-6}
RODE	$\eta_{\mathrm{2d/dir}} : [2\times10^{-6}, 3\times10^{-5}]$; $\eta_\rho : [10^{-6}, 5\times10^{-5}]$; $\eta_{\mathrm{1d}} : [3\times10^{-6}, 5\times10^{-5}]$	6.259×10^{-6}	2.919×10^{-6}	5.592×10^{-6}

B.8 Interventions and ablations

All experiments use WikiText-103, three distinct seeds, an effective batch size of 128, and cosine learning-rate decay. Table 16 lists the variants and training budgets.

Muon + NormProj inherits Muon RMS learning rates. RODE variants inherit RODE learning rates, except RODE-SNR, which uses its independently tuned rates in Table 10. Fixed-radius variants hold each matrix radius at its initial value.

Table 16: Intervention and ablation settings.

Experiment	Variants	Steps / warmup
Norm–direction intervention Standard GPT-2	Muon RMS; Muon + NormProj; RODE with fixed radius; full RODE	8,000 / 100
Component ablation Qwen2-style (114M)	RODE; RODE-SNR; fixed radius; radial only; w/o Newton–Schulz; w/o geodesic	4,000 / 200

C ADDITIONAL EXPERIMENTAL RESULTS AND DIAGNOSTICS

This appendix collects supporting diagnostics and full trajectories that complement the main results. The same plotting code and color mapping are used as in the main figures.

C.1 SMALL-MODEL LANGUAGE MODELING

Figure 4 shows the training loss, update step size, and gradient norm for the GPT-2 runs; the GPT-2 weight norm is in the main text (Figure 2).

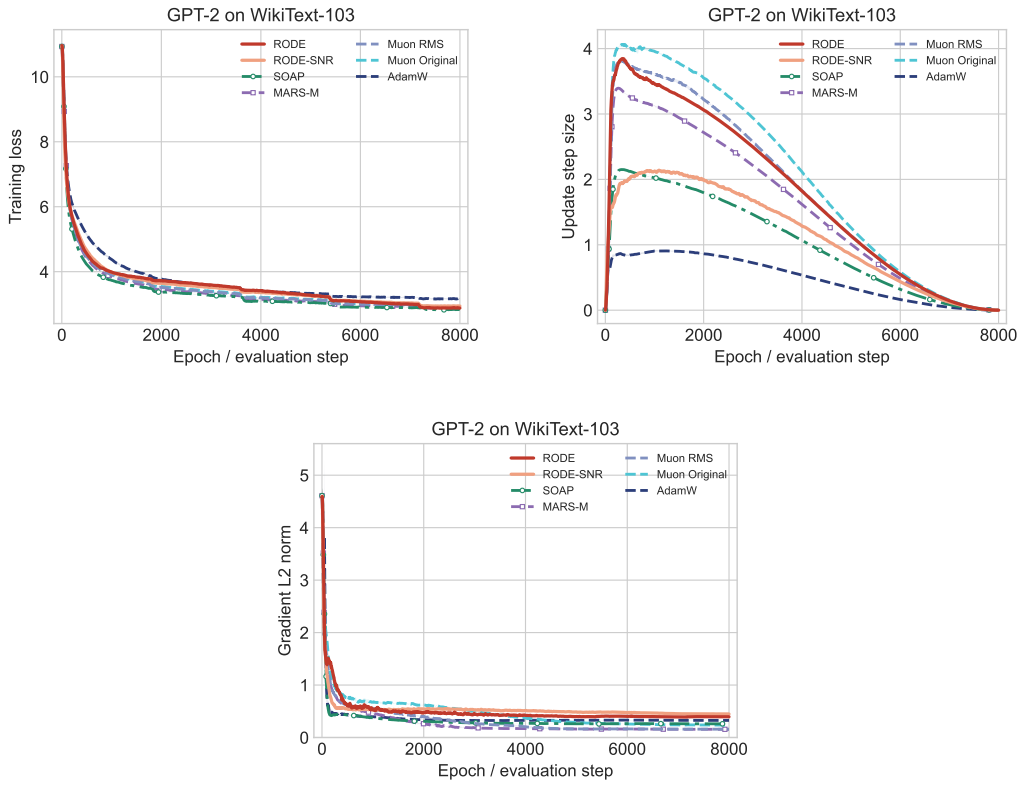

Figure 4: GPT-2/WikiText-103 training dynamics (three seeds): training loss, update step size, and gradient norm.

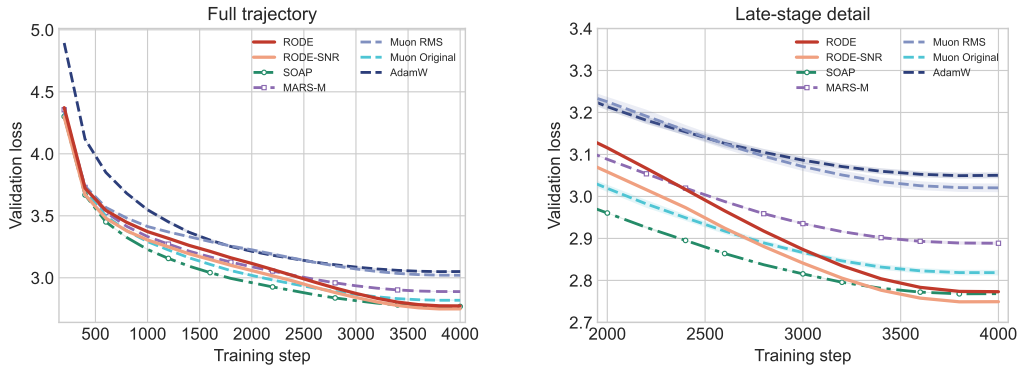

Figure 5: Qwen2-style LM/WikiText-103 validation trajectories over 4000 steps (three-seed mean ± sample SD): full trajectory (left) and late-stage detail (right).

C.2 Image classification

The vision curves accompany the transferred-rate results in Table 1.

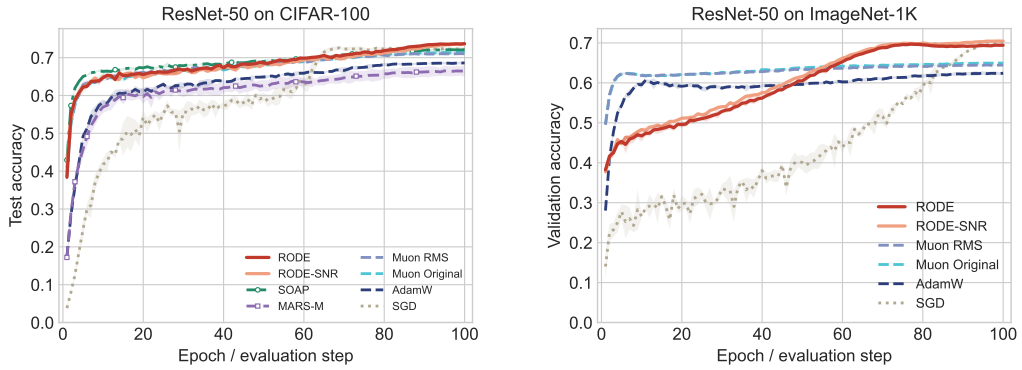

Figure 6: ResNet-50 test accuracy on CIFAR-100 (left) and validation accuracy on ImageNet-1K (right), each over three seeds.

Figures 7 and 8 show test loss for CIFAR-100, validation loss for ImageNet-1K, and observed full-model weight norm for both datasets.

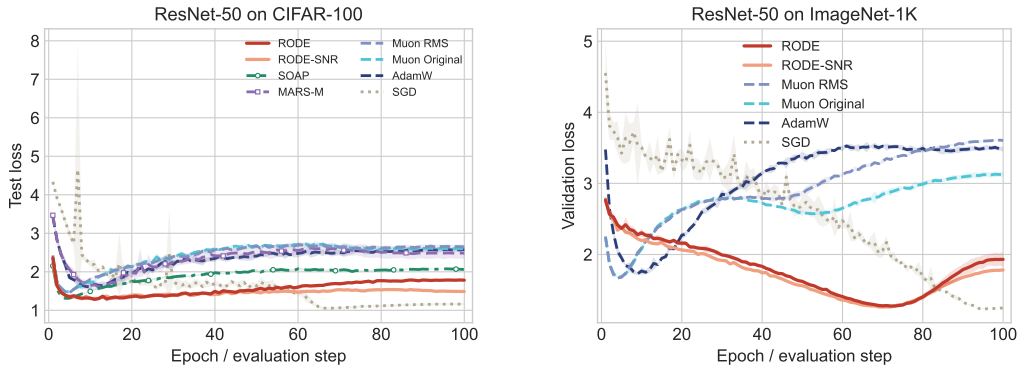

Figure 7: ResNet-50 test loss on CIFAR-100 (left) and validation loss on ImageNet-1K (right), shown as three-seed mean ± sample SD.

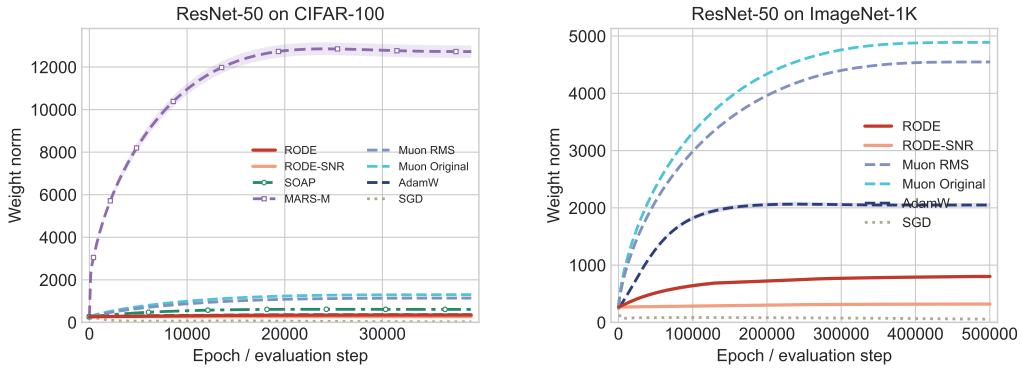

Figure 8: Observed full-model weight norms on CIFAR-100 (left) and ImageNet-1K (right), shown as three-seed mean ± sample SD. RODE directly updates managed-matrix radii; these curves additionally include parameters delegated to AdamW.

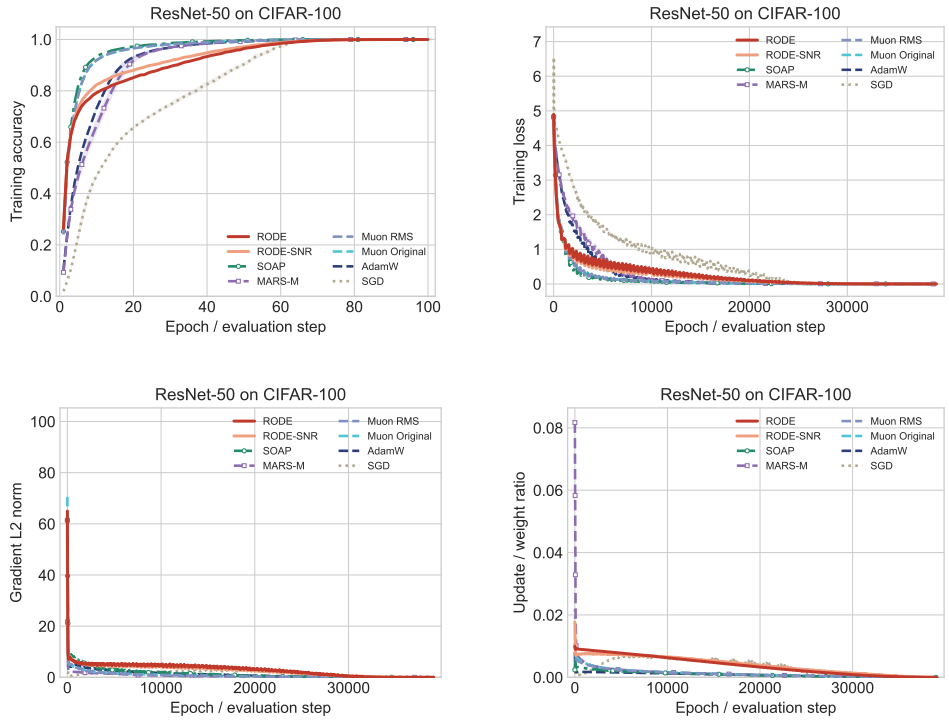

Figure 9: Additional ResNet-50/CIFAR-100 curves: training accuracy, training loss, gradient norm, and update-to-weight ratio.

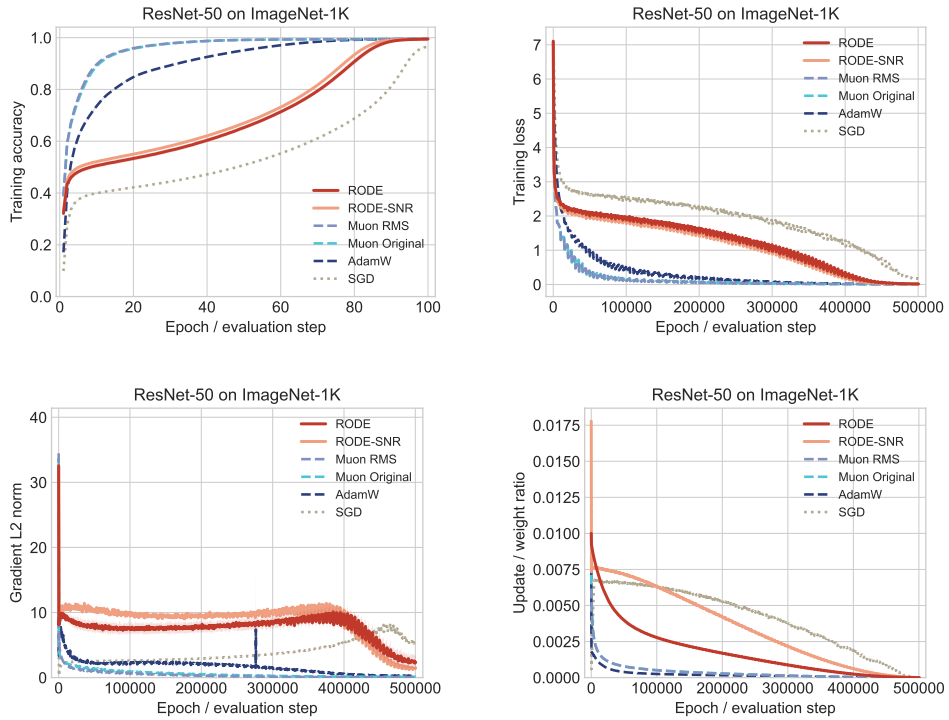

Figure 10: Additional ResNet-50/ImageNet-1K curves from the exported per-step diagnostics: three-seed means with sample-standard-deviation bands for training accuracy, training loss, gradient norm, and update-to-weight ratio. Main ImageNet validation results are reported in Table 1 and Figure 6.

C.3 Cross-task weight norms

The full-model norms below include every trainable parameter.

Table 17: Final observed full-model weight norm at the end of training (three-seed mean ± sample SD). Values include every trainable parameter, including tensors delegated to AdamW, and are comparable only within a task. A dash indicates that the task–optimizer combination was not run.

Optimizer	GPT-2	Qwen2-style LM	CIFAR-100	ImageNet-1K
AdamW	416.7 ± 2.5	711.1 ± 8.7	355.9 ± 0.7	2049.5 ± 34.7
Muon Original	1187.9 ± 0.8	1314.7 ± 1.9	1302.6 ± 0.6	4889.3 ± 0.5
Muon RMS	1074.4 ± 0.8	1722.4 ± 2.9	1142.0 ± 1.2	4547.4 ± 6.9
SGD	–	–	29.8 ± 0.7	56.6 ± 0.1
RODE	315.6 ± 6.6	683.5 ± 2.6	348.6 ± 0.3	801.8 ± 27.6
RODE-SNR	269.1 ± 0.3	695.8 ± 4.1	272.8 ± 0.1	320.6 ± 0.1
SOAP	691.6 ± 0.2	1046.9 ± 2.6	610.0 ± 0.3	–
MARS-M	959.1 ± 4.3	1417.6 ± 3.8	12726.0 ± 294.6	–

C.4 1.5B learning-rate transfer

We evaluate learning-rate transfer from the Qwen2-style 114M model to 1.5B pretraining on FineWeb without a new search, following Appendix B.6.1. Table 18 summarizes the results; Figures 11 and 12 show evaluation loss, weight norm, and training diagnostics.

Table 18: 1.5B scale transfer with rates selected on the smaller Qwen2-style LM (three seeds, mean ± sample SD).

Optimizer	Best cache loss ↓	Final cache loss ↓	Final observed global norm
RODE, fixed ρ	$\mathbf{3.13670 \pm 0.00406}$	$\mathbf{3.13670 \pm 0.00406}$	1743 ± 7
RODE	3.34568 ± 0.01108	3.34571 ± 0.01109	2183 ± 2
Muon RMS	4.14465 ± 0.01800	4.14478 ± 0.01806	11964 ± 32

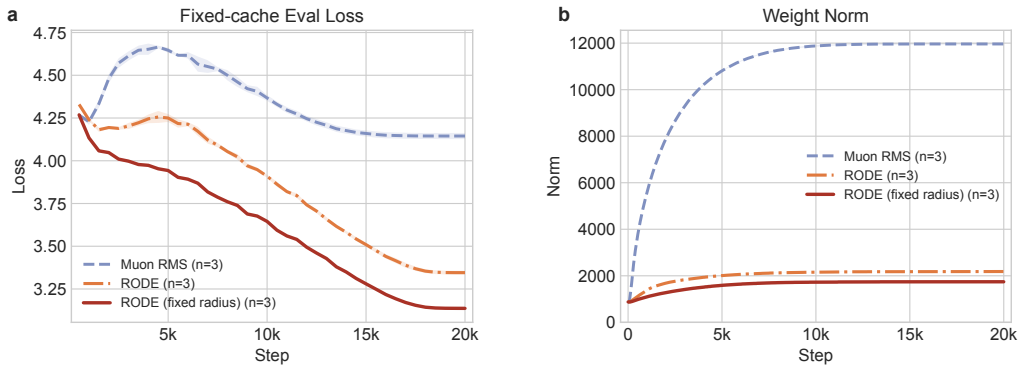

Figure 11: Qwen2-style 1.5B/FineWeb scale transfer over 20000 steps (three-seed mean ± sample SD): unsmoothed fixed-cache loss and observed global weight norm. Fixed-radius RODE freezes managed matrix radii; parameters delegated to AdamW can still change the global norm.

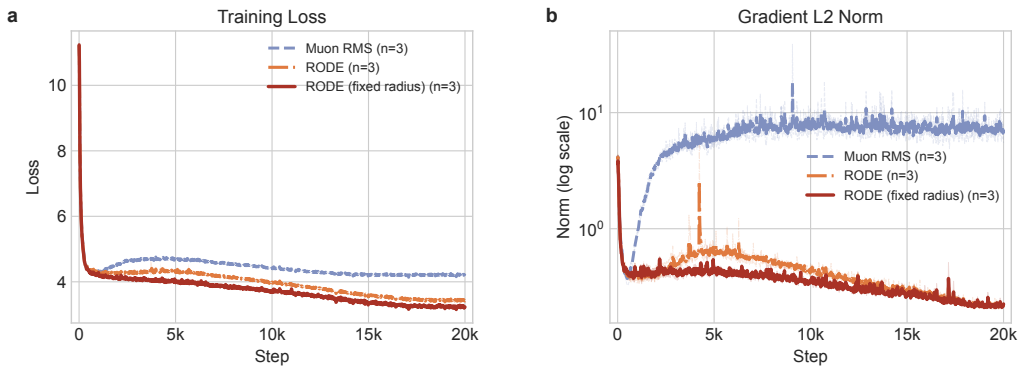

Figure 12: Qwen2-style 1.5B/FineWeb training diagnostics over 20000 steps (three seeds per method): EMA-smoothed training loss (left) and gradient norm (right). Gradient norm uses a logarithmic axis.

C.5 QWEN3.5-9B FINE-TUNING DIAGNOSTICS

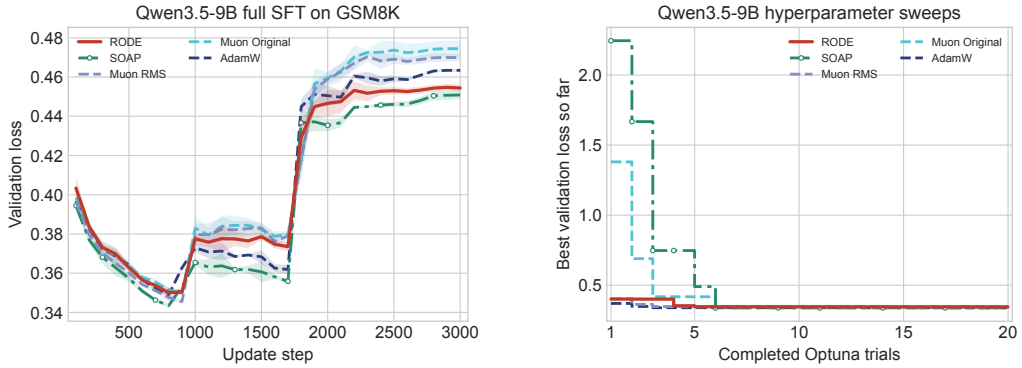

Figure 13: Qwen3.5-9B full-parameter fine-tuning diagnostics for the five optimizers in Table 5. Left: validation loss over the 3000-step formal runs (three-seed mean ± sample SD). Right: cumulative best objective over the 20 800-step Optuna trials for each optimizer; each trial's objective is its minimum validation loss. Both panels use raw, unsmoothed values.

D DETAILED GLOBAL CONVERGENCE PROOF

This appendix expands the convergence-neighborhood statement in Section 4. We derive the geodesic displacement, split radial and tangent descent, bound the tangent tracking error, and combine the potential and kinetic inequalities with a Lyapunov function.

D.1 NOTATION AND ALGORITHMIC EQUATIONS

Let $f(W) : \mathbb{R}^{m\times n} \to \mathbb{R}$ be the objective for a matrix parameter. For clarity assume $m \geq n$; if $m < n$, the implementation transposes the matrix before applying Newton–Schulz and transposes the result back. The Frobenius inner product is

$$\langle A, B\rangle_{\mathrm{F}} = \mathrm{tr}(A^\top B), \qquad \|A\|_{\mathrm{F}} = \sqrt{\langle A, A\rangle_{\mathrm{F}}}. \tag{13}$$

At iteration t, RODE writes

$$W_t = \rho_t U_t, \qquad \rho_t = \|W_t\|_{\mathrm{F}}, \qquad U_t = \frac{W_t}{\rho_t}, \qquad \|U_t\|_{\mathrm{F}} = 1. \tag{14}$$

Let $\mathcal{F}_t$ denote the history immediately before sampling the stochastic gradient at step t, and write $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot \mid \mathcal{F}_t]$. Let $G_t = \nabla f(W_t)$ denote the full gradient and let g_t denote a stochastic gradient satisfying $\mathbb{E}_t[g_t] = G_t$. We decompose both into radial and tangent components:

$$G_t^\rho = \langle G_t, U_t\rangle_{\mathrm{F}}, \qquad G_t^\perp = G_t - G_t^\rho U_t, \tag{15}$$

$$g_t^\rho = \langle g_t, U_t\rangle_{\mathrm{F}}, \qquad g_t^\perp = g_t - g_t^\rho U_t. \tag{16}$$

Thus W_t, U_t, ρ_t, and G_t are fixed under $\mathbb{E}_t$. Unsubscripted $\mathbb{E}$ denotes total expectation; below we pass from conditional one-step inequalities to total expectations using the tower property. Let $\eta_\rho > 0$ and $\eta_{\mathrm{dir}} > 0$ denote the fixed radial and directional learning rates, respectively. The radial update is

$$\rho_{t+1} = \rho_t - \eta_\rho g_t^\rho. \tag{17}$$

The implementation uses the guarded scalar update

$$\rho_{t+1} = \max\{\rho_t - \eta_\rho g_t^\rho, \kappa_{\mathrm{floor}}\rho_t\}. \tag{18}$$

Here $\kappa_{\mathrm{floor}} \in (0, 1)$ is the relative radius floor, set to 10^{-6} in our implementation. The analysis below assumes that the radius floor is inactive. The optimizer stores the raw momentum buffer B_t and obtains its tangent component by projection:

$$B_t = \beta\mathcal{P}_t(B_{t-1}) + g_t, \qquad M_{\mathrm{raw},t}^\perp = \mathcal{P}_t(B_t). \tag{19}$$

At the current step this gives the same tangent direction as projecting the newly formed buffer. Across steps, however, the radial part retained in B_t can re-enter the next tangent space through $\mathcal{P}_{t+1}(B_t)$. Write

$$B_t = M^{\perp}_{\mathrm{raw},t} + a_t U_t, \qquad a_t = \langle B_t, U_t \rangle_{\mathrm{F}}. \tag{20}$$

Then the next projected raw tangent buffer obeys

$$\begin{aligned} M^{\perp}_{\mathrm{raw},t+1} &= \mathcal{P}_{t+1}(B_{t+1}) \\ &= \beta \mathcal{P}_{t+1}(M^{\perp}_{\mathrm{raw},t}) + g^{\perp}_{t+1} + r_{t+1}, \end{aligned} \tag{21}$$

with the cross-step radial leakage residual

$$r_{t+1} = \beta a_t \mathcal{P}_{t+1}(U_t). \tag{22}$$

The implementation stores the unnormalized momentum above. For the tracking argument we analyze its normalized counterpart

$$M^{\perp}_t = (1-\beta) M^{\perp}_{\mathrm{raw},t}, \qquad \bar{r}_{t+1} = (1-\beta) r_{t+1}, \tag{23}$$

so that

$$M^{\perp}_{t+1} = \beta \mathcal{P}_{t+1}(M^{\perp}_t) + (1-\beta) g^{\perp}_{t+1} + \bar{r}_{t+1}. \tag{24}$$

The factor $1-\beta$ cancels when momentum is divided by its norm, so this rescaling preserves the Newton–Schulz direction. Below, we use the normalized momentum $M^{\perp}_t$ and include the effect of the cross-step residual $\bar{r}_{t+1}$ when bounding its error relative to the true tangent gradient $G^{\perp}_t$. For $M^{\perp}_t \neq 0$, the projected Newton–Schulz iteration starts from

$$X_0 = \frac{M^{\perp}_t}{\|M^{\perp}_t\|_{\mathrm{F}}} \tag{25}$$

and reprojects onto the tangent space after each quintic polynomial step:

$$Y_{k+1} = 3.4445 X_k + X_k \left(-4.7750 X_k^{\top} X_k + 2.0315 (X_k^{\top} X_k)^2 \right), \tag{26}$$

$$X_{k+1} = Y_{k+1} - \langle Y_{k+1}, U_t \rangle_{\mathrm{F}} U_t, \qquad k = 0, \ldots, K-1. \tag{27}$$

After K projected steps, define

$$\widehat{Q}_t = X_K. \tag{28}$$

Define its actual post-projection norm

$$q_t = \|\widehat{Q}_t\|_{\mathrm{F}}. \tag{29}$$

Projection removes the component parallel to U_t, so q_t is generally different from $\sqrt{n}$. The proof assumes that $\widehat{Q}_t$ retains a constant fraction of the polar-target alignment with $M^{\perp}_t$ and has nondegenerate projected norm, so that for some $c_{\mathrm{NS}} \in (0, 1]$,

$$\langle M^{\perp}_t, \widehat{Q}_t \rangle_{\mathrm{F}} \geq c_{\mathrm{NS}} \|M^{\perp}_t\|_*, \qquad 0 < q_{\min} \leq q_t \leq q_{\max} < \infty. \tag{30}$$

Define the unit tangent direction

$$D^{\perp}_t = \frac{\widehat{Q}_t}{q_t}, \qquad \langle D^{\perp}_t, U_t \rangle_{\mathrm{F}} = 0. \tag{31}$$

Let $\gamma_t \in [\gamma_{\min}, 1]$ be the scalar damping factor that scales the rotation angle, where $\gamma_{\min} > 0$ is a uniform lower bound. Default RODE sets $\gamma_t = \gamma_{\min} = 1$; RODE-SNR uses the factor defined in Appendix A.2. The rotation angle is

$$\theta_t = \eta_{\mathrm{dir}} \gamma_t q_t. \tag{32}$$

Finally,

$$U_{t+1} = U_t \cos\theta_t - D^{\perp}_t \sin\theta_t, \tag{33}$$

$$W_{t+1} = \rho_{t+1} U_{t+1}. \tag{34}$$

D.2 ASSUMPTIONS

Assumption 1 (Smoothness and lower boundedness)**.** *The objective is lower bounded: there exists a finite $f_{\inf}$ such that $f(W) \geq f_{\inf}$ for all W. For all matrices X, Y,*

$$f(Y) \leq f(X) + \langle \nabla f(X), Y - X \rangle_{\mathrm{F}} + \frac{L}{2}\|Y - X\|_{\mathrm{F}}^2. \tag{35}$$

Assumption 2 (Unbiased noise with bounded variance)**.** *The stochastic gradient is unbiased, and the radial and tangent noises are bounded:*

$$\mathbb{E}_t[g_t^\rho - G_t^\rho] = 0, \qquad \mathbb{E}_t[(g_t^\rho - G_t^\rho)^2] \leq \sigma_\rho^2, \tag{36}$$

$$\mathbb{E}_t[g_t^\perp - G_t^\perp] = 0, \qquad \mathbb{E}_t[\|g_t^\perp - G_t^\perp\|_{\mathrm{F}}^2] \leq \sigma_\perp^2. \tag{37}$$

Assumption 3 (Bounded physical states)**.** *There exist constants such that*

$$\begin{aligned} 0 < \rho_{\min} \leq \rho_t \leq \rho_{\max}, \quad |G_t^\rho| \leq G_{\max}, \\ \|M_t^\perp\|_{\mathrm{F}} \leq M_{\max}, \quad |a_t| \leq A_{\max} \quad \textit{for some } M_{\max}, A_{\max} > 0. \end{aligned} \tag{38}$$

Here $a_t = \langle B_t, U_t \rangle_{\mathrm{F}}$ is the radial component retained in the implemented raw buffer.

Assumption 4 (Local tangent-gradient regularity)**.** *Let $\mathcal{P}_{t+1}(X) = X - \langle X, U_{t+1} \rangle_{\mathrm{F}} U_{t+1}$. For the small geodesic moves considered here,*

$$\mathbb{E}\left[\|\mathcal{P}_{t+1}(G_t^\perp) - G_{t+1}^\perp\|_{\mathrm{F}}^2\right] \leq L^2 \mathbb{E}[\|\Delta W_t\|_{\mathrm{F}}^2], \tag{39}$$

where $\Delta W_t = W_{t+1} - W_t$.

Assumption 5 (Aligned projected Newton–Schulz kernel)**.** *After the finite projected Newton–Schulz iterations, the constant-step engine returns a tangent kernel satisfying, for some $c_{\mathrm{NS}} \in (0, 1]$,*

$$\langle M_t^\perp, \widehat{Q}_t \rangle_{\mathrm{F}} \geq c_{\mathrm{NS}} \|M_t^\perp\|_*, \qquad 0 < q_{\min} \leq \|\widehat{Q}_t\|_{\mathrm{F}} \leq q_{\max} < \infty. \tag{40}$$

This assumption isolates the optimizer dynamics from finite-iteration polynomial approximation errors, projection-induced alignment loss, and reduced-precision effects, and excludes degenerate steps where tangent projection nearly annihilates the conditioned kernel. The exact polar identity is recovered in the ideal case $c_{\mathrm{NS}} = 1$.

Assumption 6 (Bounded geodesic angle)**.** *There exists $\theta_{\max} \in (0, \pi/4]$ such that*

$$\eta_{\mathrm{dir}} \leq \frac{\theta_{\max}}{q_{\max}}. \tag{41}$$

Since $\gamma_t \leq 1$, this gives $\theta_t \leq \theta_{\max}$ and ensures the global lower bound

$$\sin(\theta_t) \geq k\theta_t, \qquad k = \frac{\sin(\theta_{\max})}{\theta_{\max}} \geq \frac{2\sqrt{2}}{\pi}. \tag{42}$$

D.3 RAW-BUFFER LEAKAGE BOUND

The geodesic update gives

$$U_{t+1} = U_t \cos\theta_t - D_t^\perp \sin\theta_t, \qquad \langle U_t, D_t^\perp \rangle_{\mathrm{F}} = 0. \tag{43}$$

The old radial direction U_t has inner product $\cos\theta_t$ with the new radial direction U_{t+1}. Its projection onto the new tangent space therefore has squared norm

$$\|\mathcal{P}_{t+1}(U_t)\|_{\mathrm{F}}^2 = 1 - \langle U_t, U_{t+1} \rangle_{\mathrm{F}}^2 = \sin^2\theta_t. \tag{44}$$

Taking square roots and using $\sin\theta_t \leq \theta_t$, $\gamma_t \leq 1$, and $q_t \leq q_{\max}$ gives

$$\|\mathcal{P}_{t+1}(U_t)\|_{\mathrm{F}} \leq \theta_t \leq \eta_{\mathrm{dir}} q_{\max}. \tag{45}$$

The radial component $a_t U_t$ retained in the momentum buffer contributes to the next tangent update through $\mathcal{P}_{t+1}(U_t)$. By (22) and (23), its contribution to the normalized momentum is $\bar{r}_{t+1} = (1-\beta)\beta a_t \mathcal{P}_{t+1}(U_t)$. Applying (45) and $|a_t| \leq A_{\max}$ gives

$$\|\bar{r}_{t+1}\|_{\mathrm{F}}^2 \leq (1-\beta)^2 \beta^2 A_{\max}^2 q_{\max}^2 \eta_{\mathrm{dir}}^2 =: K_{\mathrm{leak}} \eta_{\mathrm{dir}}^2. \tag{46}$$

D.4 Exact displacement expansion and second-order bound

The physical displacement is

$$\Delta W_t = W_{t+1} - W_t = \rho_{t+1}(U_t \cos\theta_t - D_t^{\perp} \sin\theta_t) - \rho_t U_t. \tag{47}$$

Collecting the orthogonal basis terms U_t and $D_t^{\perp}$ gives

$$\Delta W_t = (\rho_{t+1}\cos\theta_t - \rho_t)U_t - \rho_{t+1}\sin\theta_t D_t^{\perp}. \tag{48}$$

Because $\langle U_t, D_t^{\perp}\rangle_{\mathrm{F}} = 0$ and both directions have unit Frobenius norm,

$$\|\Delta W_t\|_{\mathrm{F}}^2 = (\rho_{t+1}\cos\theta_t - \rho_t)^2 + \rho_{t+1}^2 \sin^2\theta_t \tag{49}$$

$$= \rho_{t+1}^2\cos^2\theta_t - 2\rho_{t+1}\rho_t\cos\theta_t + \rho_t^2 + \rho_{t+1}^2\sin^2\theta_t \tag{50}$$

$$= \rho_{t+1}^2 + \rho_t^2 - 2\rho_{t+1}\rho_t\cos\theta_t \tag{51}$$

$$= (\rho_{t+1} - \rho_t)^2 + 2\rho_{t+1}\rho_t(1 - \cos\theta_t). \tag{52}$$

Using $1 - \cos\theta \le \theta^2/2$,

$$\|\Delta W_t\|_{\mathrm{F}}^2 \le (\rho_{t+1} - \rho_t)^2 + \rho_{t+1}\rho_t\theta_t^2. \tag{53}$$

Substituting $\rho_{t+1} - \rho_t = -\eta_\rho g_t^\rho$ and $\theta_t = \eta_{\mathrm{dir}}\gamma_t q_t$,

$$\|\Delta W_t\|_{\mathrm{F}}^2 \le \eta_\rho^2 (g_t^\rho)^2 + \rho_{t+1}\rho_t\eta_{\mathrm{dir}}^2\gamma_t^2 q_t^2. \tag{54}$$

Using $\rho_t, \rho_{t+1} \le \rho_{\max}$ from Assumption 3, $q_t \le q_{\max}$ from Assumption 5, and $\gamma_t \le 1$, we obtain

$$\|\Delta W_t\|_{\mathrm{F}}^2 \le \eta_\rho^2 (g_t^\rho)^2 + \rho_{\max}^2 q_{\max}^2 \eta_{\mathrm{dir}}^2. \tag{55}$$

Taking conditional expectation and using

$$\mathbb{E}_t[(g_t^\rho)^2] = (\mathbb{E}_t[g_t^\rho])^2 + \mathrm{Var}_t(g_t^\rho) \le (G_t^\rho)^2 + \sigma_\rho^2, \tag{56}$$

we obtain the second-order smoothness contribution

$$\frac{L}{2}\mathbb{E}_t[\|\Delta W_t\|_{\mathrm{F}}^2] \le \frac{L}{2}\eta_\rho^2 (G_t^\rho)^2 + \frac{L}{2}\eta_\rho^2\sigma_\rho^2 + \frac{L\rho_{\max}^2 q_{\max}^2}{2}\eta_{\mathrm{dir}}^2. \tag{57}$$

D.5 First-order descent: radial part

Applying the smoothness inequality (35) with $X = W_t$ and $Y = W_{t+1}$, the first-order term is

$$\langle G_t, \Delta W_t\rangle_{\mathrm{F}}. \tag{58}$$

Using $G_t = G_t^\rho U_t + G_t^{\perp}$ and the displacement expansion,

$$\langle G_t, \Delta W_t\rangle_{\mathrm{F}} = G_t^\rho(\rho_{t+1}\cos\theta_t - \rho_t) - \rho_{t+1}\sin\theta_t \langle G_t^{\perp}, D_t^{\perp}\rangle_{\mathrm{F}}. \tag{59}$$

For the radial component,

$$I_\rho = G_t^\rho(\rho_{t+1}\cos\theta_t - \rho_t) \tag{60}$$

$$= G_t^\rho(\rho_{t+1} - \rho_t) + G_t^\rho\rho_{t+1}(\cos\theta_t - 1) \tag{61}$$

$$= -\eta_\rho G_t^\rho g_t^\rho - \rho_{t+1}G_t^\rho(1 - \cos\theta_t). \tag{62}$$

Taking conditional expectation, the first term becomes

$$\mathbb{E}_t[-\eta_\rho G_t^\rho g_t^\rho] = -\eta_\rho G_t^\rho \mathbb{E}_t[g_t^\rho] = -\eta_\rho (G_t^\rho)^2. \tag{63}$$

For the curvature contamination term,

$$-\rho_{t+1}G_t^\rho(1 - \cos\theta_t) \le |\rho_{t+1}G_t^\rho|(1 - \cos\theta_t) \tag{64}$$

$$\le \rho_{\max}G_{\max}\frac{\theta_t^2}{2} \tag{65}$$

$$\le \frac{G_{\max}\rho_{\max}q_{\max}^2}{2}\eta_{\mathrm{dir}}^2. \tag{66}$$

The last line uses $\theta_t = \eta_{\mathrm{dir}}\gamma_t q_t$, $\gamma_t \le 1$, and the projected-kernel bound $q_t \le q_{\max}$. Therefore

$$\mathbb{E}_t[I_\rho] \le -\eta_\rho (G_t^\rho)^2 + \frac{G_{\max}\rho_{\max}q_{\max}^2}{2}\eta_{\mathrm{dir}}^2. \tag{67}$$

D.6 First-order descent: tangent part

The tangent component in (59) is

$$I_\perp = -\rho_{t+1}\sin\theta_t \langle G_t^\perp, D_t^\perp\rangle_{\mathrm{F}}. \tag{68}$$

Insert the identity $G_t^\perp = M_t^\perp - (M_t^\perp - G_t^\perp)$ and use $D_t^\perp = \widehat{Q}_t/q_t$:

$$I_\perp = -\rho_{t+1}\frac{\sin\theta_t}{q_t}\langle M_t^\perp, \widehat{Q}_t\rangle_{\mathrm{F}} + \rho_{t+1}\frac{\sin\theta_t}{q_t}\langle M_t^\perp - G_t^\perp, \widehat{Q}_t\rangle_{\mathrm{F}} \tag{69}$$

$$\le -\rho_{t+1}\frac{\sin\theta_t}{q_t}c_{\mathrm{NS}}\|M_t^\perp\|_* + \rho_{t+1}\frac{\sin\theta_t}{q_t}\langle M_t^\perp - G_t^\perp, \widehat{Q}_t\rangle_{\mathrm{F}}, \tag{70}$$

where the second line uses Assumption 5; the negative coefficient turns the alignment lower bound into an upper bound on $I_\perp$. By Assumption 6, $\sin\theta_t \ge k\theta_t$ for the negative term. The remaining inner product can have either sign, so we first upper-bound it by its absolute value and then use $\sin\theta_t \le \theta_t$. Since $\theta_t = \eta_{\mathrm{dir}}\gamma_t q_t$, the actual post-projection norm cancels:

$$I_\perp \le -k\rho_{t+1}\eta_{\mathrm{dir}}c_{\mathrm{NS}}\gamma_t\|M_t^\perp\|_* + \rho_{t+1}\eta_{\mathrm{dir}}\gamma_t\left|\langle M_t^\perp - G_t^\perp, \widehat{Q}_t\rangle_{\mathrm{F}}\right|. \tag{71}$$

We next lower bound the possibly damped and finite-NS-aligned nuclear norm. Since $\gamma_t \ge \gamma_{\min}$, $\|M\|_* \ge \|M\|_{\mathrm{F}}$, and $\|M_t^\perp\|_{\mathrm{F}} \le M_{\max}$,

$$c_{\mathrm{NS}}\gamma_t\|M_t^\perp\|_* \ge c_{\mathrm{NS}}\gamma_{\min}\|M_t^\perp\|_{\mathrm{F}} \ge \frac{c_{\mathrm{NS}}\gamma_{\min}}{M_{\max}}\|M_t^\perp\|_{\mathrm{F}}^2. \tag{72}$$

Using $\rho_{t+1} \ge \rho_{\min}$ and defining

$$\alpha_{\mathrm{dir}} = \frac{kc_{\mathrm{NS}}\rho_{\min}\gamma_{\min}}{M_{\max}}, \tag{73}$$

the main tangent term is bounded by

$$-k\rho_{t+1}\eta_{\mathrm{dir}}c_{\mathrm{NS}}\gamma_t\|M_t^\perp\|_* \le -\alpha_{\mathrm{dir}}\eta_{\mathrm{dir}}\|M_t^\perp\|_{\mathrm{F}}^2. \tag{74}$$

For the tracking-error term, Cauchy-Schwarz gives

$$\rho_{t+1}\eta_{\mathrm{dir}}\gamma_t\left|\langle M_t^\perp - G_t^\perp, \widehat{Q}_t\rangle_{\mathrm{F}}\right| \le \rho_{\max}\eta_{\mathrm{dir}}\|M_t^\perp - G_t^\perp\|_{\mathrm{F}}\|\widehat{Q}_t\|_{\mathrm{F}} \tag{75}$$

$$\le \rho_{\max}q_{\max}\eta_{\mathrm{dir}}\|M_t^\perp - G_t^\perp\|_{\mathrm{F}}. \tag{76}$$

Applying Young's inequality $ab \le a^2/2 + b^2/2$ with $a = \rho_{\max}q_{\max}\eta_{\mathrm{dir}}$ and $b = \|M_t^\perp - G_t^\perp\|_{\mathrm{F}}$,

$$\le \frac{1}{2}\rho_{\max}^2 q_{\max}^2\eta_{\mathrm{dir}}^2 + \frac{1}{2}\|M_t^\perp - G_t^\perp\|_{\mathrm{F}}^2. \tag{77}$$

Taking conditional expectation,

$$\mathbb{E}_t[I_\perp] \le -\alpha_{\mathrm{dir}}\eta_{\mathrm{dir}}\mathbb{E}_t[\|M_t^\perp\|_{\mathrm{F}}^2] + \frac{1}{2}\mathbb{E}_t[\|M_t^\perp - G_t^\perp\|_{\mathrm{F}}^2] + \frac{1}{2}\rho_{\max}^2 q_{\max}^2\eta_{\mathrm{dir}}^2. \tag{78}$$

D.7 Potential-energy inequality

By smoothness,

$$\mathbb{E}_t[f(W_{t+1})] - f(W_t) \le \mathbb{E}_t[\langle G_t, \Delta W_t\rangle_{\mathrm{F}}] + \frac{L}{2}\mathbb{E}_t[\|\Delta W_t\|_{\mathrm{F}}^2]. \tag{79}$$

Substituting (57), (67), and (78), then taking total expectation and using the tower property, we obtain

$$\mathbb{E}[f(W_{t+1})] - \mathbb{E}[f(W_t)] \le -\left(\eta_\rho - \frac{L}{2}\eta_\rho^2\right)\mathbb{E}[(G_t^\rho)^2] - \alpha_{\mathrm{dir}}\eta_{\mathrm{dir}}\mathbb{E}[\|M_t^\perp\|_{\mathrm{F}}^2]$$
$$+ \frac{1}{2}\mathbb{E}[\|M_t^\perp - G_t^\perp\|_{\mathrm{F}}^2] + C_1\eta_{\mathrm{dir}}^2 + \frac{L}{2}\sigma_\rho^2\eta_\rho^2, \tag{80}$$

where

$$C_1 = \frac{1}{2}\rho_{\max}^2 q_{\max}^2 + \frac{L}{2}\rho_{\max}^2 q_{\max}^2 + \frac{1}{2}G_{\max}\rho_{\max}q_{\max}^2. \tag{81}$$

Equation (80) bounds the change in expected loss over one update. When $0 < \eta_\rho < 2/L$, the radial and tangential terms contribute descent, while momentum error, curvature, and gradient noise offset this decrease. To establish convergence, we must also control the momentum error. Appendix D.8 derives its evolution, and Appendix D.9 combines the two bounds to obtain the convergence neighborhood.

D.8 TANGENT TRACKING-ERROR DYNAMICS

Define the tangent tracking error

$$E_t^\perp = \mathbb{E}[\|M_t^\perp - G_t^\perp\|_{\mathrm{F}}^2]. \tag{82}$$

Here $M_t^\perp$ is the normalized analysis momentum defined in Appendix D; the implemented projected raw tangent buffer $M_{\mathrm{raw},t}^\perp$ differs from it only by the positive scalar $(1-\beta)^{-1}$ before Newton–Schulz normalization. For the actual raw-buffer recurrence, the normalized analysis momentum obeys

$$M_{t+1}^\perp = \beta \mathcal{P}_{t+1}(M_t^\perp) + (1-\beta) g_{t+1}^\perp + \bar{r}_{t+1}. \tag{83}$$

Subtract $G_{t+1}^\perp$ and add/subtract $\mathcal{P}_{t+1}(G_t^\perp)$:

$$\begin{aligned} M_{t+1}^\perp - G_{t+1}^\perp = \beta \mathcal{P}_{t+1}(M_t^\perp - G_t^\perp) + \beta(\mathcal{P}_{t+1}(G_t^\perp) - G_{t+1}^\perp) \\ + (1-\beta)(g_{t+1}^\perp - G_{t+1}^\perp) + \bar{r}_{t+1}. \end{aligned} \tag{84}$$

Let

$$x_t = \beta \mathcal{P}_{t+1}(M_t^\perp - G_t^\perp), \qquad y_t = \beta(\mathcal{P}_{t+1}(G_t^\perp) - G_{t+1}^\perp), \tag{85}$$

$$z_t = (1-\beta)(g_{t+1}^\perp - G_{t+1}^\perp), \qquad w_t = \bar{r}_{t+1}. \tag{86}$$

Using

$$\|x+y+z+w\|^2 \le (1+\delta)\|x\|^2 + 3\left(1+\frac{1}{\delta}\right)(\|y\|^2 + \|z\|^2 + \|w\|^2), \tag{87}$$

we obtain

$$\begin{aligned} E_{t+1}^\perp \le{}& \beta^2(1+\delta)\mathbb{E}[\|\mathcal{P}_{t+1}(M_t^\perp - G_t^\perp)\|_{\mathrm{F}}^2] \\ &+ 3\beta^2\left(1+\frac{1}{\delta}\right)\mathbb{E}[\|\mathcal{P}_{t+1}(G_t^\perp) - G_{t+1}^\perp\|_{\mathrm{F}}^2] \\ &+ 3(1-\beta)^2\left(1+\frac{1}{\delta}\right)\sigma_\perp^2 \\ &+ 3\left(1+\frac{1}{\delta}\right)\mathbb{E}[\|\bar{r}_{t+1}\|_{\mathrm{F}}^2]. \end{aligned} \tag{88}$$

Orthogonal projection is non-expansive:

$$\|\mathcal{P}_{t+1}(M_t^\perp - G_t^\perp)\|_{\mathrm{F}}^2 \le \|M_t^\perp - G_t^\perp\|_{\mathrm{F}}^2. \tag{89}$$

Choose $\delta = (1-\beta)/(2\beta)$. Then

$$\beta^2(1+\delta) = \beta^2\left(1+\frac{1-\beta}{2\beta}\right) = \beta\frac{1+\beta}{2} < \beta, \tag{90}$$

and

$$3\left(1+\frac{1}{\delta}\right) = 3\left(1+\frac{2\beta}{1-\beta}\right) = 3\frac{1+\beta}{1-\beta} \le \frac{6}{1-\beta}. \tag{91}$$

Combining (88)–(91) with the tangent-gradient bound (39), the displacement bound (55), and the residual bound (46), we obtain

$$E_{t+1}^\perp \le \beta E_t^\perp + \frac{6\beta^2 L^2}{1-\beta}\mathbb{E}[\eta_\rho^2 (g_t^\rho)^2 + \rho_{\max}^2 q_{\max}^2 \eta_{\mathrm{dir}}^2] + 6(1-\beta)\sigma_\perp^2 + \frac{6K_{\mathrm{leak}}}{1-\beta}\eta_{\mathrm{dir}}^2. \tag{92}$$

Using $\mathbb{E}[(g_t^\rho)^2] \le \mathbb{E}[(G_t^\rho)^2] + \sigma_\rho^2$,

$$\begin{aligned} E_{t+1}^\perp - E_t^\perp \le{}& -(1-\beta)E_t^\perp + \frac{6\beta^2 L^2}{1-\beta}\eta_\rho^2 \mathbb{E}[(G_t^\rho)^2] + C_2 \eta_{\mathrm{dir}}^2 \\ &+ 6(1-\beta)\sigma_\perp^2 + \frac{6\beta^2 L^2}{1-\beta}\sigma_\rho^2 \eta_\rho^2, \end{aligned} \tag{93}$$

where

$$C_2 = \frac{6\beta^2 L^2 \rho_{\max}^2 q_{\max}^2}{1-\beta} + \frac{6K_{\mathrm{leak}}}{1-\beta}. \tag{94}$$

Equation (93) bounds the one-step change in the error between momentum and the true tangent gradient. The negative term $-(1-\beta)E_t^\perp$ counteracts additional error from parameter movement and gradient noise. In Appendix D.9, we combine a weighted version of (93) with the loss bound (80) so that the total coefficient of $E_t^\perp$ becomes negative, allowing us to control the loss and momentum error together.

D.9 Lyapunov synthesis and convergence neighborhood

Define the Lyapunov function

$$\Phi_t = \mathbb{E}[f(W_t)] + c_\perp E_t^\perp. \tag{95}$$

Choose

$$c_\perp = \frac{1}{1-\beta}, \tag{96}$$

so that the $E_t^\perp$ coefficient after combining (80) and $c_\perp$ times (93) becomes

$$\frac{1}{2} - c_\perp(1-\beta) = -\frac{1}{2}. \tag{97}$$

In the same sum of (80) and $c_\perp$ times (93), the coefficient multiplying $\mathbb{E}[(G_t^\rho)^2]$ is

$$-\left(\eta_\rho - \frac{L}{2}\eta_\rho^2\right) + c_\perp \frac{6\beta^2 L^2}{1-\beta}\eta_\rho^2 = -\eta_\rho + \frac{L}{2}\eta_\rho^2 + \frac{6\beta^2 L^2}{(1-\beta)^2}\eta_\rho^2 \tag{98}$$

$$= -\eta_\rho\left(1 - \frac{L}{2}\eta_\rho - \frac{6\beta^2 L^2}{(1-\beta)^2}\eta_\rho\right). \tag{99}$$

For sufficiently small η_ρ, the parenthesized term is positive. Denote the resulting positive constant by $\widehat{\alpha}_\rho > 0$. Combining all constant terms, including the raw-buffer leakage contribution $c_\perp C_2 \eta_{\mathrm{dir}}^2$, into $\mathcal{K}_{\mathrm{curv}}$, $\mathcal{K}_n^\perp$, and $\mathcal{K}_n^\rho$ gives

$$\begin{aligned}\Phi_{t+1} - \Phi_t \le &-\widehat{\alpha}_\rho \eta_\rho \mathbb{E}[(G_t^\rho)^2] - \alpha_{\mathrm{dir}}\eta_{\mathrm{dir}}\mathbb{E}[\|M_t^\perp\|_{\mathrm{F}}^2] - \frac{1}{2}E_t^\perp \\ &+ \mathcal{K}_{\mathrm{curv}}\eta_{\mathrm{dir}}^2 + \mathcal{K}_n^\perp \sigma_\perp^2 + \mathcal{K}_n^\rho \sigma_\rho^2 \eta_\rho^2.\end{aligned} \tag{100}$$

Summing from $t = 1$ to T,

$$\Phi_{T+1} - \Phi_1 \le \sum_{t=1}^{T}\begin{bmatrix} -\widehat{\alpha}_\rho \eta_\rho \mathbb{E}[(G_t^\rho)^2] - \alpha_{\mathrm{dir}}\eta_{\mathrm{dir}}\mathbb{E}[\|M_t^\perp\|_{\mathrm{F}}^2] - \frac{1}{2}E_t^\perp \\ + \mathcal{K}_{\mathrm{curv}}\eta_{\mathrm{dir}}^2 + \mathcal{K}_n^\perp \sigma_\perp^2 + \mathcal{K}_n^\rho \sigma_\rho^2 \eta_\rho^2 \end{bmatrix}. \tag{101}$$

Since f is lower bounded and $E_t^\perp \ge 0$, there is a finite $\Phi_{\inf}$ with $\Phi_t \ge \Phi_{\inf}$. Moving the nonnegative descent terms to the left and using $\Phi_{T+1} \ge \Phi_{\inf}$ gives the finite-time bound

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T}\left[\widehat{\alpha}_\rho \eta_\rho \mathbb{E}[(G_t^\rho)^2] + \alpha_{\mathrm{dir}}\eta_{\mathrm{dir}}\mathbb{E}[\|M_t^\perp\|_{\mathrm{F}}^2] + \frac{1}{2}E_t^\perp\right] \\ \le \mathcal{K}_{\mathrm{curv}}\eta_{\mathrm{dir}}^2 + \mathcal{K}_n^\perp \sigma_\perp^2 + \mathcal{K}_n^\rho \sigma_\rho^2 \eta_\rho^2 + \frac{\Phi_1 - \Phi_{\inf}}{T}.\end{aligned} \tag{102}$$

Since $(\Phi_1 - \Phi_{\inf})/T \to 0$, taking $\limsup_{T\to\infty}$ in (102) gives

$$\begin{aligned}\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\left[\widehat{\alpha}_\rho \eta_\rho \mathbb{E}[(G_t^\rho)^2] + \alpha_{\mathrm{dir}}\eta_{\mathrm{dir}}\mathbb{E}[\|M_t^\perp\|_{\mathrm{F}}^2] + \frac{1}{2}E_t^\perp\right] \\ \le \mathcal{K}_{\mathrm{curv}}\eta_{\mathrm{dir}}^2 + \mathcal{K}_n^\perp \sigma_\perp^2 + \mathcal{K}_n^\rho \sigma_\rho^2 \eta_\rho^2.\end{aligned}$$

Retaining each nonnegative term on the left in turn and dividing by its positive coefficient gives the three separate bounds in the following theorem.

Theorem 1 (Interior-regime decoupled convergence neighborhood)**.** *Under Assumptions 1–6, sufficiently small η_ρ, inactive radial floors, and the nondegenerate raw-buffer recurrence above, the time-averaged radial gradient, normalized tangent momentum, and tangent tracking error admit the following asymptotic upper bounds:*

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[(G_t^\rho)^2] \le \frac{\mathcal{K}_{\mathrm{curv}}}{\widehat{\alpha}_\rho}\frac{\eta_{\mathrm{dir}}^2}{\eta_\rho} + \frac{\mathcal{K}_n^\perp}{\widehat{\alpha}_\rho}\frac{\sigma_\perp^2}{\eta_\rho} + \frac{\mathcal{K}_n^\rho}{\widehat{\alpha}_\rho}\eta_\rho\sigma_\rho^2 =: \epsilon_\rho, \tag{103}$$

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|M_t^\perp\|_{\mathrm{F}}^2] \le \frac{\mathcal{K}_{\mathrm{curv}}}{\alpha_{\mathrm{dir}}}\eta_{\mathrm{dir}} + \frac{\mathcal{K}_n^\perp}{\alpha_{\mathrm{dir}}}\frac{\sigma_\perp^2}{\eta_{\mathrm{dir}}} + \frac{\mathcal{K}_n^\rho}{\alpha_{\mathrm{dir}}}\frac{\eta_\rho^2}{\eta_{\mathrm{dir}}}\sigma_\rho^2, \tag{104}$$

$$\limsup_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}E_t^\perp \le 2\mathcal{K}_{\mathrm{curv}}\eta_{\mathrm{dir}}^2 + 2\mathcal{K}_n^\perp\sigma_\perp^2 + 2\mathcal{K}_n^\rho\sigma_\rho^2\eta_\rho^2. \tag{105}$$

The true tangent gradient is controlled jointly by (104) *and* (105)*:*

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[\|G_t^{\perp}\|_{\mathrm{F}}^2] \le \epsilon_{\perp}, \tag{106}$$

where the explicit tolerance is

$$\epsilon_{\perp} := \left(\frac{2}{\alpha_{\mathrm{dir}}} + 4\eta_{\mathrm{dir}}\right)\left(\mathcal{K}_{\mathrm{curv}}\eta_{\mathrm{dir}} + \mathcal{K}_n^{\perp}\frac{\sigma_{\perp}^2}{\eta_{\mathrm{dir}}} + \mathcal{K}_n^{\rho}\frac{\eta_{\rho}^2\sigma_{\rho}^2}{\eta_{\mathrm{dir}}}\right). \tag{107}$$

By Assumption 6, the prefactor in (107) *satisfies*

$$\frac{2}{\alpha_{\mathrm{dir}}} + 4\eta_{\mathrm{dir}} \le \frac{2}{\alpha_{\mathrm{dir}}} + \frac{4\theta_{\max}}{q_{\max}}. \tag{108}$$

For fixed assumption constants, (108) *is a uniform bound over the admissible positive direction stepsizes. Absorbing this prefactor into the constants in* (107) *gives*

$$\epsilon_{\perp} = \mathcal{O}(\eta_{\mathrm{dir}}) + \mathcal{O}\left(\frac{\sigma_{\perp}^2}{\eta_{\mathrm{dir}}}\right) + \mathcal{O}\left(\frac{\eta_{\rho}^2}{\eta_{\mathrm{dir}}}\sigma_{\rho}^2\right). \tag{109}$$

Proof. In (102), retain each of the three nonnegative terms in turn, divide by its coefficient $\widehat{\alpha}_{\rho}\eta_{\rho}$, $\alpha_{\mathrm{dir}}\eta_{\mathrm{dir}}$, or $1/2$, and take $\limsup_{T\to\infty}$. This gives (103), (104), and (105), respectively.

To obtain (106), first write the true tangent gradient as the momentum minus its tracking error:

$$\|G_t^{\perp}\|_{\mathrm{F}}^2 = \|M_t^{\perp} - (M_t^{\perp} - G_t^{\perp})\|_{\mathrm{F}}^2 \le 2\|M_t^{\perp}\|_{\mathrm{F}}^2 + 2\|M_t^{\perp} - G_t^{\perp}\|_{\mathrm{F}}^2. \tag{110}$$

Taking expectations in (110), averaging over $t = 1, \dots, T$, and using $E_t^{\perp} = \mathbb{E}[\|M_t^{\perp} - G_t^{\perp}\|_{\mathrm{F}}^2]$ gives

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[\|G_t^{\perp}\|_{\mathrm{F}}^2] \le \frac{2}{T}\sum_{t=1}^{T} \mathbb{E}[\|M_t^{\perp}\|_{\mathrm{F}}^2] + \frac{2}{T}\sum_{t=1}^{T} E_t^{\perp}. \tag{111}$$

For clarity, denote the common residual in (102) by

$$\mathcal{R} := \mathcal{K}_{\mathrm{curv}}\eta_{\mathrm{dir}}^2 + \mathcal{K}_n^{\perp}\sigma_{\perp}^2 + \mathcal{K}_n^{\rho}\sigma_{\rho}^2\eta_{\rho}^2. \tag{112}$$

To connect (112) to the two terms on the right of (111), retain the momentum term in (102) and divide by $\alpha_{\mathrm{dir}}\eta_{\mathrm{dir}}/2$:

$$\frac{2}{T}\sum_{t=1}^{T} \mathbb{E}[\|M_t^{\perp}\|_{\mathrm{F}}^2] \le \frac{2}{\alpha_{\mathrm{dir}}\eta_{\mathrm{dir}}}\left(\mathcal{R} + \frac{\Phi_1 - \Phi_{\inf}}{T}\right). \tag{113}$$

Similarly, retaining the tracking-error term in (102) and multiplying by 4 gives

$$\frac{2}{T}\sum_{t=1}^{T} E_t^{\perp} \le 4\left(\mathcal{R} + \frac{\Phi_1 - \Phi_{\inf}}{T}\right). \tag{114}$$

Thus (113) bounds the first term on the right of (111), and (114) bounds the second. Substituting both yields

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[\|G_t^{\perp}\|_{\mathrm{F}}^2] \le \left(\frac{2}{\alpha_{\mathrm{dir}}\eta_{\mathrm{dir}}} + 4\right)\left(\mathcal{R} + \frac{\Phi_1 - \Phi_{\inf}}{T}\right). \tag{115}$$

Taking $\limsup_{T\to\infty}$ in (115) removes the initial-gap term and gives

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[\|G_t^{\perp}\|_{\mathrm{F}}^2] \le \frac{2\mathcal{R}}{\alpha_{\mathrm{dir}}\eta_{\mathrm{dir}}} + 4\mathcal{R} = \epsilon_{\perp}, \tag{116}$$

where expanding (112) gives exactly (107). □

The radial and tangent bounds together give a stationarity guarantee for the original objective $f(W)$. Since $\nabla f(W_t) = G_t^\rho U_t + G_t^\perp$, $\|U_t\|_{\mathrm{F}} = 1$, and $\langle U_t, G_t^\perp \rangle_{\mathrm{F}} = 0$,

$$\|\nabla f(W_t)\|_{\mathrm{F}}^2 = (G_t^\rho)^2 + \|G_t^\perp\|_{\mathrm{F}}^2. \tag{117}$$

Taking expectations and time averages in (117), then applying (103) and (106), yields

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[\|\nabla f(W_t)\|_{\mathrm{F}}^2] \le \epsilon_\rho + \epsilon_\perp. \tag{118}$$

Thus, under the stated assumptions, RODE attains approximate first-order stationarity in a time-averaged sense, with the mean squared full-gradient norm bounded by a tolerance determined by the learning rates and gradient noise.

E Linear Convergence under the Polyak–Łojasiewicz Condition

This appendix studies how radial and tangential updates together yield linear convergence of the loss. We first expand the spherical update and retain the terms proportional to the learning rate. We then analyze the resulting first-order model under the Polyak–Łojasiewicz condition.

We use a full-gradient, projected-polar model: the radial update uses the radial component of the true gradient, and the tangent update uses the model momentum and projected polar factor defined in Appendix E.3. We first relate this model to the spherical update.

Write

$$W_t = \rho_t U_t, \qquad \|U_t\|_{\mathrm{F}} = 1, \tag{119}$$

and let

$$G_t = \nabla f(W_t), \qquad G_t^\rho = \langle G_t, U_t \rangle_{\mathrm{F}}. \tag{120}$$

Using a common stepsize $\eta > 0$, write the radial and directional learning rates as

$$\eta_\rho = a_\rho \eta, \qquad \eta_{\mathrm{dir}} = a_{\mathrm{dir}} \eta, \tag{121}$$

where $a_\rho, a_{\mathrm{dir}} > 0$ are fixed ratios. This preserves the difference between the two learning rates while expressing the overall step size through η.

Let

$$O_t = \mathcal{P}_t(\mathrm{Polar}(M_t)), \qquad q_t = \|O_t\|_{\mathrm{F}}, \qquad D_t^\perp = \frac{O_t}{q_t}, \tag{122}$$

where M_t is the model momentum in Appendix E.3, and $\mathcal{P}_t$ projects onto the tangent space at U_t. Thus,

$$\langle U_t, D_t^\perp \rangle_{\mathrm{F}} = 0, \qquad \|D_t^\perp\|_{\mathrm{F}} = 1. \tag{123}$$

Consider an update with an inactive radius floor and $q_t > 0$. The radial increment and rotation angle are

$$\Delta\rho_t = -a_\rho \eta G_t^\rho, \qquad \theta_t = a_{\mathrm{dir}} \eta \gamma_t q_t, \tag{124}$$

where γ_t is the scalar damping factor for the rotation angle; default RODE uses $\gamma_t = 1$. The corresponding spherical update is

$$W_{t+1}^{\mathrm{geo}} = (\rho_t + \Delta\rho_t) \left(U_t \cos\theta_t - D_t^\perp \sin\theta_t \right). \tag{125}$$

The superscript geo distinguishes this spherical update from the first-order update below.

Using

$$\cos\theta_t = 1 + \mathcal{O}(\theta_t^2), \qquad \sin\theta_t = \theta_t + \mathcal{O}(\theta_t^3), \tag{126}$$

we can write (125) as

$$W_{t+1}^{\mathrm{geo}} - W_t = \Delta\rho_t U_t - \rho_t \theta_t D_t^\perp + R_t, \tag{127}$$

where the remainder is

$$\begin{aligned} R_t = {} & (\rho_t + \Delta\rho_t)(\cos\theta_t - 1) U_t \\ & - \left[\rho_t (\sin\theta_t - \theta_t) + \Delta\rho_t \sin\theta_t \right] D_t^\perp. \end{aligned} \tag{128}$$

These terms arise from the cosine curvature, the higher-order part of the sine, and the product of radial change and directional rotation. The inequalities

$$|1-\cos\theta| \le \frac{\theta^2}{2}, \qquad |\sin\theta-\theta| \le \frac{|\theta|^3}{6}, \qquad |\sin\theta| \le |\theta| \tag{129}$$

give

$$\|R_t\|_{\mathrm{F}} \le \frac{|\rho_t+\Delta\rho_t|}{2}\theta_t^2 + \frac{\rho_t}{6}|\theta_t|^3 + |\Delta\rho_t|\,|\theta_t|. \tag{130}$$

If the radius is bounded and both $|\Delta\rho_t|$ and $|\theta_t|$ are bounded by constants times η, then

$$\|R_t\|_{\mathrm{F}} = \mathcal{O}(\eta^2). \tag{131}$$

Thus, for small stepsizes, the two first-order terms in (127) describe the leading change in the spherical update.

Substituting (124) and $q_t D_t^{\perp} = O_t$ into (127) yields

$$W_{t+1}^{\mathrm{geo}} = W_t - \eta\,(a_\rho G_t^\rho U_t + a_{\mathrm{dir}}\rho_t\gamma_t O_t) + R_t. \tag{132}$$

Define the radial and tangent effective directions by

$$H_{\mathrm{rad},t} = a_\rho G_t^\rho U_t, \qquad H_{\tan,t} = a_{\mathrm{dir}}\rho_t\gamma_t O_t, \tag{133}$$

and their sum by

$$H_t = H_{\mathrm{rad},t} + H_{\tan,t}. \tag{134}$$

The factors a_ρ and a_{dir} retain the learning-rate ratios, while ρ_t converts directional rotation into a displacement of the parameter matrix. Retaining the first-order terms in (132) defines the update model analyzed below:

$$W_{t+1} = W_t - \eta H_t. \tag{135}$$

Appendix E.1 states the three conditions used in the PL theorem. Appendices E.2–E.4 give sufficient conditions for the radial and tangential update directions to align with their corresponding gradient components in the projected-polar model, introducing two local regularity assumptions where needed. Appendix E.5 then derives the loss-gap recursion and its convergence neighborhood.

E.1 Assumptions for Linear Convergence

Theorem 2 uses the following three assumptions for (135).

Assumption 7 (PL smooth setting)**.** *The objective is L-smooth, has finite optimal value $f^* = \inf_W f(W)$, and satisfies the Polyak–Lojasiewicz condition (Karimi et al., 2016) for some $\mu > 0$:*

$$\frac{1}{2}\|\nabla f(W_t)\|_{\mathrm{F}}^2 \ge \mu(f(W_t) - f^*). \tag{136}$$

Assumption 8 (Bounded effective direction)**.** *The full effective update direction H_t satisfies $\|H_t\|_{\mathrm{F}} \le G$.*

Assumption 9 (Strong directional alignment)**.** *There exists a constant $\alpha > 0$, uniform along the trajectory, such that*

$$\langle G_t, H_t\rangle_{\mathrm{F}} \ge \alpha\|G_t\|_{\mathrm{F}}^2. \tag{137}$$

The next three subsections give sufficient conditions for (137) in the projected-polar model.

E.2 Radial alignment

Lemma 1 (Radial alignment)**.** *In the first-order model, gradient descent on the radius with U_t fixed gives radial alignment with coefficient $K_{\mathrm{rad}} = a_\rho$.*

Proof. Write $W_t = \rho_t U_t$ with $\|U_t\|_{\mathrm{F}} = 1$. Holding U_t fixed, the chain rule gives the derivative of the objective with respect to the radius:

$$\frac{\mathrm{d}}{\mathrm{d}\rho} f(\rho U_t)\bigg|_{\rho=\rho_t} = \langle \nabla f(W_t), U_t\rangle_{\mathrm{F}} = \langle G_t, U_t\rangle_{\mathrm{F}}. \tag{138}$$

Multiplying this scalar component by U_t gives the radial gradient matrix

$$G_{\mathrm{rad}} = \langle G_t, U_t \rangle_{\mathrm{F}} U_t. \tag{139}$$

A gradient step on the radius with learning rate $a_\rho \eta$, using (138), is

$$\Delta\rho_t = -a_\rho \eta \langle G_t, U_t \rangle_{\mathrm{F}}. \tag{140}$$

Since U_t is fixed during this radial step, (140) and (139) yield the corresponding matrix displacement:

$$\Delta W_{\mathrm{rad},t} = \Delta\rho_t U_t = -a_\rho \eta \langle G_t, U_t \rangle_{\mathrm{F}} U_t = -a_\rho \eta G_{\mathrm{rad}}. \tag{141}$$

Comparing (141) with the update convention in (135), the radial contribution to H_t is therefore $H_{\mathrm{rad}} = a_\rho G_{\mathrm{rad}}$. Consequently,

$$\langle G_{\mathrm{rad}}, H_{\mathrm{rad}} \rangle_{\mathrm{F}} = a_\rho \|G_{\mathrm{rad}}\|_{\mathrm{F}}^2 = K_{\mathrm{rad}} \|G_{\mathrm{rad}}\|_{\mathrm{F}}^2, \qquad K_{\mathrm{rad}} = a_\rho. \tag{142}$$

This is the radial alignment bound used in Appendix E.4. □

E.3 Tangent alignment

The tangent-alignment proof uses Assumption 8 and the following two local assumptions.

Assumption 10 (Local reprojected tangent-gradient regularity)**.** *Let $G_i^{\tan}$ denote the tangent gradient from step i reprojected into the current tangent space at U_t, and let $G_{\tan} = G_t^{\perp}$. Assume there is a constant $L_{\tan}$ such that*

$$\|G_i^{\tan} - G_{\tan}\|_{\mathrm{F}} \le L_{\tan} \|W_i - W_t\|_{\mathrm{F}}. \tag{143}$$

Assumption 11 (Bounded states and local polar stability)**.** *Along the trajectory considered here, assume $0 < \rho_{\min} \le \rho_t \le \rho_{\max} < \infty$ and $\|G_{\tan}\|_{\mathrm{F}} \le G_{\max}$. For the momentum tracking errors E considered below, assume there is a constant C_{pol} such that*

$$\|\mathcal{P}_t(\mathrm{Polar}(G_{\tan} + E)) - \mathcal{P}_t(\mathrm{Polar}(G_{\tan}))\|_{\mathrm{F}} \le C_{\mathrm{pol}} \|E\|_{\mathrm{F}}. \tag{144}$$

Lemma 2 (Tangent alignment in the projected-polar model)**.** *Under Assumptions 8, 10, and 11, with $0 < \beta < 1$ and sufficiently small learning rate, the projected-polar tangent step in the first-order model remains positively aligned with the true tangent gradient. The same result holds for scalar damping bounded below by a positive constant.*

Proof. Let $G_{\tan}$ denote the tangent component of G_t at the current direction U_t. For this sufficient-condition argument, define the model momentum as an EMA of gradients reprojected into the current tangent space: $G_i^{\tan}$ denotes a past tangent gradient reprojected into the current tangent space, and

$$M_t = \sum_{i=0}^{t} w_i G_i^{\tan}, \tag{145}$$

where, with initialization at step 0,

$$w_i = \begin{cases} (1-\beta)\beta^{t-i}, & i \ge 1, \\ \beta^t, & i = 0. \end{cases} \tag{146}$$

The weights sum to one:

$$\sum_{i=0}^{t} w_i = \left(\sum_{i=1}^{t} (1-\beta)\beta^{t-i} \right) + \beta^t \tag{147}$$

$$= (1 - \beta^t) + \beta^t = 1. \tag{148}$$

Define the momentum tracking error $E_t = M_t - G_{\tan}$. Because $\sum_i w_i = 1$,

$$E_t = \sum_{i=0}^{t} w_i G_i^{\tan} - \sum_{i=0}^{t} w_i G_{\tan} \tag{149}$$

$$= \sum_{i=0}^{t} w_i (G_i^{\tan} - G_{\tan}). \tag{150}$$

By the triangle inequality,

$$\|E_t\|_{\mathrm{F}} \le \sum_{i=0}^{t} w_i \|G_i^{\tan} - G_{\tan}\|_{\mathrm{F}}. \tag{151}$$

Using Assumption 10,

$$\|G_i^{\tan} - G_{\tan}\|_{\mathrm{F}} \le L_{\tan} \|W_i - W_t\|_{\mathrm{F}}. \tag{152}$$

By (135) and Assumption 8,

$$\|W_i - W_t\|_{\mathrm{F}} = \left\| \sum_{k=i}^{t-1} \eta H_k \right\|_{\mathrm{F}} \tag{153}$$

$$\le \sum_{k=i}^{t-1} \eta \|H_k\|_{\mathrm{F}} \tag{154}$$

$$\le (t-i)\eta G. \tag{155}$$

Thus

$$\|E_t\|_{\mathrm{F}} \le L_{\tan} G \eta \sum_{i=0}^{t} w_i (t-i). \tag{156}$$

Let

$$S = \sum_{i=0}^{t} w_i (t-i) = t\beta^t + (1-\beta) \sum_{k=1}^{t-1} k\beta^k. \tag{157}$$

For $A = \sum_{k=1}^{t-1} k\beta^k$,

$$A = \beta + 2\beta^2 + \cdots + (t-1)\beta^{t-1}, \tag{158}$$

and

$$\beta A = \beta^2 + 2\beta^3 + \cdots + (t-2)\beta^{t-1} + (t-1)\beta^t. \tag{159}$$

Subtracting,

$$(1-\beta)A = \beta + \beta^2 + \cdots + \beta^{t-1} - (t-1)\beta^t \tag{160}$$

$$= \frac{\beta(1-\beta^{t-1})}{1-\beta} - (t-1)\beta^t. \tag{161}$$

Substituting into S,

$$S = t\beta^t + \frac{\beta - \beta^t}{1-\beta} - t\beta^t + \beta^t \tag{162}$$

$$= \frac{\beta - \beta^{t+1}}{1-\beta} < \frac{\beta}{1-\beta}. \tag{163}$$

Therefore

$$\|E_t\|_{\mathrm{F}} \le L_{\tan} G \eta \frac{\beta}{1-\beta} = C_1 \eta. \tag{164}$$

Now consider the projected polar factor. Let

$$O_t = \mathcal{P}_t(\mathrm{Polar}(M_t)) = \mathcal{P}_t(\mathrm{Polar}(G_{\tan} + E_t)), \qquad O_g = \mathcal{P}_t(\mathrm{Polar}(G_{\tan})). \tag{165}$$

By Assumption 11, the polar factor is locally stable, so

$$O_t = O_g + \Delta_o, \qquad \|\Delta_o\|_{\mathrm{F}} \le C_{\mathrm{pol}} \|E_t\|_{\mathrm{F}} \le C_1 C_{\mathrm{pol}} \eta. \tag{166}$$

By (133), the first-order tangent contribution is $H_{\tan} = a_{\mathrm{dir}} \rho_t \gamma_t O_t$, which is tangent by construction because of the projection $\mathcal{P}_t$. For default RODE, $\gamma_t = 1$; for an optional damped variant, assume

$$0 < \gamma_{\min} \le \gamma_t \le 1. \tag{167}$$

Then

$$\langle G_{\tan}, H_{\tan} \rangle_{\mathrm{F}} = a_{\mathrm{dir}} \rho_t \gamma_t \left(\langle G_{\tan}, O_g \rangle_{\mathrm{F}} + \langle G_{\tan}, \Delta_o \rangle_{\mathrm{F}} \right). \tag{168}$$

For the projected polar target, since $G_{\tan}$ lies in the current tangent space,

$$\langle G_{\tan}, O_g \rangle_{\mathrm{F}} = \langle G_{\tan}, \mathrm{Polar}(G_{\tan}) \rangle_{\mathrm{F}} = \|G_{\tan}\|_* \geq \|G_{\tan}\|_{\mathrm{F}}. \tag{169}$$

For the perturbation term,

$$|\langle G_{\tan}, \Delta_o \rangle_{\mathrm{F}}| \leq \|G_{\tan}\|_{\mathrm{F}} \|\Delta_o\|_{\mathrm{F}} \leq C_1 C_{\mathrm{pol}} \eta \|G_{\tan}\|_{\mathrm{F}}. \tag{170}$$

For $\eta < 1/(C_1 C_{\mathrm{pol}})$, using $\rho_t \geq \rho_{\min}$ and $\gamma_t \geq \gamma_{\min}$ gives

$$\langle G_{\tan}, H_{\tan} \rangle_{\mathrm{F}} \geq a_{\mathrm{dir}} \rho_t \gamma_t (1 - C_1 C_{\mathrm{pol}} \eta) \|G_{\tan}\|_{\mathrm{F}} \tag{171}$$

$$\geq a_{\mathrm{dir}} \rho_{\min} \gamma_{\min} (1 - C_1 C_{\mathrm{pol}} \eta) \|G_{\tan}\|_{\mathrm{F}}. \tag{172}$$

If $\|G_{\tan}\|_{\mathrm{F}} \leq G_{\max}$, then

$$\|G_{\tan}\|_{\mathrm{F}} \geq \frac{\|G_{\tan}\|_{\mathrm{F}}^2}{G_{\max}}. \tag{173}$$

Therefore

$$\langle G_{\tan}, H_{\tan} \rangle_{\mathrm{F}} \geq \frac{a_{\mathrm{dir}} \rho_{\min} \gamma_{\min} (1 - C_1 C_{\mathrm{pol}} \eta)}{G_{\max}} \|G_{\tan}\|_{\mathrm{F}}^2. \tag{174}$$

For $\eta < 1/(C_1 C_{\mathrm{pol}})$, define

$$K_{\tan} = \frac{a_{\mathrm{dir}} \rho_{\min} \gamma_{\min} (1 - C_1 C_{\mathrm{pol}} \eta)}{G_{\max}} > 0. \tag{175}$$

This proves tangent alignment. □

E.4 Alignment of the full update direction

The previous two subsections establish radial and tangent alignment separately. We now use the orthogonality of these components to show that the full update direction H_t also aligns with the full gradient G_t.

Lemma 3 (Full-direction alignment)**.** *Decompose the gradient and effective update direction into radial and tangent components at U_t:*

$$G_t = G_{\mathrm{rad}} + G_{\tan}, \qquad H_t = H_{\mathrm{rad}} + H_{\tan}. \tag{176}$$

Suppose constants $K_{\mathrm{rad}}, K_{\tan} > 0$, uniform along the trajectory, satisfy

$$\langle G_{\mathrm{rad}}, H_{\mathrm{rad}} \rangle_{\mathrm{F}} \geq K_{\mathrm{rad}} \|G_{\mathrm{rad}}\|_{\mathrm{F}}^2, \tag{177}$$

$$\langle G_{\tan}, H_{\tan} \rangle_{\mathrm{F}} \geq K_{\tan} \|G_{\tan}\|_{\mathrm{F}}^2. \tag{178}$$

Then the full update direction satisfies

$$\langle G_t, H_t \rangle_{\mathrm{F}} \geq \alpha \|G_t\|_{\mathrm{F}}^2, \qquad \alpha = \min\{K_{\mathrm{rad}}, K_{\tan}\} > 0. \tag{179}$$

This is the full-direction alignment condition in Assumption 9. Lemmas 1 and 2 provide (177) *and* (178)*, respectively, under their stated conditions.*

Proof. Radial and tangent components are orthogonal, so the cross terms vanish:

$$\langle G_{\mathrm{rad}}, H_{\tan} \rangle_{\mathrm{F}} = \langle G_{\tan}, H_{\mathrm{rad}} \rangle_{\mathrm{F}} = 0. \tag{180}$$

Thus, the full inner product is the sum of the two component inner products. Applying (177) and (178), followed by the definition of α, gives

$$\langle G_t, H_t \rangle_{\mathrm{F}} = \langle G_{\mathrm{rad}}, H_{\mathrm{rad}} \rangle_{\mathrm{F}} + \langle G_{\tan}, H_{\tan} \rangle_{\mathrm{F}} \tag{181}$$

$$\geq K_{\mathrm{rad}} \|G_{\mathrm{rad}}\|_{\mathrm{F}}^2 + K_{\tan} \|G_{\tan}\|_{\mathrm{F}}^2 \tag{182}$$

$$\geq \alpha \left(\|G_{\mathrm{rad}}\|_{\mathrm{F}}^2 + \|G_{\tan}\|_{\mathrm{F}}^2 \right). \tag{183}$$

Orthogonality also gives

$$\|G_t\|_{\mathrm{F}}^2 = \|G_{\mathrm{rad}}\|_{\mathrm{F}}^2 + \|G_{\tan}\|_{\mathrm{F}}^2. \tag{184}$$

Substituting (184) into (183) yields

$$\langle G_t, H_t \rangle_{\mathrm{F}} \geq \alpha \|G_t\|_{\mathrm{F}}^2. \tag{185}$$

□

This result combines the radial and tangent alignment bounds into the full-direction bound used in the loss-descent and linear-convergence analysis of Appendix E.5.

E.5 Linear Convergence Rate under the PL Condition

Theorem 2 (PL recursion for an aligned first-order model)**.** *Under Assumptions 7, 8, and 9, the update* (135) *with* $0 < 2\eta\alpha\mu < 1$ *satisfies*

$$f(W_t) - f^* \le (1 - 2\eta\alpha\mu)^t (f(W_0) - f^*) + \frac{K_{\mathrm{err}}\eta}{2\alpha\mu}, \tag{186}$$

where $K_{\mathrm{err}} = \frac{L}{2}G^2$.

Proof. Consider the first-order effective update

$$W_{t+1} = W_t - \eta H_t. \tag{187}$$

By L-smoothness,

$$f(W_{t+1}) \le f(W_t) + \langle \nabla f(W_t), W_{t+1} - W_t \rangle_{\mathrm{F}} + \frac{L}{2}\|W_{t+1} - W_t\|_{\mathrm{F}}^2 \tag{188}$$

$$= f(W_t) - \eta \langle G_t, H_t \rangle_{\mathrm{F}} + \frac{L}{2}\eta^2 \|H_t\|_{\mathrm{F}}^2. \tag{189}$$

Using $\|H_t\|_{\mathrm{F}} \le G$,

$$f(W_{t+1}) \le f(W_t) - \eta \langle G_t, H_t \rangle_{\mathrm{F}} + K_{\mathrm{err}}\eta^2, \qquad K_{\mathrm{err}} = \frac{L}{2}G^2. \tag{190}$$

Using the alignment condition (137),

$$-\eta \langle G_t, H_t \rangle_{\mathrm{F}} \le -\eta\alpha \|G_t\|_{\mathrm{F}}^2. \tag{191}$$

Therefore

$$f(W_{t+1}) \le f(W_t) - \eta\alpha \|G_t\|_{\mathrm{F}}^2 + K_{\mathrm{err}}\eta^2. \tag{192}$$

Subtract f^* from both sides:

$$f(W_{t+1}) - f^* \le f(W_t) - f^* - \eta\alpha \|G_t\|_{\mathrm{F}}^2 + K_{\mathrm{err}}\eta^2. \tag{193}$$

Multiplying both sides of the PL condition (136) by 2 gives

$$\|G_t\|_{\mathrm{F}}^2 \ge 2\mu (f(W_t) - f^*). \tag{194}$$

Thus

$$f(W_{t+1}) - f^* \le f(W_t) - f^* - 2\eta\alpha\mu (f(W_t) - f^*) + K_{\mathrm{err}}\eta^2 \tag{195}$$

$$= (1 - 2\eta\alpha\mu)(f(W_t) - f^*) + K_{\mathrm{err}}\eta^2. \tag{196}$$

Let $\Delta_t = f(W_t) - f^*$ and $\rho = 2\eta\alpha\mu$. The recursion is

$$\Delta_{t+1} \le (1 - \rho)\Delta_t + K_{\mathrm{err}}\eta^2. \tag{197}$$

Expanding the first few steps makes the geometric structure explicit:

$$\Delta_1 \le (1 - \rho)\Delta_0 + K_{\mathrm{err}}\eta^2, \tag{198}$$

$$\Delta_2 \le (1 - \rho)^2\Delta_0 + (1 - \rho)K_{\mathrm{err}}\eta^2 + K_{\mathrm{err}}\eta^2, \tag{199}$$

$$\Delta_3 \le (1 - \rho)^3\Delta_0 + (1 - \rho)^2 K_{\mathrm{err}}\eta^2 + (1 - \rho)K_{\mathrm{err}}\eta^2 + K_{\mathrm{err}}\eta^2. \tag{200}$$

By induction,

$$\Delta_t \le (1 - \rho)^t \Delta_0 + K_{\mathrm{err}}\eta^2 \sum_{i=0}^{t-1} (1 - \rho)^i. \tag{201}$$

Since

$$\sum_{i=0}^{t-1} (1 - \rho)^i = \frac{1 - (1 - \rho)^t}{\rho} < \frac{1}{\rho}, \tag{202}$$

we obtain

$$\Delta_t \le (1 - \rho)^t \Delta_0 + \frac{K_{\mathrm{err}}\eta^2}{\rho}. \tag{203}$$

Substituting $\rho = 2\eta\alpha\mu$ and $\Delta_0 = f(W_0) - f^*$ gives

$$\Delta_t \leq \underbrace{(1 - 2\eta\alpha\mu)^t (f(W_0) - f^*)}_{\text{geometrically decaying term}} + \underbrace{\frac{K_{\text{err}}\eta}{2\alpha\mu}}_{\text{iteration-independent error bound}}, \tag{204}$$

which is exactly the bound stated in (186).

Since $0 < 2\eta\alpha\mu < 1$, the first term decreases by the fixed factor $1 - 2\eta\alpha\mu$ at each iteration. For a fixed stepsize, the second term is independent of t. This is the form of linear convergence up to a fixed solution level described by Karimi et al. (2016).

The second term comes from accumulating the per-step contributions $K_{\text{err}}\eta^2$. As the first term tends to zero,

$$\limsup_{t\to\infty} \big(f(W_t) - f^*\big) \leq \frac{K_{\text{err}}\eta}{2\alpha\mu}. \tag{205}$$

For a fixed smoothness constant L, update-direction norm bound G, alignment constant α between the update direction and the gradient, and Polyak–Łojasiewicz constant μ, this bound is proportional to the stepsize η. This establishes the claimed linear convergence rate: under the stated assumptions, the first-order model approaches an $\mathcal{O}(\eta)$ neighborhood of the optimal loss with contraction factor $1 - 2\eta\alpha\mu$. □