HYPERBALL MAY NOT BE A FREE LUNCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Norm-constrained optimizers perform well in large language model pretraining, but the source of their gains and their relationship with learning-rate scheduling remain unclear. We study their mechanisms and learning-rate schedules from a scale-invariance perspective. First, we derive the relationship between the scale and angular effective learning rates in preconditioned optimizers. Numerical analyses show that radial updates have a limited direct effect on one-step angular displacement. By adjusting the nominal learning rate to match the angular effective learning rate, we can largely reproduce the phase-dependent loss trajectories of Hyperball and non-Hyperball optimizers in both directions. Scale effective learning-rate alignment also yields training outcomes close to those of angular alignment. Theoretically, we establish a local dynamic regret bound under strict scale invariance, local geodesic convexity, and bounded directional gradients. Combining this analysis with a quotient-space optimization perspective, we propose Effective Learning Rate Stable–Decay (ELR-SD), a heuristic schedule that holds the scale effective learning rate constant before linearly decaying it once the gradient norm before preconditioning stabilizes. Experiments on dense, mixture-of-experts (MoE), and long-horizon pretraining tasks show that ELR-SD improves validation loss. These results further suggest that the effectiveness of long-horizon linear decay in Hyperball is linked to its control of angular step sizes, motivating the use of step-size design principles from convex optimization in large language model pretraining.

1 INTRODUCTION

In deep networks containing scale-invariant parameter blocks, parameter norms, learning rates, and the magnitudes and directions of optimizer updates jointly determine how parameter directions evolve Wan et al. (2020); Bergsma et al. (2026); Wu & He (2018); Ren et al. (2026). Recent matrix-constrained optimizers, such as MuonH, restrict matrix-valued parameters to norm spheres with prescribed radii to reduce the influence of parameter-scale variation on training dynamics Li et al. (2025); Cesista (2025); Deng et al. (2026); Wen et al. (2026a). Despite their strong empirical performance in large-scale pretraining, the source of their advantage remains unclear. In particular, under preconditioned updates, can the conventional scale effective learning rate still account for changes in parameter direction and guide learning-rate scheduling?

Previous work often argues that weight decay or norm constraints influence training primarily by regulating the scale effective learning rate Van Laarhoven (2017); Hoffer et al. (2018); Zhang et al. (2018). For scale-invariant parameters, an increase in parameter norm reduces the gradient magnitude relative to the parameter, thereby shrinking the effective directional update. This interpretation, however, typically treats the scale effective learning rate as determined by the learning rate and parameter norm, implicitly assuming approximate orthogonality between the parameter and the optimizer update Kodryan et al. (2022). This assumption need not hold for optimizers that use momentum, preconditioning, or matrix-valued update rules Bernstein & Newhouse (2024). In these cases, the angle between the parameter and the update, the update norm, and their evolution over training all affect the one-step angular displacement. Consequently, examining the parameter norm alone is insufficient to determine when the conventional scale effective learning rate accurately captures the angular motion induced by these optimizers or to directly guide learning-rate scheduling.

An empirical pattern in Hyperball training motivates our study. Under some training configurations, a Hyperball optimizer may underperform its corresponding non-Hyperball baseline early in training

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.

yet overtake it in the middle or later stages Wen et al. (2026b;a). This phase-dependent pattern provides a way to examine the scale effective learning rate interpretation: after accounting for update direction and norm, can the angular effective learning rate explain the pattern? To what extent can changing only the learning-rate schedule reproduce it?

To answer these questions, we study the scale effective learning rate through the angular displacement induced by each parameter update. First, we derive the angular effective learning rate on the unit sphere, identify the conditions under which the norm-based scale effective learning rate approximates it, and numerically evaluate the approximation error under preconditioned updates. Next, we prescribe a target trajectory for the angular effective learning rate and invert its relation to the nominal learning rate, so that the realized angular effective learning rate follows the target as closely as possible. This procedure allows us to assess how much of Hyperball's phase-dependent behavior can be reproduced by changing only the learning-rate schedule. Finally, drawing on convex optimization theory, we propose a scheduling strategy for the scale effective learning rate and evaluate its effectiveness in long-horizon pretraining tasks. Overall, our contributions are summarized as follows:

- **Conceptually**, we derive an angular effective learning rate that accounts for parameter–update geometry and identify the conditions under which the scale effective learning rate provides an accurate approximation.
- **Empirically**, we find that radial updates have a limited direct effect on angular motion in our settings. Learning-rate alignment alone largely reproduces Hyperball's phase-dependent loss trajectories, while scale effective learning-rate alignment achieves results close to angular alignment in long-horizon training.
- **Theoretically**, we establish a local dynamic regret bound for scale-invariant objectives under local geodesic convexity and bounded directional gradients. Together with a quotient-space optimization perspective, this result motivates step-size design in direction space.
- **Practically**, we propose Effective Learning Rate Stable–Decay (ELR-SD), a heuristic schedule that holds the scale effective learning rate constant before linearly decaying it once the gradient norm before preconditioning stabilizes. ELR-SD improves validation loss in dense, mixture-of-experts (MoE), and long-horizon pretraining experiments.

2 RELATED WORK

Scale invariance: Normalization makes some parameter blocks approximately scale-invariant, so training primarily changes their direction. Prior work relates weight decay to the scale effective learning rate through the parameter norm Wan et al. (2020); Van Laarhoven (2017); Hoffer et al. (2018); Zhang et al. (2018); Kosson et al. (2025); Su (2025b;c); Li & Arora (2019); Li et al. (2020). Although the instantaneous gradient is orthogonal to an exactly scale-invariant parameter, updates produced by momentum or preconditioning need not remain orthogonal.

Geometric constraints and normalization: Weight normalization separates parameter magnitude and direction Salimans & Kingma (2016). Other methods normalize or precondition matrix updates, or impose geometric constraints on parameters Li et al. (2025); Deng et al. (2026); Yuan (2026); Cesista (2025); Su (2025a; 2024). We use Hyperball to study how constraints on matrix parameters and updates shape angular displacement and training trajectories, and whether matching angular effective learning-rate trajectories can reproduce these dynamics.

Learning-rate scheduling: Existing methods prescribe nominal learning-rate curves or reduce reliance on them through schedule-free optimization Wen et al. (2025); Defazio et al. (2024); Defazio (2026); Li et al. (2026). For scale-invariant parameters, angular displacement also depends on the evolving geometry of parameters and updates. We study how to match a target angular effective learning-rate trajectory by scheduling the nominal learning rate, and use a convex-optimization step-size rule based on the scale effective learning rate to improve long-horizon pretraining.

Concurrent work: Two concurrent studies examine how the scale effective learning rate shapes large-model pretraining dynamics. ELR-MPL models loss using its cumulative value Zhou et al. (2026), while ELR-FSL studies alignment and scaling-law prediction across training settings Liu et al. (2026). Both studies overlap with our work in explaining training dynamics. Our work focuses

more on non-orthogonal preconditioned updates, the distinction between scale and angular effective learning rates, and convex-optimization-based scheduling for long-horizon pretraining.

3 PRELIMINARIES

This section briefly introduces the scale effective learning rate, the angular effective learning rate, and the update rule of Hyperball-style optimizers.

3.1 UPDATE RULE AND SCALE INVARIANCE

Consider a matrix parameter block W_t analyzed independently. Let U_t denote the update produced by the optimizer at step t, including momentum, preconditioning, and other internal transformations, but excluding the external learning rate. A general update with decoupled weight decay is:

$$W_{t+1} = \alpha_t W_t - \eta_t U_t, \qquad \alpha_t = 1 - \eta_t \lambda_t, \tag{1}$$

where $\eta_t > 0$ is the learning rate and $\lambda_t \geq 0$ is the weight-decay coefficient. Define the cosine similarity between the parameter and the update as $c_t = \frac{\langle W_t, U_t \rangle_F}{\|W_t\|_F \|U_t\|_F}$. For a scale-invariant parameter, the loss satisfies $\mathcal{L}(\rho W) = \mathcal{L}(W)$ for all $\rho > 0$, which implies:

$$\langle W, \nabla_W \mathcal{L}(W) \rangle_F = 0, \qquad \nabla_W \mathcal{L}(\rho W) = \frac{1}{\rho} \nabla_W \mathcal{L}(W). \tag{2}$$

When U_t is the instantaneous gradient, $c_t = 0$, which forms the basis of scale effective learning-rate analysis. We define the scale effective learning rate as $\eta_{\text{eff},t} = \frac{\eta_t}{\alpha_t \|W_t\|_F}$. For updates transformed by momentum, preconditioning, or other operations, however, c_t need not be zero.

3.2 ANGULAR EFFECTIVE LEARNING RATE

We measure the change in parameter direction by the angular displacement between consecutive parameter states:

$$\Delta\phi_t = \angle(W_t, W_{t+1}). \tag{3}$$

For the update rule in equation 1, if $\alpha_t \|W_t\|_F - \eta_t c_t \|U_t\|_F > 0$, then

$$\tan(\Delta\phi_t) = \frac{\eta_t \|U_t\|_F \sqrt{1 - c_t^2}}{\alpha_t \|W_t\|_F - \eta_t c_t \|U_t\|_F}. \tag{4}$$

When the one-step angular displacement is small, $\tan(\Delta\phi_t) \approx \Delta\phi_t$. Accordingly, we define the angular effective learning rate as

$$\eta^{\phi}_{\text{eff},t} \triangleq \frac{\Delta\phi_t}{\|U_t\|_F} \approx \frac{\eta_t \sqrt{1 - c_t^2}}{\alpha_t \|W_t\|_F - \eta_t c_t \|U_t\|_F}. \tag{5}$$

When $c_t = 0$, this reduces to the conventional expression for the scale effective learning rate:

$$\eta^{\phi}_{\text{eff},t} \approx \eta_{\text{eff},t} = \frac{\eta_t}{\alpha_t \|W_t\|_F}. \tag{6}$$

3.3 HYPERBALL UPDATE

Let the radius of the constraint sphere be $R = \|W_0\|_F$, and assume that $\|W_t\|_F = R$. Hyperball rescales the optimizer update U_t to have the same norm as the parameter, then projects the updated parameter back onto the sphere of radius R. Thus,

$$W_{t+1} = \frac{\|W_t\|_F}{\left\| W_t - \eta_t \frac{\|W_t\|_F}{\|U_t\|_F} U_t \right\|_F} W_t - \frac{\eta_t \|W_t\|_F}{\left\| W_t - \eta_t \frac{\|W_t\|_F}{\|U_t\|_F} U_t \right\|_F} \frac{\|W_t\|_F}{\|U_t\|_F} U_t. \tag{7}$$

Define the state-dependent projection coefficient:

$$\alpha_t^{\mathrm{H}} \triangleq \frac{\|W_t\|_F}{\left\| W_t - \eta_t \frac{\|W_t\|_F}{\|U_t\|_F} U_t \right\|_F}. \tag{8}$$

The resulting Hyperball update can be viewed as a particular form of weight decay and gradient scaling:

$$W_{t+1} = \alpha_t^{\mathrm{H}} W_t - \eta_t \alpha_t^{\mathrm{H}} \frac{\|W_t\|_F}{\|U_t\|_F} U_t. \tag{9}$$

4 Do Radial Updates Provide an Additional Benefit?

The angular effective learning rate depends not only on the learning rate and parameter norm, but also on the angle between the parameter and the optimizer update and on the update norm. This section examines whether radial updates are the primary cause of MuonH's phase-dependent training behavior. One possible explanation is that, when $c_t < 0$, the optimizer update increases the parameter norm; Hyperball projects the parameter back onto a sphere of fixed radius, thereby suppressing this radial change and potentially altering the optimization speed at different stages of training.

Radial updates may act in two ways: they may directly affect the angular displacement of the current step, or they may change the parameter and optimizer states and thereby indirectly affect subsequent updates. We first analyze their direct effect on one-step angular displacement and then use single-factor sensitivity calculations to compare the effects of radial and other factors on the angular effective learning rate.

By equation 4, the angular displacement at the current step satisfies:

$$\tan(\Delta\phi_t) = \frac{\eta_t \sqrt{1-c_t^2}\|U_t\|_F}{\alpha_t\|W_t\|_F - \eta_t c_t \|U_t\|_F}. \tag{10}$$

This expression separates two effects of the radial component. First, c_t appears in the numerator: as $|c_t|$ increases, the tangential magnitude $\|U_t\|_F\sqrt{1-c_t^2}$ decreases, leaving less of the update available to rotate the parameter. Second, c_t appears in the denominator: when $c_t < 0$, the denominator increases and the angular displacement is reduced further. Thus, a negative radial component suppresses the one-step angular motion, although the magnitude of this effect depends on the relative scales of η_t, $\|W_t\|_F$, and $\|U_t\|_F$. To analyze how these factors jointly evolve during training, treat the step index as a continuous variable and differentiate $\log \eta_{\mathrm{eff},t}^{\phi}$. This gives

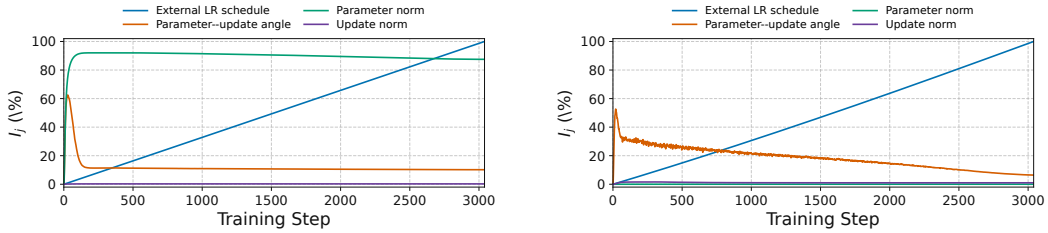
$$\frac{\mathrm{d}\log \eta_{\mathrm{eff},t}^{\phi}}{\mathrm{d}t} = \frac{\dot\eta_t}{\eta_t} - \frac{c_t\dot c_t}{1-c_t^2} - \frac{\dot\alpha_t\|W_t\|_F + \alpha_t\frac{\mathrm{d}\|W_t\|_F}{\mathrm{d}t} - \dot\eta_t\|U_t\|_F c_t - \eta_t c_t \frac{\mathrm{d}\|U_t\|_F}{\mathrm{d}t} - \eta_t\|U_t\|_F\dot c_t}{\alpha_t\|W_t\|_F - \eta_t\|U_t\|_F c_t}. \tag{11}$$

The equation 11 decomposes the evolution of the angular effective learning rate into four sources: the external learning-rate schedule, changes in the parameter–update angle, changes in the parameter norm, and changes in the update norm. As shown in Fig. 1, we further use equation 11 to quantify the contribution of each factor to the evolution of the angular effective learning rate, using

$$I_j(t) = 100\left(1 - \frac{\eta_{\mathrm{eff},t}^{\phi,(j)}}{\eta_{\mathrm{eff},t}^{\phi,(j-1)}}\right)\% \tag{12}$$

as the metric, where $\eta_{\mathrm{eff},t}^{\phi,(j)}$ denotes the angular effective learning rate at the jth stage of the single-factor comparison. Under the training configurations considered here, c_t changes substantially at the beginning of training and then enters a slowly varying negative regime. During this later regime, the terms involving c_t make only a limited instantaneous contribution to equation 11; the overall trend is governed primarily by the external learning-rate schedule and the evolution of the parameter norm.

These results do not support the hypothesis that MuonH's two-stage training dynamics are driven primarily by the removal of radial updates. Moreover, after aligning the angular effective learning-rate trajectories of MuonWD and MuonH, the resulting nominal learning rates differ only slightly in the later stages of training. The next section examines an alternative mechanism: Hyperball may act chiefly by fixing the scales of the parameter and update, thereby inducing a state-dependent effective-learning-rate schedule.



(a) (b)

Figure 1: The decline in learning rate caused by different factors. (a) MuonWD. (b) MuonH.

5 No Free Lunch: Hyperball as an Implicit Learning-Rate Scheduler

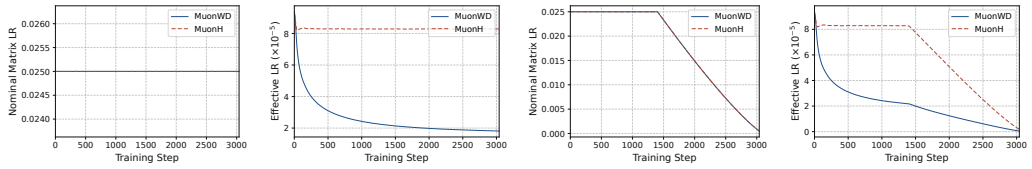
Section 4 finds that radial updates have a limited direct effect on the one-step angular displacement, motivating a step-size explanation for MuonH's slow-then-fast behavior. A related pattern appears when comparing AdamW with Adam: under the same nominal learning-rate schedule, AdamW can start with a higher loss yet finish with a lower one Wen et al. (2025). By fixing parameter norms and normalizing updates, Hyperball makes its angular effective learning rate approximately track the nominal schedule after the transient in c_t, rather than decline with parameter-norm growth (Wen et al., 2026a; Kosson et al., 2025). We next test whether aligning angular effective learning-rate trajectories can reproduce this phase-dependent behavior.

To test whether the slower initial convergence of MuonH is attributable to its larger effective step size, we conduct twopre-experiments:

1. We visualize and compare the effective-learning-rate trajectories of MuonWD and MuonH.
2. We test whether the validation-loss dynamics of Hyperball and non-Hyperball optimizers can be transformed into one another by changing only the learning-rate schedule.

5.1 Empirical evolution of the effective learning rate under different schedules

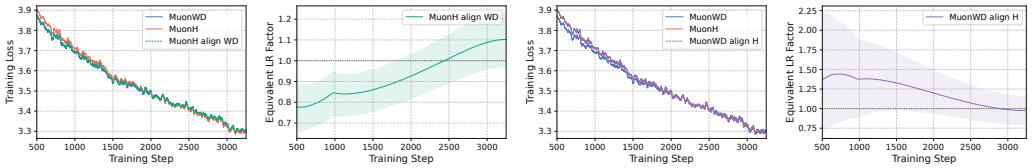
The effective learning rates of non-Hyperball optimizers such as AdamW and MuonWD have been characterized by norm-based analyses. Section 4 extends this picture to the non-orthogonal regime $c_t < 0$. Here we directly compare MuonWD and MuonH. The purpose is to establish two empirical facts: MuonH operates at a larger angular effective learning rate in the configurations considered, and its effective learning rate remains approximately proportional to the externally specified learning rate.



(a) (b) (c) (d)

Figure 2: Learning-rate schedules and the resulting effective learning rates. (a) Constant learning-rate schedule. (b) Effective-learning-rate comparison under the constant schedule. (c) Constant-then-decay schedule. (d) Effective-learning-rate comparison under the constant-then-decay schedule.

Figure 2 shows the learning-rate schedules and their corresponding effective-learning-rate trajectories. Under a constant base learning rate, MuonWD's effective learning rate decays rapidly as its parameter norm grows. MuonH, in contrast, maintains an approximately constant effective learning rate after an initial transient caused by the abrupt evolution of c_t. Under a linearly decaying base learning rate, MuonH's effective learning rate remains approximately proportional to that schedule.



(a) (b) (c) (d)

Figure 3: Achieving Mutual Alignment between Hyperball and Non-Hyperball Dynamics in the 4K-Step nanoGPT Setting. (a) Training loss when MuonH is aligned with MuonWD. (b) MuonH learning-rate alignment factors during this experiment. (c) Training loss when MuonWD is aligned with MuonH. (d) MuonWD learning-rate alignment factors during the reverse experiment. Shaded regions in (b) and (d) show variation across layers.

5.2 Transforming the Dynamics of Hyperball and non-Hyperball Optimizers

We next test whether the effective learning rate is the principal difference between Hyperball and non-Hyperball optimizers. Because fixed parameter and update norms can change multiple aspects of a high-dimensional optimization trajectory, observational comparisons alone cannot isolate their contribution. We therefore perform a controlled ablation in which the optimizer is held fixed and only its learning rate is changed so as to match the target angular effective learning rate at every step. Denote the effective learning rates of the non-Hyperball and Hyperball optimizers under the reference schedule by $\eta^{\phi}_{\mathrm{eff},t}$ and $\eta^{H,\phi}_{\mathrm{eff},t}$, respectively. To align MuonWD with MuonH, we proceed as follows:

1. Run MuonH and record its angular effective learning rate $\eta^{H,\phi}_{\mathrm{eff},t}$ at every step.
2. Before each MuonWD update, record c_t, W_t, and U_t, and solve

$$\eta^{H,\phi}_{\mathrm{eff},t} \approx \frac{\eta_t \sqrt{1-c_t^2}}{\alpha_t \|W_t\|_F - c_t \eta_t \|U_t\|_F} \tag{13}$$

   for the step-specific learning rate η_t.
3. Use the resulting η_t for the current MuonWD update.

We also apply the same learning-rate alignment procedure in the reverse direction. Figure 3 shows that adjusting only the nominal learning rate to match the angular effective learning rate allows the training-loss trajectories of MuonH and MuonWD to largely reproduce one another. Figure 4 shows that, on the long-horizon task, scale effective learning-rate alignment performs slightly worse than angular effective learning-rate alignment but yields comparable training loss. Thus, the scale effective learning rate can guide learning-rate scheduling for loss reduction nearly as effectively as the angular effective learning rate. Together, these results suggest that Hyperball alone may not be a free lunch: improving performance requires a more effective learning-rate schedule.

Because linear learning-rate decay schedules from convex optimization do not directly apply here Su (2025d; 2026a;b), we adapt linear decay schedules Defazio et al. (2024; 2023) to design a two-stage schedule for the scale effective learning rate. We hold it constant; once the norm of the gradient before preconditioning stabilizes, we decay it linearly. Scheduling the scale effective learning rate directly reduces the influence of changes in parameter scale on the actual step size. The gradient-norm stability test is described in Algorithm 1. We refer to this method as ELR-SD and discuss its motivation in Section 7 and Appendix B.

6 Experiments

We evaluate ELR-SD, a heuristic scheduling strategy based on the scale effective learning rate, in language-model pretraining. Our experiments use nanoGPT with FineWeb, C4, and OpenWebText data. We first compare optimizer–schedule configurations on dense and mixture-of-experts (MoE) models, and then evaluate ELR-SD against linear learning-rate decay with MuonH over a longer training horizon. Validation loss serves as the evaluation metric, with lower values indicating better performance.

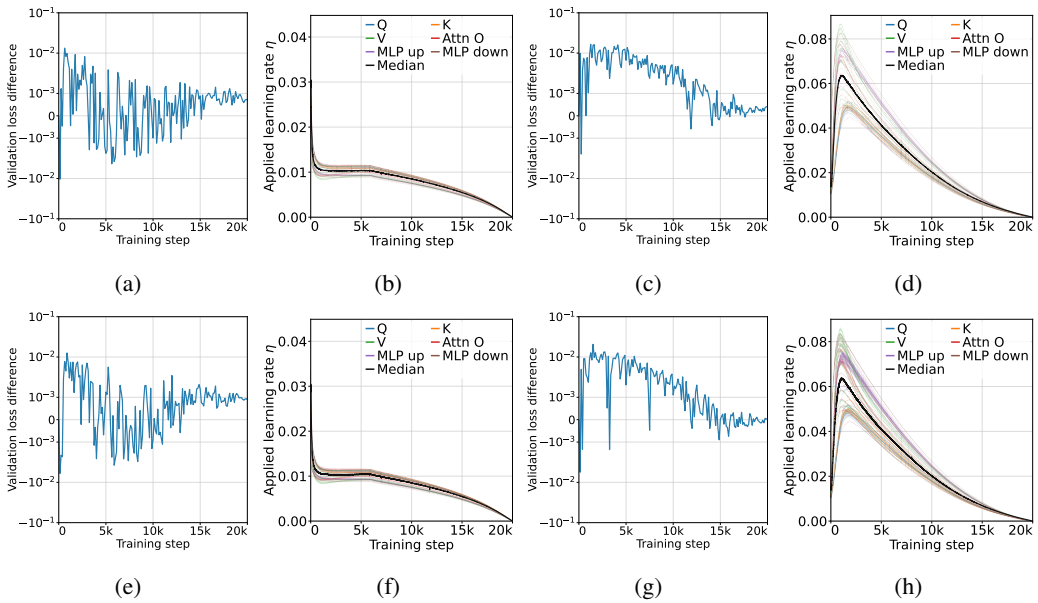


Figure 4: Mutual alignment of MuonH and MuonWD in the 20K-step nanoGPT setting. Panel pairs (a,b) and (c,d) align MuonH with MuonWD and MuonWD with MuonH, respectively, using the scale effective learning rate. Pairs (e,f) and (g,h) repeat these directions using the angular effective learning rate. In each pair, the first panel shows validation-loss error, and the second shows the resulting layerwise nominal learning rates.

Algorithm 1 Online gradient-stability trigger for learning-rate switching

Require: Window length $O \geq 2$, threshold $\tau > 0$, and gradients $g_t = (g_{p,t})_{p \in \mathcal{M}}$ observed before preconditioning.

Ensure: First switching update s, if one occurs.

for $t = 0, 1, 2, \ldots$ **do**

  $x_t \leftarrow \log\Big(\sum_{p \in \mathcal{M}} \|g_{p,t}\|_F^2\Big)^{1/2}$;

  if $t \geq 2O - 1$ **then**

    $m_{\text{old}} \leftarrow \text{median}\{x_i\}_{i=t-2O+1}^{t-O}$, $m_{\text{new}} \leftarrow \text{median}\{x_i\}_{i=t-O+1}^{t}$;

    $D_t \leftarrow |\exp(m_{\text{new}} - m_{\text{old}}) - 1|$, $b_t \leftarrow \dfrac{\sum_{j=0}^{O-1} \left(j - \frac{O-1}{2}\right) x_{t-O+1+j}}{\sum_{j=0}^{O-1} \left(j - \frac{O-1}{2}\right)^2}$;

    $S_t \leftarrow (O-1)|b_t|$, $a_t \leftarrow \mathbf{1}\{D_t \leq \tau \ \wedge \ S_t \leq \tau\}$;

    if $t \geq 2O + 1$ **and** $a_t a_{t-1} a_{t-2} = 1$ **then**

      $s \leftarrow t$

      return s ▷ Switch the learning-rate rule for update t

    end if

  end if

end for

6.1 Dense and MoE Pretraining

We consider two training settings: a 130M-parameter dense model trained on 10B tokens and a 1B/0.3B MoE model trained on 20B tokens. In each setting, we compare three optimizer–schedule configurations: AdamW with warmup–stable–decay (WSD), AdamC with ScheduleFree++, and AdamH with ELR-SD.

We conduct 10 and 4 peak-learning-rate search trials for the dense and MoE settings, respectively. For ScheduleFree++, we separately tune the weight-decay coefficient and multiplier. Table 1 reports the selected hyperparameters and validation losses averaged over random seeds 0 and 1.

AdamH with ELR-SD achieves the lowest mean validation loss among the evaluated configurations in both settings. For the dense model, it reduces the loss from 3.159 to 3.126 relative to AdamW with WSD, and from 3.234 to 3.126 relative to AdamC with ScheduleFree++, corresponding to absolute

Table 1: Validation loss for different optimizer–schedule configurations. The dense model has 130M parameters and is trained on 10B tokens; the 1B/0.3B MoE model is trained on 20B tokens. Results are averaged over seeds 0 and 1. Boldface denotes the lowest mean loss within each model setting.

Model	Optimizer	Schedule	WD	Peak LR	Multiplier	Val. loss
Dense	AdamW	WSD	0.1	0.0022	–	3.159
Dense	AdamC	ScheduleFree+	20	–	1	3.234
Dense	AdamH	ELR-SD	0	0.01	–	**3.126**
MoE	AdamW	WSD	0.1	0.0007	–	2.808
MoE	AdamC	ScheduleFree+	5	–	0.75	2.805
MoE	AdamH	ELR-SD	0	0.005	–	**2.778**

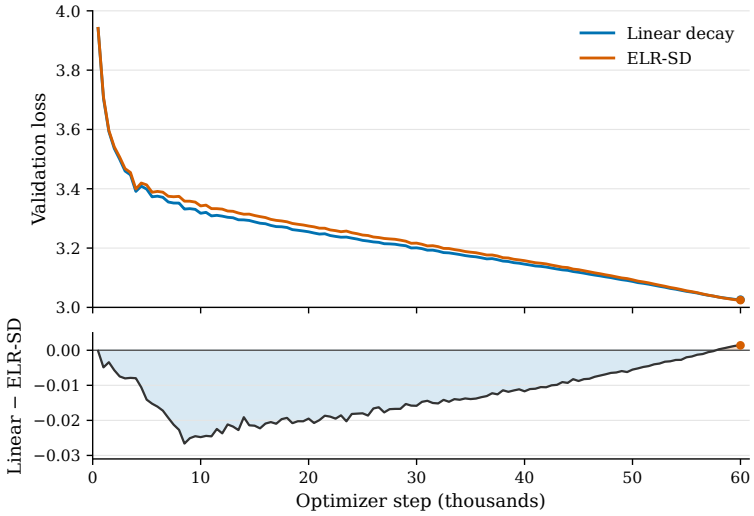


Figure 5: Validation-loss trajectories for MuonH with linear learning-rate decay and ELR-SD over 60K optimizer steps. The upper panel shows validation loss; the lower panel shows the pointwise difference, Linear − ELR-SD. Negative values indicate lower validation loss for linear decay, and positive values indicate lower validation loss for ELR-SD.

reductions of 0.033 and 0.108, respectively. For the MoE model, the corresponding reductions are 0.030 and 0.027. Because the optimizer and schedule vary jointly across configurations, these results characterize the performance of the complete optimizer–schedule combinations.

6.2 Long-Horizon Pretraining with MuonH

To examine scheduling choices with a fixed optimizer, we compare ELR-SD with linear learning-rate decay using MuonH on a 130M-parameter dense model trained on 30B tokens. Both configurations use a nominal peak learning rate of 0.06.

Figure 5 shows the validation-loss trajectories over 60K optimizer steps, together with their pointwise difference. The two schedules remain close for most of training: linear decay has a small advantage through much of the middle stage, while ELR-SD catches up near the end and finishes slightly lower. Consistent with the endpoint in the figure, Table 2 reports a validation loss of 3.024528 for ELR-SD and 3.025933 for linear decay. The absolute reduction of 0.001405 represents a modest improvement in the reported validation loss for this long-horizon setting.

7 Optimization Theory

This section starts from online regret bounds and examines the regret properties of scale-invariant objectives after alignment in angular space. We first define the parameter direction and the unit Frobenius sphere as $\overline{W} := W/\|W\|_F$ and $\mathbb{S}_F := \{W : \|W\|_F = 1\}$. For $\overline{W} \in \mathbb{S}_F$, the tangent space is $T_{\overline{W}}\mathbb{S}_F = \{V : \langle \overline{W}, V\rangle_F = 0\}$. Let $\Omega \subset \mathbb{S}_F$ denote a local directional domain. All Riemannian derivatives of $\mathcal{L}_t$ below refer to derivatives of its restriction to $\mathbb{S}_F$.

Table 2: MuonH schedule comparison on the 130M-parameter dense model trained on 30B tokens. Both schedules use a nominal peak learning rate of 0.06. Boldface denotes the lower validation loss.

Model	Optimizer	Schedule	Peak LR	Val. loss
Dense	MuonH	Linear decay	0.06	3.025933
Dense	MuonH	ELR-SD	0.06	**3.024528**

Assumption 7.1 (Scale invariance)**.** For each $t = 1, \dots, T$, the loss satisfies:

$$\mathcal{L}_t(\rho W) = \mathcal{L}_t(W) = \mathcal{L}_t(\overline{W}), \qquad W \neq 0, \quad \rho > 0. \tag{14}$$

There exists a known constant $\gamma > 0$ such that the output parameters satisfy $\|W_t\|_F \geq \gamma$.

Assumption 7.2 (Local tangential convexity)**.** Each $\mathcal{L}_t$ is twice continuously differentiable on an open neighborhood of Ω in the nonzero parameter space. Its restriction to $\mathbb{S}_F$ satisfies:

$$\mathrm{Hess}_{\mathbb{S}_F} \mathcal{L}_t(\overline{W})[V, V] \geq 0, \qquad \overline{W} \in \Omega, \quad V \in T_{\overline{W}}\mathbb{S}_F, \quad t = 1, \dots, T. \tag{15}$$

Assumption 7.3 (Bounded directional gradients)**.** There exist a fixed direction $W_\circ \in \mathbb{S}_F$ and known constants $0 < \theta_\Omega < \pi/4$ and $G > 0$ such that, on $\Omega = \left\{ \overline{W} \in \mathbb{S}_F : \arccos\langle \overline{W}, \overline{W}_\circ \rangle_F \leq \theta_\Omega \right\}$, the following bound holds:

$$\left\| \mathrm{grad}_{\mathbb{S}_F} \mathcal{L}_t(\overline{W}) \right\|_F \leq G, \qquad \overline{W} \in \Omega, \quad t = 1, \dots, T. \tag{16}$$

Under Assumption 7.3, we measure directional distances using the equivalent representatives $\gamma\overline{W}$ on the sphere of radius γ:

$$d_\gamma(W, W') := \gamma\, d_{\mathbb{S}_F}(\overline{W}, \overline{W}') = \gamma \arccos\langle \overline{W}, \overline{W}' \rangle_F, \qquad D := 2\gamma\theta_\Omega. \tag{17}$$

These conditions yield a dynamic regret guarantee of the type obtained in convex optimization for tangential updates, even when the neural-network objective is nonconvex in the original parameter space. Specifically, we have the following theorem.

Theorem 7.1 (Dynamic regret under local tangential convexity)**.** *Under Assumptions 7.1–7.3, for a given time horizon T, there exists an online algorithm based on spherical projection whose output parameters $\{W_t\}_{t=1}^T$ satisfy $\overline{W}_t \in \Omega$ and $\|W_t\|_F \geq \gamma$.*

For any nonzero comparator sequence $W_{1:T}^{\mathrm{ref}}$ whose directions all lie in Ω, the dynamic regret satisfies

$$\sum_{t=1}^{T} \mathcal{L}_t(W_t) - \sum_{t=1}^{T} \mathcal{L}_t(W_t^{\mathrm{ref}}) \leq \frac{35\sqrt{2}\, G}{8\gamma} \sqrt{T \left(D^2 + D\, P_T(W_{1:T}^{\mathrm{ref}}) \right)}, \tag{18}$$

where $P_T(W_{1:T}^{\mathrm{ref}}) := \sum_{t=2}^{T} d_\gamma \left(W_t^{\mathrm{ref}}, W_{t-1}^{\mathrm{ref}} \right)$ .The proof is provided in Appendix C.

This result shows that controlling directional updates provides a local dynamic regret guarantee when the losses satisfy local geodesic convexity and bounded directional gradients on a shared directional domain. It provides a theoretical foundation for designing step sizes in directional space and aligning the angular effective learning rate through the nominal learning rate. Appendix B further discusses an alternative update approach from a quotient-space perspective. Building on these theoretical foundations, we design a long-horizon linear decay schedule for the scale effective learning rate to match the theoretically optimal linear step-size decay schedule in convex optimization.

8 Conclusion

We analyze the relationship between the scale and angular effective learning rates in preconditioned optimizers. Experiments show that learning-rate alignment alone can largely reproduce Hyperball's phase-dependent training behavior; in long-horizon tasks, scale effective learning-rate alignment also achieves results close to angular effective learning-rate alignment. Drawing on convex regret bounds and quotient-space optimization, we propose the heuristic schedule ELR-SD, which improves validation loss in dense, MoE, and long-horizon pretraining experiments. These results suggest that Hyperball may not be a free lunch: further gains require a more suitable learning-rate schedule. Due to space constraints, we discuss limitations and future work in Appendix D.

AI USE STATEMENT

We used ChatGPT to assist with language polishing and AI tools to assist with coding. Both lead authors reviewed the AI-assisted work.

ETHICS STATEMENT

This work concerns large language model pretraining. Its primary impact is to accelerate research in this area. We do not identify additional ethical concerns.

REPRODUCIBILITY STATEMENT

We release the code in an anonymous repository: `https://anonymous.4open.science/r/hyperball-may-not-be-a-free-lunch-622E/README.md`.

REFERENCES

Shane Bergsma, Nolan Dey, Gurpreet Gosal, Gavia Gray, Daria Soboleva, and Joel Hestness. Power lines: Scaling laws for weight decay and batch size in llm pre-training. *Advances in Neural Information Processing Systems*, 38:125153–125188, 2026.

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

F. L. Cesista. Heuristic solutions for steepest descent on the stiefel manifold, 2025. URL `https://leloykun.github.io/ponder/steepest-descent-stiefel/`.

Huanran Chen, Huaqing Zhang, Xiao Li, Yinpeng Dong, Ke Shen, and Jun Zhu. Nexus: Same pretraining loss, better downstream generalization via common minima. *arXiv preprint arXiv:2604.09258*, 2026.

Aaron Defazio. Schedulefree+: Scaling learning-rate-free & schedule-free learning to large language models. *arXiv preprint arXiv:2605.19095*, 2026.

Aaron Defazio, Ashok Cutkosky, Harsh Mehta, and Konstantin Mishchenko. Optimal linear decay learning rate schedules and further refinements. *arXiv preprint arXiv:2310.07831*, 2023.

Aaron Defazio, Xingyu Yang, Harsh Mehta, Konstantin Mishchenko, Ahmed Khaled, and Ashok Cutkosky. The road less scheduled. *Advances in Neural Information Processing Systems*, 37: 9974–10007, 2024.

Shenyang Deng, Zhuoli Ouyang, Tianyu Pang, Zihang Liu, Ruochen Jin, Shuhua Yu, and Yaoqing Yang. Rmnp: Row-momentum normalized preconditioning for scalable matrix-based optimization. *arXiv preprint arXiv:2603.20527*, 2026.

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

Maxim Kodryan, Ekaterina Lobacheva, Maksim Nakhodnov, and Dmitry Vetrov. Training scale-invariant neural networks on the sphere can happen in three regimes. *Advances in Neural Information Processing Systems*, 35:14058–14070, 2022.

A. Kosson, J. Welborn, Y. Liu, M. Jaggi, and X. Chen. Weight decay may matter more than μP for learning rate transfer in practice. *arXiv preprint arXiv:2510.19093*, 2025. URL `https://arxiv.org/abs/2510.19093`.

Binghui Li, Zilin Wang, Fengling Chen, Shiyang Zhao, Ruiheng Zheng, and Lei Wu. Optimal learning-rate schedules under functional scaling laws: Power decay and warmup-stable-decay. *arXiv preprint arXiv:2602.06797*, 2026.

Z. Li, L. Liu, C. Liang, W. Chen, and T. Zhao. NorMuon: Making Muon more efficient and scalable. *arXiv preprint arXiv:2510.05491*, 2025. URL https://arxiv.org/abs/2510.05491.

Zhiyuan Li and Sanjeev Arora. An exponential learning rate schedule for deep learning. *arXiv preprint arXiv:1910.07454*, 2019.

Zhiyuan Li, Kaifeng Lyu, and Sanjeev Arora. Reconciling modern deep learning with traditional optimization analyses: The intrinsic learning rate. *Advances in Neural Information Processing Systems*, 33:14544–14555, 2020.

Zihan Liu, Ruiheng Zheng, Shaobo Zhang, Changxin Tian, Kunlong Chen, Zhiqiang Zhang, and Lei Wu. Effective learning rate governs loss dynamics in language model pretraining. *arXiv preprint arXiv:2608.24814*, 2026.

Liliang Ren, Yang Liu, Yelong Shen, and Weizhu Chen. Rethinking language model scaling under transferable hypersphere optimization. *arXiv preprint arXiv:2603.28743*, 2026.

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*, 2016. URL https://arxiv.org/abs/1602.07868.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost, 2018. URL https://arxiv.org/abs/1804.04235.

J. Su. Thinking about spectral norm gradient and spectral weight decay, 2024. URL https://kexue.fm/archives/10648.

J. Su. Muon on the stiefel manifold, 2025a. URL https://kexue.fm/archives/11221.

J. Su. AdamW weight RMS asymptotics (part I), 2025b. URL https://kexue.fm/archives/11307.

J. Su. AdamW weight RMS asymptotics (part II), 2025c. URL https://kexue.fm/archives/11404.

Jianlin Su. Making model training more scientific (iv): A new identity, a new learning rate, December 2025d. URL https://spaces.ac.cn/archives/11494. In Chinese.

Jianlin Su. Making model training more scientific (v): Fine-tuning the learning rate based on gradients, January 2026a. URL https://spaces.ac.cn/archives/11530. In Chinese.

Jianlin Su. Making model training more scientific (vi): An ingenious top-down construction, January 2026b. URL https://spaces.ac.cn/archives/11540. In Chinese.

Twan Van Laarhoven. L2 regularization versus batch and weight normalization. *arXiv preprint arXiv:1706.05350*, 2017.

Ruosi Wan, Zhanxing Zhu, Xiangyu Zhang, and Jian Sun. Spherical motion dynamics: Learning dynamics of neural network with normalization, weight decay, and sgd. *arXiv preprint arXiv:2006.08419*, 2020.

Kaiyue Wen, Zhiyuan Li, Jason Wang, David Hall, Percy Liang, and Tengyu Ma. Understanding warmup-stable-decay learning rates: A river valley loss landscape view. In *International Conference on Learning Representations*, volume 2025, pp. 42840–42885, 2025.

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*, 2026a.

Kaiyue Wen, David Hall, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them. In *International Conference on Learning Representations*, volume 2026, pp. 144731–144838, 2026b.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

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

Ganzhao Yuan. Optmuon: Closed-loop orthogonalized momentum methods for stochastic optimization with zero-noise optimality. *arXiv preprint arXiv:2606.08783*, 2026.

Guodong Zhang, Chaoqi Wang, Bowen Xu, and Roger Grosse. Three mechanisms of weight decay regularization. *arXiv preprint arXiv:1810.12281*, 2018.

Zhanpeng Zhou, Wenjie Zhou, Yufei Gu, Juqiu Wang, Guanwei Zhang, Ruobing Xie, Xianbiao Qi, Yushi Bai, Kaifeng Lyu, and Shunyu Yao. From lr to elr: A better heuristic for pretraining dynamics, 2026. URL https://hy.tencent.ai/research/elr.

A Appendix

This appendix presents the quotient-space optimization motivation for ELR-SD, provides proofs of the theorems in the main text, and includes additional discussion of the work and its limitations.

B A Heuristic Motivation from Quotient-Space Optimization

This section examines how parameter norms affect directional optimization progress in scale-invariant problems. Through theoretical examples and numerical experiments, we motivate learning-rate schedule design and relate this perspective to the scale control imposed by weight decay and Hyperball.

For a scale-invariant parameter block w, the loss satisfies

$$\mathcal{L}(\alpha w) = \mathcal{L}(w), \qquad \forall \alpha > 0. \tag{19}$$

Parameters with the same direction but different norms therefore represent the same objective-function state. The optimization problem can be formulated over equivalence classes $[w] = \{\alpha w : \alpha > 0\}$, with unit-norm directional variables serving as representatives.

This structure makes the interpretation of Euclidean gradient norms and nominal learning rates dependent on parameter scale. A smaller gradient may reflect progress in directional optimization or an increase in the parameter norm. The Euclidean gradient norm alone is therefore insufficient to distinguish these cases.

A classical scale-invariant optimization problem arises from the variational formulation of symmetric eigenvalue problems. Given a real symmetric matrix $A \in \mathbb{R}^{n\times n}$, consider Rayleigh quotient minimization:

$$\min_{y\in\mathbb{R}^n\setminus\{0\}} \rho(y) = \frac{y^\top A y}{y^\top y}. \tag{20}$$

The objective satisfies

$$\rho(\alpha y) = \rho(y), \qquad \forall \alpha > 0. \tag{21}$$

Its effective optimization variable is thus the parameter direction. The following results show that norm growth can attenuate the directional step induced by Euclidean GD, whereas Euclidean Newton updates may move entirely along the scaling direction.

Lemma B.1 (Directional step-size attenuation in GD for the scale-invariant Rayleigh quotient)**.** *Let $y_t = R_t u_t$, where $R_t = \|y_t\|_2$ and $\|u_t\|_2 = 1$. Ordinary Euclidean GD induces the following effective step size on the directional variable u_t:*

$$\eta_t^{\text{eff}} = \frac{\eta_t}{R_t^2}. \tag{22}$$

For a fixed nominal learning rate, consecutive directional effective step sizes satisfy

$$\frac{\eta_{t+1}^{\text{eff}}}{\eta_t^{\text{eff}}} = \left(\frac{R_t}{R_{t+1}}\right)^2. \tag{23}$$

Hence, an increase in the parameter norm reduces the directional effective step size.

Proposition B.2 (Absence of objective descent in Newton's method for the scale-invariant Rayleigh quotient)**.** *Apply full-step Euclidean Newton updates to the Rayleigh quotient $\rho(y)$. If $\rho(y_t)$ is not an eigenvalue of A, then*

$$y_{t+1} = 2y_t. \tag{24}$$

This update changes only the parameter scale, so

$$\rho(y_{t+1}) = \rho(y_t), \tag{25}$$

and produces no objective descent.

Lemma B.1 and Proposition B.2 show that changes in parameter space need not correspond to directional optimization progress. For a differentiable scale-invariant objective,

$$\nabla\mathcal{L}(Ru) = \frac{1}{R}\nabla\mathcal{L}(u). \tag{26}$$

Increasing the parameter norm can therefore reduce the Euclidean gradient norm even when the directional variable remains far from a stationary point. A region with a large parameter norm and a small gradient need not indicate that directional optimization is complete. Interpreting Euclidean gradient convergence or a gradient-based local regret bound consequently requires accounting for parameter scale or adopting a stationarity measure in directional space.

A common approach is to optimize in a quotient space that removes the scaling degree of freedom, for example by using directional variables on the unit sphere. We next compare direct decay of the nominal step size with a directional step-size schedule that compensates for parameter scale.

Consider the scale-invariant function

$$g(x, y) = \frac{x^4 - 6x^2y^2 + y^4}{(x^2 + y^2)^2}, \qquad (x, y) \neq (0, 0). \tag{27}$$

Writing $x = R\cos\theta$ and $y = R\sin\theta$ gives

$$g(x, y) = \cos(4\theta), \qquad \|\nabla g(x, y)\|_2 = \frac{4|\sin(4\theta)|}{R}. \tag{28}$$

The objective depends only on direction, whereas the Euclidean gradient norm decreases as the radius grows.

We compare two GD step-size schedules:

$$\eta_t = \frac{\gamma}{\sqrt{t+1}}, \qquad \eta'_t = \frac{\gamma(x_t^2 + y_t^2)}{\sqrt{t+1}}, \qquad \gamma > 0. \tag{29}$$

The first directly decays the nominal learning rate and induces a directional effective step size of $\gamma/(R_t^2\sqrt{t+1})$. The second compensates for parameter scale by multiplying the step size by R_t^2, yielding

$$\frac{\eta'_t}{R_t^2} = \frac{\gamma}{\sqrt{t+1}}. \tag{30}$$

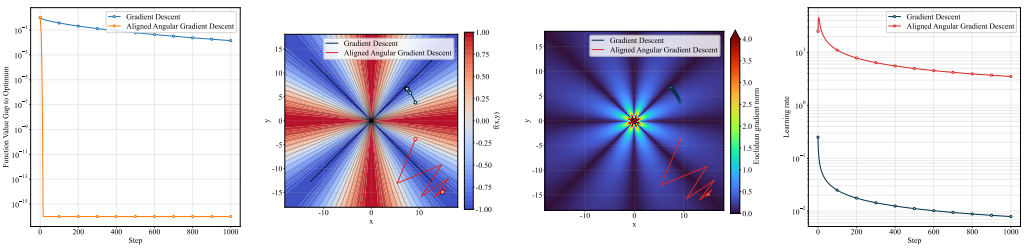
This compensation controls the directional step size without requiring the objective to be convex.

Figure 6(a,b) compares the objective values and parameter trajectories of the two methods. Ordinary GD gradually enters regions with larger parameter norms and smaller Euclidean gradients, slowing its directional updates. The compensated method maintains the prescribed directional step-size decay and approaches convergence within approximately 200 steps in this experiment. Figure 6(c) illustrates how the gradient norm of the fixed objective varies with the parameter radius, while Figure 6(d) compares the nominal learning rates required by the two methods.

The compensated method may exhibit an early spike in its nominal learning rate because η'_t depends on both parameter-norm growth and the prescribed decay curve. For an exactly scale-invariant objective, the raw gradient is orthogonal to the parameter. Let $z_t = (x_t, y_t)$ and apply GD with step size h_t. Then

$$\|z_{t+1}\|_2^2 = \|z_t\|_2^2 + h_t^2\|\nabla g(z_t)\|_2^2. \tag{31}$$

Finite tangential steps therefore also increase the parameter norm, which amplifies the nominal learning rate through scale compensation. Interpreting this spike requires examining the evolution of the directional variable as well.



(a) (b) (c) (d)

Figure 6: Numerical experiments on the scale-invariant function $g(x, y)$, comparing ordinary GD with GD using parameter-scale compensation. (a) Objective values over iterations. (b) Optimization trajectories in parameter space. (c) The effect of parameter scale on the Euclidean gradient norm. (d) Nominal learning-rate trajectories.

These numerical results show that, in the setting considered, parameter-norm growth suppresses Euclidean gradients and directional updates, slowing objective descent. Compensating for parameter scale helps maintain the prescribed directional step size. Changes in the nominal learning rate should therefore be interpreted jointly with parameter scale and directional updates.

These observations motivate designing step-size schedules for directional variables in quotient space. For scale-invariant parameters, decaying the nominal learning rate according to a theoretical schedule need not preserve the intended directional step-size trajectory. We therefore introduce parameter-scale compensation and design a heuristic scheduling strategy centered on long-horizon decay.

This perspective also offers a possible explanation for the differing performance of linear decay across optimizers. In conventional optimizers, linear-decay updates are affected by parameter scale, leading to poor performance. In Hyperball-style optimizers, the learning rate is proportional to the angular effective learning rate, allowing linear decay to perform well.

C Proof of Theorem 7.1

Theorem C.1 (Dynamic regret under local tangential convexity)**.** *Under Assumptions 7.1–7.3, for a given time horizon T, there exists an online algorithm based on spherical projection and online aggregation over multiple step sizes whose output parameter sequence $\{W_t\}_{t=1}^T$ satisfies*

$$\overline{W}_t = \frac{W_t}{\|W_t\|_F} \in \Omega, \qquad \|W_t\|_F \geq \gamma.$$

For any nonzero comparator sequence $W_{1:T}^{\mathrm{ref}} = (W_1^{\mathrm{ref}}, \ldots, W_T^{\mathrm{ref}})$ whose directions all lie in Ω, define its path length as

$$P_T(W_{1:T}^{\mathrm{ref}}) = \sum_{t=2}^{T} d_\gamma \left(W_t^{\mathrm{ref}}, W_{t-1}^{\mathrm{ref}}\right), \tag{32}$$

where d_γ denotes the geodesic distance on the directional sphere of radius γ.

Then the dynamic regret with respect to this comparator sequence satisfies

$$R_T(W_{1:T}^{\mathrm{ref}}) = \sum_{t=1}^{T} \mathcal{L}_t(W_t) - \sum_{t=1}^{T} \mathcal{L}_t(W_t^{\mathrm{ref}}) \leq \frac{35G}{8\gamma} \sqrt{T\big(D^2 + 2D\, P_T(W_{1:T}^{\mathrm{ref}})\big)}. \tag{33}$$

Proof. Let

$$X = \gamma \frac{W}{\|W\|_F}, \qquad \mathcal{K} = \gamma\Omega, \qquad \ell_t(X) = \mathcal{L}_t(X/\gamma), \qquad H := \frac{G}{\gamma}.$$

By scale invariance,

$$\mathcal{L}_t(W) = \ell_t(X), \qquad \|\operatorname{grad} \ell_t(X)\|_F = \frac{1}{\gamma} \left\|\operatorname{grad}_{\mathbb{S}_F} \mathcal{L}_t(X/\gamma)\right\|_F \leq H.$$

Moreover, Assumption 7.2 implies that ℓ_t is geodesically convex on the geodesically convex set $\mathcal{K}$. Thus, for any $X, Z \in \mathcal{K}$,

$$\ell_t(X) - \ell_t(Z) \leq - \langle \operatorname{grad} \ell_t(X), \operatorname{Log}_X Z \rangle_F . \tag{34}$$

Consider a base learner with a fixed step size η:

$$Y_t = \operatorname{Exp}_{X_t}(-\eta \operatorname{grad} \ell_t(X_t)) , \qquad X_{t+1} = \Pi_{\mathcal{K}}(Y_t).$$

Since $\rho < \pi/4$, the squared geodesic distance on the spherical cap under consideration satisfies

$$\operatorname{Hess} \frac{1}{2} d_\gamma(\,\cdot\,, Z)^2 \preceq I,$$

and projection onto the spherical cap satisfies $d_\gamma(\Pi_{\mathcal{K}}(Y), Z) \leq d_\gamma(Y, Z)$ for every $Z \in \mathcal{K}$. Combining these properties with equation 34 gives

$$\ell_t(X_t) - \ell_t(Z_t) \leq \frac{d_\gamma(X_t, Z_t)^2 - d_\gamma(X_{t+1}, Z_t)^2}{2\eta} + \frac{\eta H^2}{2}. \tag{35}$$

Let

$$P := P_T(W^{\mathrm{ref}}_{1:T}) = \sum_{t=2}^{T} d_\gamma(Z_t, Z_{t-1}), \qquad Z_t = \gamma \frac{W^{\mathrm{ref}}_t}{\|W^{\mathrm{ref}}_t\|_F}. \tag{36}$$

Since $\operatorname{diam}(\mathcal{K}) = D$,

$$\begin{aligned} &\sum_{t=1}^{T} \left[d_\gamma(X_t, Z_t)^2 - d_\gamma(X_{t+1}, Z_t)^2 \right] \\ &\quad = d_\gamma(X_1, Z_1)^2 - d_\gamma(X_{T+1}, Z_T)^2 + \sum_{t=1}^{T-1} \left[d_\gamma(X_{t+1}, Z_{t+1})^2 - d_\gamma(X_{t+1}, Z_t)^2 \right] \\ &\quad \leq D^2 + 2D \sum_{t=1}^{T-1} d_\gamma(Z_{t+1}, Z_t) = D^2 + 2DP. \end{aligned} \tag{37}$$

Summing equation 35 yields the following bound for any base learner with a fixed step size:

$$\sum_{t=1}^{T} \left[\ell_t(X_t) - \ell_t(Z_t) \right] \leq \frac{D^2 + 2DP}{2\eta} + \frac{\eta H^2 T}{2}. \tag{38}$$

Next, set

$$N = 1 + \left\lceil \frac{1}{2} \log_2 T \right\rceil , \qquad \eta_j = \frac{D}{H} \min \left\{ \frac{2^{j-1}}{\sqrt{T}}, 1 \right\}, \qquad j \in [N], \tag{39}$$

and let

$$p_{1,j} = \frac{N+1}{N\, j(j+1)}, \qquad \beta = \frac{1}{HD\sqrt{T}}. \tag{40}$$

At each round, aggregate the base learners' outputs through a geodesic convex combination with weights $p_{t,j}$, and update the weights exponentially:

$$p_{t+1,j} = \frac{p_{t,j} e^{-\beta \ell_t(X_{t,j})}}{\sum_{k=1}^{N} p_{t,k} e^{-\beta \ell_t(X_{t,k})}}. \tag{41}$$

By the geodesic convexity of ℓ_t,

$$\ell_t(X_t) \leq \sum_{j=1}^{N} p_{t,j} \ell_t(X_{t,j}). \tag{42}$$

Furthermore, $\|\operatorname{grad}\ell_t\|_F \le H$ and $\operatorname{diam}(\mathcal{K}) = D$ imply that the loss difference between any two experts at each round is at most HD. Hoeffding's inequality therefore gives, for every j,

$$\sum_{t=1}^{T}\left[\ell_t(X_t) - \ell_t(X_{t,j})\right] \le HD\sqrt{T}\left(\ln\frac{1}{p_{1,j}} + \frac{1}{8}\right). \tag{43}$$

Now let

$$m = \frac{P}{D}, \qquad q = \sqrt{1+2m}, \qquad \eta_\star = \frac{Dq}{H\sqrt{T}}. \tag{44}$$

These quantities are used only for analysis; the algorithm does not know P. The geometric step-size grid ensures that there exists an index j such that

$$b := \frac{\eta_j}{\eta_\star} \in \left[\frac{1}{2}, 1\right]. \tag{45}$$

Substituting $\eta_j = b\eta_\star$ into equation 38 and using $D^2 + 2DP = D^2q^2$, we obtain

$$\sum_{t=1}^{T}\left[\ell_t(X_{t,j}) - \ell_t(Z_t)\right] \le \frac{HD\sqrt{T}\,q}{2}\left(b + \frac{1}{b}\right) \le \frac{5}{4}HD\sqrt{T}\,q.$$

Here we use $\frac{1}{2}\left(b+\frac{1}{b}\right) \le \frac{5}{4}, b \in \left[\frac{1}{2}, 1\right]$.

On the other hand, this index satisfies $j \le 2 + \log_2 q$, so

$$\ln\frac{1}{p_{1,j}} \le \ln(j(j+1)) \le 2\ln(j+1) \le 2\ln(3q) \le 3q. \tag{46}$$

By equation 43,

$$\sum_{t=1}^{T}\left[\ell_t(X_t) - \ell_t(X_{t,j})\right] \le HD\sqrt{T}\left(3q + \frac{1}{8}\right) \le \frac{25}{8}HD\sqrt{T}\,q, \tag{47}$$

where the last inequality uses $q \ge 1$.

Adding equation 46 and equation 47, and using scale invariance, gives

$$\begin{aligned} R_T(W_{1:T}^{\mathrm{ref}}) = \sum_{t=1}^{T}\left[\ell_t(X_t) - \ell_t(Z_t)\right] &\le \left(\frac{5}{4} + \frac{25}{8}\right)HD\sqrt{T}\,q \\ &= \frac{35}{8}HD\sqrt{T}\sqrt{1+\frac{2P}{D}} = \frac{35G}{8\gamma}\sqrt{T\,(D^2+2DP)}. \end{aligned} \tag{48}$$

we further obtain

$$R_T(W_{1:T}^{\mathrm{ref}}) \le \frac{35\sqrt{2}\,G}{8\gamma}\sqrt{T\left(D^2 + D\,P_T(W_{1:T}^{\mathrm{ref}})\right)}. \tag{49}$$

Both the step-size grid and the weight updates are independent of the comparator sequence and $P_T(W_{1:T}^{\mathrm{ref}})$. The bound therefore holds for every admissible comparator sequence. □

D Discussions and Limitations

D.1 Discussions

Rethinking first-order optimization. Beyond optimizers based on geometric constraints, another promising direction is to use first-order information to approximate second-order Hessian information or even third-order information for improved optimization. Existing approaches can be broadly divided into two categories: methods that use gradient information of different orders to estimate the step size Shazeer & Stern (2018); Yuan (2026), and methods that use such information to estimate

the update direction Chen et al. (2026); Xie et al. (2026). In the context of learning-rate scheduling, it may be necessary to investigate more deeply how gradient or even Hessian information can be used to characterize the coupling between the step size and update direction.

Does scale invariance hold? The scale-invariance assumption on which our analysis relies does not strictly hold in practical training. Interestingly, however, the two desirable properties implied by scale invariance—namely, that the stochastic gradient is orthogonal to the model parameters and that the gradient decreases as the model norm increases—are both observed in practice Salimans & Kingma (2016); Wan et al. (2020). We therefore encounter an intriguing situation in which an incorrect assumption leads to correct predictions. This phenomenon warrants further investigation, as it may reflect more fundamental properties of neural-network architectures.

D.2 Limitations: Scope of the Scale-Invariance Assumption

The concurrent work ELR-FSL notes that practical Transformer architectures generally lack strict scale invariance, and directly rescaling matrix parameters does not necessarily preserve model outputs approximately. Its ablation experiments further show that fixing RMSNorm gains makes the parameterization more scale invariant but reduces the precision of scale effective learning-rate alignment. These findings indicate that static scale invariance alone cannot fully explain the alignment of loss trajectories. Consequently, our theoretical analysis under strict scale invariance has limited applicability to practical models and does not yet fully explain the empirical results of angular effective learning-rate alignment. Future work should examine how normalization configurations, optimizer states, and changes in parameter scale jointly affect alignment, and develop a theoretical explanation for models that are not strictly scale invariant.