YOUR LR SCHEDULE MAY NOT BE PRINCIPLED: EFFECTIVE LEARNING RATE SCHEDULING IN WEIGHT-NORMALIZED TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Weight-normalized (WN) training is emerging as a compelling alternative to conventional weight decay (WD), with recent studies reporting substantial pretraining loss gains. We show that these gains are driven primarily by the induced effective learning rate (ELR) trajectory rather than by normalization alone. Matching ELR trajectories largely collapses the training loss dynamics of WN and WD. *Schedule design should therefore target ELR rather than the nominal learning rate (LR) alone.* Building on this principle, we propose **AIR** (**A**ffine **I**nverse-square-**R**oot decay), which combines a theoretically derived $1/\sqrt{t}$ decay with an affine adjustment to satisfy prescribed peak and terminal LRs over a finite training horizon. We further identify late-stage ELR scheduling as a key source of its gains, because a shared late schedule attenuates the effect of earlier LR choices. Experiments on 128 B-token Mixture-of-Experts pretraining demonstrate that AIR outperforms cosine and other candidate schedules in training loss, downstream performance, and hyperparameter robustness, reducing terminal training loss by approximately 0.01 relative to cosine at matched cumulative LR. Applying the same ELR trajectory to WD reproduces the corresponding training dynamics, showing that the benefits extend beyond fixed-norm training and establishing ELR as the principled scheduling object.

1 INTRODUCTION

In Large Language Model (LLM) pretraining, weight-normalized (WN) training (Salimans & Kingma, 2016; Wen et al., 2026) has emerged as a promising alternative to conventional weight decay (WD) (Loshchilov & Hutter, 2019). It keeps parameter norms fixed after each update, with recent work reporting lower pretraining loss, improved token efficiency, greater stability, and better hyperparameter transfer (Wen et al., 2026; Loshchilov & Ginsburg, 2026; An et al., 2026). Yet WN and WD exhibit markedly different training dynamics under the same learning rate (LR) schedule. As shown in Figure 1(a), under cosine decay (Loshchilov & Hutter, 2017), WN initially trails WD but eventually outperforms it. However, under warmup-stable-decay (WSD) (Hu et al., 2024), WN remains behind WD throughout training and ends with a substantially higher loss. Therefore, understanding this contrast is essential to explaining when and why WN outperforms WD.

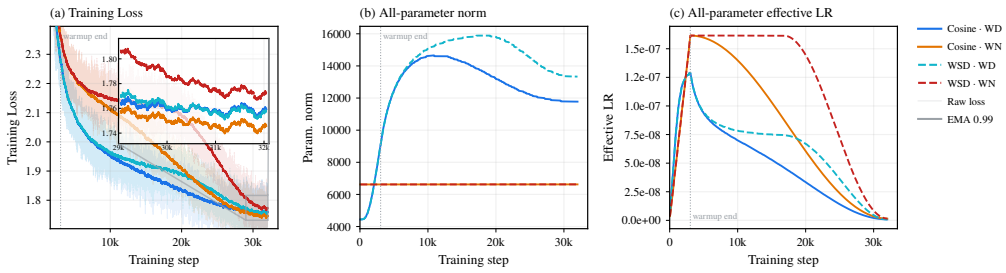


Figure 1: **WN achieves lower final loss than WD under cosine, but significantly higher under WSD.** Effective LR (nominal LR / all-parameter norm) is the key driver of loss dynamics.

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.

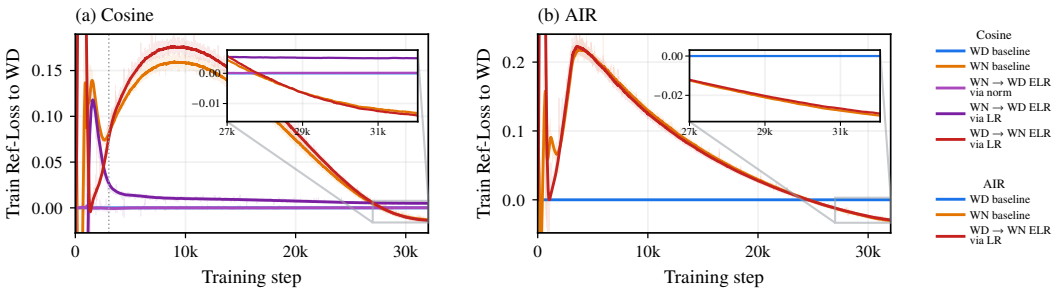


Figure 2: **ELR matching collapses WD-WN training loss dynamics.** Training loss minus the WD baseline under **(a)** cosine or **(b)** AIR, paired at the same step; zero denotes the reference. Arrows indicate the ELR target, matched by replaying WD norms (via norm) or rescaling blockwise LR (via LR). WD→WN also replays WD norms.

We attribute this contrast to the *effective learning rate* (ELR) (van Laarhoven, 2017; D'Angelo et al., 2024), defined as $\eta^{\text{eff}} = \eta/r$, where η is the nominal LR and r is the norm of a parameter block. In our study, we discover that the loss dynamics is strongly related to the decay of ELR. When WN and WD share the same WSD LR schedule, their ELR is governed entirely by the parameter norm r, which evolves very differently in the two regimes (Figure 1(b,c)). WN fixes the parameter norm, holding its ELR constant at peak throughout the entire stable phase and leading to a slow decrease in loss. By contrast, WD's ELR keeps falling steadily, owing to the continuing growth of its parameter norm. This sustained ELR decay keeps its loss dropping much faster than WN.

A more interesting finding is that the loss dynamics of WN and WD largely collapse onto a common trajectory once we align their ELR trajectories, whether by replaying WD's norms or by rescaling one's blockwise LRs. (see Figure 2 and Section 2.2 for details). Concurrent studies (Liu et al., 2026; Xiao et al., 2026; Zhou et al., 2026) also report similar results, corroborating our findings.

Taken together, these findings lead us to our core conclusion: ELR is the key driver of the loss dynamics. WN's core advantage actually comes from a better ELR schedule, not normalization alone. **Principled LR schedule design should therefore target *ELR* rather than the nominal LR alone.** This core idea naturally inspires an essential research question:

What should a better ELR schedule look like?

To address this question, we study ELR scheduling under WN, a clean setting in which each block's ELR is proportional to its nominal LR. Here, we find that $1/\sqrt{t}$ decay further improves training. A theoretical analysis of WN dynamics shows that, in the unconstrained setting, $1/\sqrt{t}$ decay maximizes alignment with an oracle direction at every step. To make this practical, we introduce **AIR** (**A**ffine **I**nverse-square-**R**oot decay), which adapts this $1/\sqrt{t}$ decay through an affine transformation to meet prescribed peak and terminal LRs over a finite horizon. A further analysis shows how a shared late-stage schedule attenuates the alignment differences induced by earlier LR choices. This helps explain our finding that *late-stage ELR scheduling* is a key source of AIR's gains.

Empirically, we evaluate AIR against cosine decay and alternative LR schedule families. In 128 B-token Mixture-of-Experts (MoE) (Shazeer et al., 2017) pretraining, AIR reduces terminal training loss by approximately 0.01 relative to cosine decay. Among these families, it achieves the lowest loss, the best downstream performance, and the greatest robustness to hyperparameter choices. Consistent with our theoretical analysis, controlled experiments show that the final training loss gaps between schedules narrow substantially when they share the same late-stage AIR schedule and differ only in their earlier LR trajectories. Transferring the induced ELR trajectory to WD also reproduces the corresponding training dynamics, showing that the benefits extend beyond fixed-norm training.

In summary, our contributions are:

- We identify the **induced ELR trajectory as the key driver** of WN's loss advantage over WD. Matching ELR trajectories largely aligns their training loss dynamics, highlighting ELR, rather than nominal LR alone, as the target for schedule design.

- We propose AIR, an affine $1/\sqrt{t}$ schedule for finite-horizon pretraining. Its decay shape follows from a theoretical analysis of directional alignment under WN, which also helps explain the role of *late-stage ELR scheduling* in AIR's gains.
- We empirically validate AIR against cosine decay and alternative schedule families in 128 B-token MoE pretraining, demonstrating improvements in training loss, downstream performance, and hyperparameter robustness. It **reduces terminal training loss by approximately 0.01 relative to cosine** at matched cumulative LR. Applying the same ELR trajectory to WD reproduces the corresponding training loss dynamics.

2 Effective Learning Rate as the Scheduling Principle

2.1 Effective learning rate under WN and WD

Let $\boldsymbol{\theta}_k$ denote a nonzero, vectorized parameter block at optimizer step k, η_k the nominal learning rate, and $\boldsymbol{u}_k$ the optimizer update before LR scaling. With decay coefficient $\lambda \geq 0$ and prescribed norm $R > 0$, decoupled WD (Loshchilov & Hutter, 2019) and fixed-norm WN (Wen et al., 2026) use the following update rules, respectively:

$$\text{WD:} \quad \boldsymbol{\theta}_{k+1} = (1 - \lambda\eta_k)\boldsymbol{\theta}_k + \eta_k \boldsymbol{u}_k, \tag{1}$$

$$\text{WN:} \quad \boldsymbol{\theta}_{k+1} = R\frac{\boldsymbol{\theta}_k + \eta_k \boldsymbol{u}_k}{\|\boldsymbol{\theta}_k + \eta_k \boldsymbol{u}_k\|}. \tag{2}$$

WD shrinks the current parameters before adding the optimizer update, while WN rescales the updated block to norm R, preserving its direction.

The same nominal LR can produce different relative updates when parameter norms differ. Writing $r_k = \|\boldsymbol{\theta}_k\|$ for the block's norm at step k, we define the blockwise norm-based *effective learning rate* (van Laarhoven, 2017; D'Angelo et al., 2024; Kosson et al., 2026) as

$$\eta_k^{\text{eff}} := \frac{\eta_k}{r_k}. \tag{3}$$

It captures how much a given optimizer update can affect the parameter direction, providing a more informative measure of its impact than the nominal LR alone (see Appendices A.1 and A.2).

2.2 Aligning the training dynamics of WN and WD

Experimental setup. To validate the claim that matching ELRs in Eq. (3) aligns the training dynamics of WN and WD, we match WN's ELR to WD's on each parameter block in two ways: (i) *Via norm.* We rescale each WN block at each step to the corresponding recorded WD norm while retaining the same nominal LR as WD. (ii) *Via LR.* We keep each WN block at its fixed norm R and adjust its blockwise nominal LR to satisfy

$$\frac{\eta_k^{\text{WN}}}{R} = \frac{\eta_k^{\text{WD}}}{r_k^{\text{WD}}} \implies \eta_k^{\text{WN}} = R\frac{\eta_k^{\text{WD}}}{r_k^{\text{WD}}}, \tag{4}$$

where r_k^{WD} is the recorded WD norm before the current update, and η_k^{WN} and η_k^{WD} are the respective nominal LRs.

For the reverse matching, we replay the WD norm trajectories and adjust blockwise LRs to match WN's ELR. Details are given in Appendix A.3. We conduct this reverse comparison under both cosine and AIR, alongside the corresponding WD and WN baselines.

Results. Figure 2 shows that ELR matching collapses the training loss dynamics between WD and WN. Under cosine, norm replay precisely reproduces WD dynamics, with a gap less than 2×10^{-4}, while LR matching at fixed WN norms also closely tracks the WD baseline, leaving a residual loss gap of several $\times 10^{-3}$. Under both cosine and AIR, matching WN's ELR along recorded WD norms yields a loss dynamics close to the WN baseline. The consistent collapse across both schedules supports ELR as a key driver of training dynamics and thus a principled target for schedule design.

2.3 WN AS A SETTING FOR ELR SCHEDULE DESIGN

Under WD, changing the nominal LR also affects the subsequent parameter norms, coupling the two quantities in Eq. (3), so that rescaling the nominal LR alone does not reproduce a target ELR trajectory (Kosson et al., 2026). WN removes this coupling by fixing $r_k = R$, giving $\eta_k^{\text{eff}} = \eta_k/R$, making each block's ELR proportional to its nominal LR (Wen et al., 2026; An et al., 2026). This direct proportionality makes WN a clean setting. Therefore, we study ELR scheduling under WN, using its update rule as the starting point for the theory.

3 DESIGNING A BETTER ELR SCHEDULE UNDER WN: AIR

3.1 DIRECTIONAL-ALIGNMENT DYNAMICS UNDER WN

Starting from the WN update in Eq. (2), which fixes a parameter block's norm and thus leaves only its direction to evolve, we analyze how the LR schedule affects this evolution. Let $\boldsymbol{v}$ denote an idealized unit target direction, fixed throughout the analysis. Given the training history $\mathcal{F}_0$ at the analysis starting point, define

$$m_k := \mathbb{E}\left[\frac{\boldsymbol{v}^\top \boldsymbol{\theta}_k}{R}\middle|\mathcal{F}_0\right]. \tag{5}$$

This is the expected cosine alignment after k updates from the same starting point. The expectation averages over minibatch sampling and other randomness in these k updates, while the initial history stays fixed. For this analysis, we consider an LR schedule specified in advance and assume tangential updates of fixed magnitude, $\boldsymbol{\theta}_k^\top \boldsymbol{u}_k = 0$ and $\|\boldsymbol{u}_k\| = U > 0$.

Deriving the alignment dynamics. Taking the conditional expectation of the inner product with $\boldsymbol{v}$ in Eq. (2) gives

$$m_{k+1} = \frac{m_k + a\eta_k}{\sqrt{1 + 2b\eta_k^2}}, \tag{6}$$

where

$$a := \mathbb{E}\left[\frac{\boldsymbol{v}^\top \boldsymbol{u}_k}{R}\middle|\mathcal{F}_0\right] > 0, \qquad b := \frac{U^2}{2R^2}. \tag{7}$$

We assume that a and b are constant across steps and schedules. The optimization below concerns these constant-coefficient scalar dynamics. For $2b\eta_k^2 \ll 1$, a second-order expansion yields the continuous approximation, with time measured in optimizer steps,

$$\dot{m}(t) = a\eta(t) - b\eta(t)^2 m(t). \tag{8}$$

A larger LR increases new alignment linearly but strengthens the retraction of existing alignment quadratically. Detailed derivations are provided in Appendix B.1.

Exact integral solution. For a nonnegative, square-integrable schedule on $[0, h]$ and initial alignment $m(0) = m_0 > 0$, Eq. (8) has the exact solution

$$m(h) = m_0 H(h,0) + a\int_0^h H(h,t)\eta(t)\,\mathrm{d}t, \quad H(h,t) := \exp\left[-b\int_t^h \eta(s)^2\,\mathrm{d}s\right]. \tag{9}$$

The solution decomposes endpoint alignment into retained initial alignment and retained contributions from subsequent updates. Each contribution is weighted by $H(h,t)$, which represents the fraction of alignment present at time t that remains at the endpoint h. Detailed derivations are provided in Appendix B.2.

3.2 UNCONSTRAINED OPTIMAL SCHEDULE: INVERSE-SQUARE-ROOT DECAY

We first maximize terminal alignment $m(h)$ in Eq. (9) without prescribing a peak, a terminal LR, or an LR budget. For fixed $a, b, m_0, h > 0$, completing the square in the alignment dynamics gives

$$\frac{\mathrm{d}}{\mathrm{d}t}m(t)^2 = \frac{a^2}{2b} - 2b\left(m(t)\eta(t) - \frac{a}{2b}\right)^2. \tag{10}$$

Since $m(t) > 0$, integration yields the bound $m(h) \le \sqrt{m_0^2 + a^2 h/(2b)}$, with equality if and only if $m(t)\eta(t) = a/(2b)$ almost everywhere on $[0, h]$. Under this condition, $m(t)^2$ grows linearly, giving $m^\star(t) = \sqrt{m_0^2 + a^2 t/(2b)}$ and hence the optimal time-shifted schedule:

$$\eta^\star(t) = \frac{\eta_0}{\sqrt{1 + 2b\eta_0^2 t}}, \qquad \eta_0 = \frac{a}{2bm_0}. \tag{11}$$

Because the same schedule attains the bound for every endpoint h, it maximizes expected alignment at every time under these dynamics. The $1/\sqrt{t}$ ELR profiles have also been observed to arise implicitly: without weight decay, growing parameter norms make the realized relative update decay as $t^{-1/2}$ (Kosson et al., 2026). Here the same shape instead follows from maximizing directional alignment under WN. Appendix B.3 gives the full proof and shows that its integer samples are also optimal for the discrete recurrence in Eq. (6). Appendix B.4 derives the geometric restrictions needed to interpret these states as expected cosine alignments.

3.3 AIR: A PRACTICAL SCHEDULE DESIGN

Building on the $1/\sqrt{t}$ decay in Eq. (11), we propose **AIR** (**A**ffine **I**nverse-square-**R**oot decay) for pretraining with prescribed peak and terminal LRs. The decay schedule controls how rapidly the LR decreases at different stages of training. To preserve the relative magnitudes of LR decay, we use an affine transformation that scales them by a common positive factor. Let $k = 0, \dots, K$ index the decay stage, with $K > 0$ its horizon. For a decay time scale $\alpha > 0$, define

$$q(k) = \left(1 + \frac{k}{\alpha}\right)^{-1/2}, \qquad f(k) = \frac{q(k) - q(K)}{1 - q(K)}. \tag{12}$$

Here, $f(k)$ rescales the $1/\sqrt{t}$ curve $q(k)$ from 1 to 0. For a peak LR p and a terminal LR $0 < \eta_{\min} < p$, the resulting schedule is

$$\eta_{\mathrm{AIR}}(k) = \eta_{\min} + (p - \eta_{\min}) f(k), \qquad k = 0, \dots, K. \tag{13}$$

This schedule decreases monotonically from p at $k = 0$ to $\eta_{\min}$ at $k = K$. Appendix B.5 derives this affine construction and shows that it preserves the ratios between decay rates. We use $\alpha = K/8$ in the main comparisons and evaluate sensitivity to this choice in Section 4.3.

When an LR budget is prescribed instead of the peak, we determine p from the complete discrete LR sum, including warmup, with the other settings fixed in Appendix B.6.

4 EMPIRICAL EVALUATION

4.1 EXPERIMENTAL SETUP

Pretraining setting. We pretrain a mixture-of-experts (MoE) model with 110 M activated parameters and 4.7 B total parameters on 128 B tokens using Muon (Jordan et al., 2024) under WN. Each complete run takes 48 hours on 16 GPUs. To further test scalability, we apply the same scheduler design to a larger MoE with 790 M activated and 35.7 B total parameters, trained on 500 B tokens, with each run taking 168 hours on 32 GPUs. Additional experimental settings and reporting conventions are given in Appendix C.

Comparison controls. We perform a hyperparameter search on the cosine baseline and select the best peak LR and warmup ratio. For a fair comparison across all schedule families, we use linear warmup with the same duration and match the terminal LR and cumulative LR, defined as the sum of LRs over all training steps. Each schedule's peak LR is calibrated to match this sum, including warmup, as described in Appendix B.6. All schedules are compared at the end of training.

Figure 3(a) shows the cosine baseline, raw $1/\sqrt{t}$ decay (Vaswani et al., 2017), and its affine transformation AIR, defined by Eqs. (12)–(13). The raw curve is shown as a reference with the same peak as AIR, without matching its terminal or cumulative LR. We also compare other decay shapes under the same controls and affine construction, including e^{-t} (Xiao et al., 2026), $1 - \log(1 + t)$ (Loshchilov & Ginsburg, 2026), and $1 - \sqrt{t}$ (Hägele et al., 2024; Zhou et al., 2026). Appendix C.2 lists the complete set of decay shapes.

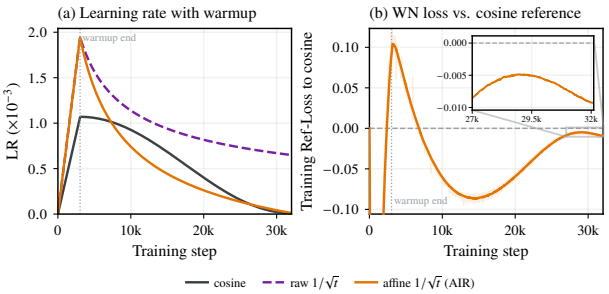


Figure 3: **Learning rate schedules and training loss under WN.** With the same terminal and cumulative LRs, AIR achieves lower final training loss than cosine.

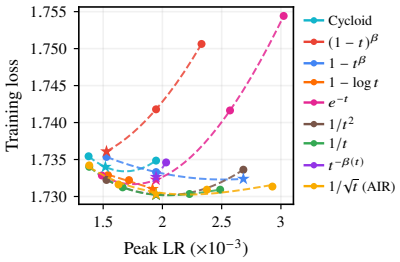


Figure 4: **AIR stays near its best loss over the widest peak LR range.** Circles are measured configurations, stars the best per family, and dashed curves family-wise convex fits.

Table 1: **Terminal losses and downstream performance of matched-budget decay shapes under WN.** Each row is labeled by its raw decay shape before the common affine transformation and uses that shapes best parameter selected during evaluation (Appendix C.2). AIR achieves the lowest terminal losses and attains the best downstream average at both model scales.

Schedule	Train Loss (↓)	Eval Loss (↓)	MMLU (↑)	MMLU-Pro (↑)	AGIEval (↑)	WinoGrande (↑)	BBH (↑)	DROP (↑)	GSM8k (↑)	MATH (↑)	HumanEval (↑)	MBPP+ (↑)	TriviaQA (↑)	Avg. (↑)
110 M/4.7 B model														
Cosine	1.740	2.185	49.0	18.2	29.1	49.5	30.9	22.0	17.0	11.1	28.7	35.2	30.9	29.24
Linear	1.742	2.187	48.9	16.4	26.3	49.6	31.1	22.6	13.4	11.3	26.2	35.4	31.2	28.40
Half-cosine	1.749	2.195	48.6	15.5	28.2	50.7	**32.5**	**23.9**	15.0	11.7	24.4	34.7	29.5	28.61
Rear-cosine	1.736	2.180	48.0	16.2	28.7	**51.2**	31.6	22.3	14.3	10.5	25.0	36.2	30.7	28.61
Cycloid	1.734	2.178	47.3	16.0	27.5	49.9	30.7	22.6	15.1	10.0	26.2	33.6	31.1	28.18
$(1-t)^2$	1.736	2.181	48.1	16.8	27.6	50.7	30.6	22.1	16.5	12.0	27.4	**36.8**	29.5	28.92
$1-t^\beta$	1.732	**2.174**	48.5	18.1	**30.2**	50.9	31.8	22.3	16.4	10.9	**31.1**	35.2	32.3	29.79
$1-\log t$	1.731	2.175	48.1	15.6	29.6	48.8	32.0	21.5	16.8	11.6	23.8	34.4	**33.2**	28.67
e^{-t}	1.732	2.176	48.9	16.0	29.2	50.6	31.6	21.2	16.8	10.8	27.4	**36.8**	31.7	29.18
$1/t^2$	**1.730**	2.175	**50.5**	18.4	29.0	50.8	32.1	22.2	**17.5**	11.6	29.9	33.1	30.3	29.58
$1/t$	**1.730**	**2.174**	49.7	16.9	29.6	50.8	30.8	23.0	16.7	12.2	26.8	32.8	29.7	29.00
$t^{-\beta(t)}$	1.733	2.176	48.8	16.7	30.1	50.7	30.0	22.9	14.2	10.6	28.0	34.7	30.6	28.85
$1/\sqrt{t}$ (AIR)	**1.730**	**2.174**	49.7	**18.6**	29.6	50.1	31.1	23.8	15.8	**13.0**	29.9	36.5	31.6	**29.97**
790 M/35.7 B model														
Cosine	1.431	1.843	70.7	40.6	54.2	50.4	56.2	**56.0**	59.9	36.8	67.7	60.1	65.8	56.22
$1/\sqrt{t}$ (AIR)	**1.421**	**1.832**	**72.1**	**42.4**	**55.7**	**51.8**	**59.4**	51.2	**64.2**	**37.2**	**70.1**	**65.3**	**66.2**	**57.78**

4.2 Pretraining loss and downstream performance

Experimental setup. Following Section 4.1, all schedules share the same warmup, terminal LR, and cumulative LR, and each decay shape enters the comparison at its best shape parameter selected during evaluation, listed in Table 4. We report terminal training and evaluation loss together with performance on the downstream benchmarks listed in Appendix C.4. We repeat the comparison between cosine and AIR at a larger model scale.

Results. Figure 3(b) and Table 1 show that AIR lowers terminal training and evaluation loss by about 0.01 relative to cosine at both model scales and achieves the lowest reported losses among the compared schedulers. It also attains the best downstream average among all compared shapes, and at the larger scale it improves over cosine on most benchmarks, highlighting the scale-up ability of our scheduler design.

4.3 Robustness to hyperparameter choices

Experimental setup. Under the same matched controls, we sweep each family's shape parameter, which also moves its peak LR. For each family we report its LR sensitivity S, the average loss above the family's best configuration per e-fold of peak LR, defined formally in Appendix C.5 (Eq. (31)).

Results. Figure 4 plots terminal training loss against peak LR with family-wise polynomial fits, and Table 5 in Appendix C.5 reports the resulting LR sensitivity. AIR has the lowest LR sensitivity over

the widest range of peak LR. Its region of low loss is therefore wide enough that the gain does not depend on a narrowly tuned peak LR and thus facilitates LR transfer (Ren et al., 2026).

5 Late-Stage Scheduling Matters More

Having established the gains of AIR in Section 4, we further explore where in training this advantage is created. We find that late-stage scheduling drives the final loss gap (Section 5.1) and that sharing the late schedule substantially reduces gaps created by distinct early schedules (Section 5.2). Further comparisons in Section 5.3 reveal a trade-off between sustained annealing and gradual LR changes, with AIR naturally striking this balance. Finally, we provide a brief theoretical explanation in Section 5.4 of how early and late schedules affect terminal alignment.

5.1 Late-stage scheduling drives the final loss gap

Among the candidate schedules matched to the same cumulative LR, a Cycloid schedule with $\theta_1 = \pi$ happens to take exactly the same peak LR as AIR, so the two runs share their warmup and terminal LR as well and differ only in the shape of the decay in between (Figure 5(a)). However, their final training losses still differ by approximately 0.005, a substantial gap at this scale. Notice that the two intersections of their decay curves naturally divide the training horizon into early, middle, and late stages, we then swap one stage at a time between them while leaving the other two unchanged.

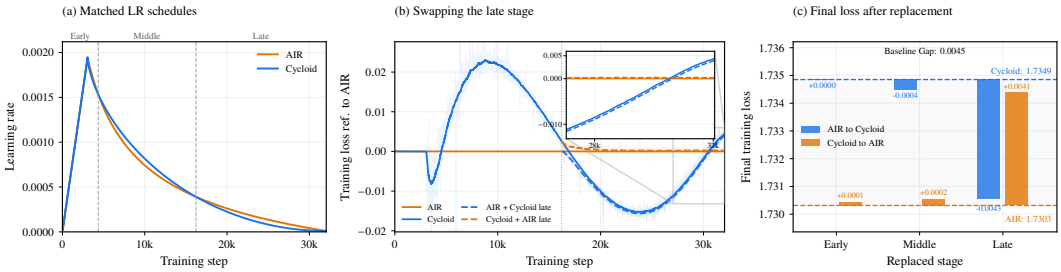


Figure 5: **Late-stage scheduling dominates the final loss difference.** Starting from two full schedules with matched peak, terminal, and cumulative LRs, swapping the late stage between AIR and Cycloid nearly transfers the full final loss gap to either schedule, whereas swaps of the early and middle stages have little effect. The bars use the terminal exponential moving average (EMA-0.99) training loss, as in Table 1; the intervention and reporting details are given in Appendix C.6.

As shown in Figure 5(b,c), swaps of the early and middle stages leave the final loss nearly unchanged, whereas swapping the late stage closes about 90 % of the gap for either schedule. For this matched pair, the gap is almost entirely attributable to the late stage. This motivates us to examine whether a shared late schedule can reduce the final loss gap created by distinct early schedules. A related effect was reported for scale-invariant networks trained with weight decay, where reaching the equilibrium of a late phase makes the outcome largely independent of earlier phases (Li et al., 2020).

5.2 Shared late scheduling reduces the final loss gap

Experimental setup. To address this question, we consider six decay schedules, Cosine, Linear, Half-cosine, Double-cosine, $(1-t)^2$, and $1-\sqrt{t}$, whose original runs end with training losses 0.005–0.018 higher than AIR. For each schedule, we retain its decay shape over the first 70 % of training steps and concatenate it with the same late-stage schedule of AIR. The early segment and its warmup target are rescaled to match the cumulative LR under the same comparison constraints as in Section 4.1. The two runs of each schedule are denoted *original* and *shared tail*, respectively.

Results. Table 2 shows that the shared late schedule lowers terminal training and evaluation loss for all six schedules, closing 42 %–83 % and 45 %–94 % of their respective gaps to AIR. Thus, a shared late schedule substantially reduces the performance differences arising from different early schedules.

Table 2: **A shared late schedule narrows the endpoint gap for every early decay shape.** Each schedule is trained twice, once in its original form and once with the common late schedule. Parentheses give the gap of each loss to the AIR anchor in the last row, and Gap is the fraction of that gap the shared tail removes. Training loss is the terminal EMA-0.99 value and evaluation loss the terminal raw value, as in Table 1.

Schedule	Training loss (relative to AIR)			Eval loss (relative to AIR)		
	Original	Shared tail	Gap closed (%) ↑	Original	Shared tail	Gap closed (%) ↑
Cosine	1.7399 (+0.0096)	1.7358 (+0.0055)	42%	2.1846 (+0.0106)	2.1799 (+0.0059)	45%
Linear	1.7415 (+0.0112)	1.7341 (+0.0038)	66%	2.1875 (+0.0135)	2.1787 (+0.0047)	65%
Half-cosine	1.7487 (+0.0184)	1.7374 (+0.0071)	62%	2.1950 (+0.0210)	2.1815 (+0.0076)	64%
Double-cosine	1.7446 (+0.0143)	1.7328 (+0.0025)	83%	2.1901 (+0.0161)	2.1769 (+0.0029)	82%
$(1-t)^2$	1.7361 (+0.0058)	1.7319 (+0.0016)	72%	2.1805 (+0.0065)	2.1753 (+0.0013)	80%
$1-\sqrt{t}$	1.7354 (+0.0050)	1.7316 (+0.0012)	75%	2.1787 (+0.0047)	2.1743 (+0.0003)	94%
$1/\sqrt{t}$ (AIR)	1.7303	–	–	2.1740	–	–

5.3 A TRADE-OFF IN LATE-STAGE SCHEDULING

Experimental setup. Having established the importance of late-stage scheduling, we next examine what makes a late schedule effective. Based on the design of AIR in Section 3.3, we vary the final 20 % of its decay stage, starting at $K_s \approx 0.8K$, as

$$\eta_\gamma(k) = \eta_{\min} + \left[\eta_{\mathrm{AIR}}(K_s) - \eta_{\min}\right] \left(\frac{K-k}{K-K_s}\right)^\gamma, \quad K_s \le k \le K, \tag{14}$$

where $\gamma \in \{0.5, 0.75, 1, 2, 3\}$. Smaller exponents postpone more of the LR reduction to the endpoint, whereas larger exponents reduce the LR earlier and then flatten out. The junction and terminal LRs remain fixed, while the area under the recorded LR traces varies by less than 1 % (Appendix C.7).

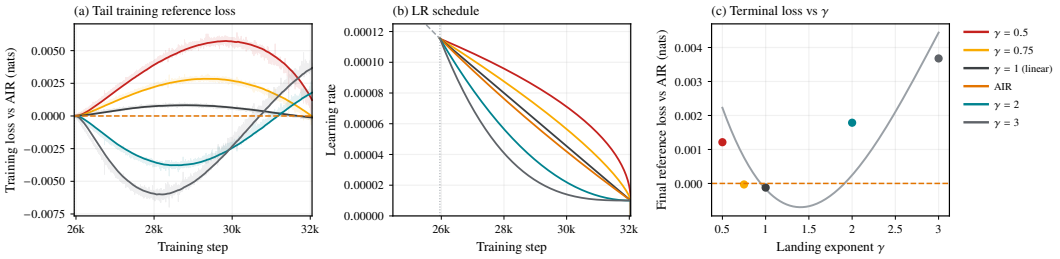


Figure 6: **Late-stage decay exhibits a two-sided trade-off.** With the junction and terminal LRs fixed, both overly abrupt decay and early flattening increase final loss.

Results. Figure 6 shows that both the abrupt decay at $\gamma = 0.5$ and the early flattening at $\gamma \in \{2, 3\}$ increase final loss, while AIR and the variants with $\gamma \in \{0.75, 1\}$ perform comparably, achieving the lowest final losses. These results suggest a trade-off: late-stage scheduling should sustain annealing while avoiding overly abrupt LR changes. The design of AIR naturally balances these requirements and performs close to the best variants in this sweep.

5.4 A THEORETICAL EXPLANATION OF SCHEDULE DEPENDENCE

To interpret these results, we use the alignment dynamics of Section 3 to separate the effects of early- and late-stage schedule changes. Consider bounded nonnegative schedules η_1, η_2 on the decay interval $[0, K]$, with common $a, b, m_0, \mathcal{F}_0$ and a fixed stage boundary K_s. Write $\Delta m = m_1 - m_2$ and $\Delta\eta = \eta_1 - \eta_2$, where the expected alignments m_i retain the conditioning in Eq. (5) throughout. For a shared late schedule, $\eta_1 = \eta_2 = \eta$ on $[K_s, K]$, Eq. (9) gives

$$\Delta m(K) = H(K, K_s)\Delta m(K_s), \quad H(K, K_s) = \exp\left[-b \int_{K_s}^{K} \eta(k)^2 \, \mathrm{d}k\right]. \tag{15}$$

Since $0 < H(K, K_s) \leq 1$, the shared late stage attenuates the entering difference without reversing its sign, ignoring stochastic fluctuations around the mean. Over a fixed late interval, higher LRs increase the cumulative squared LR and reduce $H(K, K_s)$, thereby weakening the effect of early-stage scheduling on endpoint alignment. The derivation is given in Appendix D.2.

For different late-stage schedules, a shared early schedule gives $\Delta m(K_s) = 0$. For bounds $0 \leq \eta_i \leq P$ and $0 \leq m_i \leq M$ over the late-stage interval, let $L_{12} = \max\{a, |a - 2bPM|\}$ and $H_{12} = \sqrt{H_1 H_2}$, with H_i the retention factor for η_i. This gives

$$|\Delta m(K)| \leq L_{12} \int_{K_s}^{K} H_{12}(K, k) |\Delta \eta(k)| \, \mathrm{d}k. \tag{16}$$

Smaller weighted LR differences give a tighter bound on the terminal alignment difference. Within this bound, differences nearer the endpoint receive greater weight because less subsequent attenuation remains. Details are given in Appendix D.3. Further, if the schedules differ in both stages,

$$|\Delta m(K)| \leq H_{12}(K, K_s) |\Delta m(K_s)| + L_{12} \int_{K_s}^{K} H_{12}(K, k) |\Delta \eta(k)| \, \mathrm{d}k. \tag{17}$$

The first term retains the early-stage difference, while the second bounds the late-stage contribution. Slope-based refinements are in Appendix D.4. These bounds on expected alignment provide a qualitative explanation for the observed loss gaps.

6 Related Work

WN and related constraints provide explicit parameter-scale control as an alternative to WD, with reported efficiency and stability gains (Wen et al., 2026; An et al., 2026). Methods differ in whether they separate magnitude from direction (Salimans & Kingma, 2016) or fix weight and update norms (Wen et al., 2026). The choice of matrix norm further shapes the weight constraints and update geometry (Xie et al., 2026; Xu et al., 2026). For manifold constraints, Su (2025a;b) and Bernstein (2025) optimize tangent-space descent directions before retracting weights, so that the update respects the imposed geometry.

Beyond norm control, scheduling determines how the effective step size evolves during training. River-valley analyses explain WSD (Wen et al., 2025), while convex bounds and functional scaling laws motivate cooldown and power decay (Schaipp et al., 2025; Li et al., 2026). However, these theories focus on nominal LR alone, which is not principled when parameter norms evolve (D'Angelo et al., 2024). Concurrent work corroborates our finding that matching ELR largely aligns WN and WD loss dynamics (Xiao et al., 2026; Liu et al., 2026; Zhou et al., 2026), but does not derive a decay law from an optimization objective. We address this gap by optimizing directional alignment under WN to derive $1/\sqrt{t}$ decay, then adapting it to finite horizons through an affine transformation, yielding AIR as an effective ELR schedule.

7 Conclusion and Future Work

We identify the induced effective learning rate (ELR) trajectory as the key driver of WN's advantage over WD, with matched ELRs largely aligning their training loss dynamics. This finding motivates schedule design that targets ELR directly. Our theoretical analysis of directional alignment under WN, together with an affine adaptation of the unconstrained optimal $1/\sqrt{t}$ schedule, yields AIR for finite-horizon training. Experiments on MoE models at two scales show that, at matched cumulative LR, AIR reduces terminal loss by about 0.01 relative to cosine and improves downstream performance, while schedule sweeps demonstrate hyperparameter robustness. Further stage-wise interventions show that sharing the late schedule substantially narrows final loss gaps from different early schedules, identifying late-stage scheduling as a key source of AIR's effectiveness. Transferring its ELR trajectory to WD reproduces WN training loss dynamics, demonstrating that our design extends beyond fixed-norm training.

Future work includes extending our directional-alignment theory to account for evolving update statistics and quantitatively connect alignment to loss. We also aim to investigate the ELR principle and AIR across more architectures, optimizers, and larger training scales, and to study ELR scheduling in continued training (CT), supervised fine-tuning (SFT), and reinforcement learning (RL).

REPRODUCIBILITY STATEMENT

Appendix A derives the geometry of directional updates and blockwise ELR matching. Appendix B provides the alignment derivations, continuous and discrete unconstrained optimality proofs, their geometric scope, and the affine schedule construction and discrete LR sum calibration. Appendix C documents the training settings, compared schedule definitions, loss summaries, downstream benchmarks, hyperparameter sweeps, and interventions across training stages. Appendix D proves bounds for differences between early and late schedules.

AI USE STATEMENT

Generative AI tools were used for literature discovery, including identifying relevant literature and summarizing or analyzing existing literature. They also assisted with creating or modifying scientific figures; preparing implementation and plotting code; and revising and polishing the manuscript to improve readability. In addition, generative AI tools were used in supporting roles to assist with mathematical writing and checking, including mainly checking and refining the formulation of proofs; to design or provide feedback on research methodology or experiments; to implement methods; and to assist with translation.

Generative AI tools were not used to generate synthetic datasets; develop theoretical models or conceptual frameworks; formulate mathematical claims; provide critical ingredients for proving mathematical claims; propose or refine hypotheses; clean or reformat datasets; support qualitative or thematic data analysis; or interpret results.

The authors reviewed all AI-assisted work. Implementation and plotting code was reviewed and executed; literature suggestions and citations were checked against the original sources; and translations and manuscript revisions were reviewed before inclusion. The authors take responsibility for the final content, including all text, claims, code, figures, and other artifacts produced with AI assistance. No experimental result was generated or imputed by a generative AI tool.

ETHICS STATEMENT

This work studies optimization for language model pretraining and does not involve human subjects or introduce a deployed user-facing system. The proposed schedule changes training dynamics; it is not a safety intervention. Lower pretraining loss does not by itself establish downstream safety, fairness, privacy, factuality, or suitability for deployment. The benefits and risks of models trained with the proposed schedule depend on their training data, capabilities, and deployment context, which must be evaluated separately.

REFERENCES

Kang An, Jiaxiang Li, Donald Goldfarb, and Shiqian Ma. Demystifying Manifold Constraints in LLM Pre-training, 2026. URL https://arxiv.org/abs/2605.04418.

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

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating Large Language Models Trained on Code, 2021. URL https://arxiv.org/abs/2107.03374.

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, 2021. URL https://arxiv.org/abs/2110.14168.

Francesco D'Angelo, Maksym Andriushchenko, Aditya Varre, and Nicolas Flammarion. Why Do We Need Weight Decay in Modern Deep Learning? In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 23191–23223. Curran Associates, Inc., 2024. doi: 10.52202/079017-0730. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/29496c942ed6e08ecc469f4521ebfff0-Paper-Conference.pdf.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2368–2378, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1246. URL https://aclanthology.org/N19-1246/.

Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro Von Werra, and Martin Jaggi. Scaling Laws and Compute-Optimal Training Beyond Fixed Training Durations. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 76232–76264. Curran Associates, Inc., 2024. doi: 10.52202/079017-2427. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/8b970e15a89bf5d12542810df8eae8fc-Paper-Conference.pdf.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 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 J. Vanschoren and S. Yeung (eds.), *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/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf.

Shengding Hu, Yuge Tu, Xu Han, Ganqu Cui, Chaoqun He, Weilin Zhao, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Xinrong Zhang, Zhen Leng Thai, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, dahai li, Zhiyuan Liu, and Maosong Sun. MiniCPM: Unveiling the Potential of Small Language Models with Scalable Training Strategies. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=3X2L2TFr0f.

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

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Atli Kosson, Jeremy Welborn, Yang Liu, Martin Jaggi, and Xi Chen. Weight Decay may matter more than μP for Learning Rate Transfer in Practice. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 67012–67048,

2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/6dc63b4063c978cf195bc15178e8152a-Paper-Conference.pdf.

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 (Extended Abstract). In Steve Hanneke and Tor Lattimore (eds.), *Proceedings of Thirty Ninth Conference on Learning Theory*, volume 336 of *Proceedings of Machine Learning Research*, pp. 4722–4723. PMLR, 29 Jun–03 Jul 2026. URL https://proceedings.mlr.press/v336/li26b.html.

Zhiyuan Li, Kaifeng Lyu, and Sanjeev Arora. Reconciling Modern Deep Learning with Traditional Optimization Analyses: The Intrinsic Learning Rate. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 14544–14555. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/a7453a5f026fb6831d68bdc9cb0edcae-Paper.pdf.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and LINGMING ZHANG. Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 21558–21572. Curran Associates, Inc., 2023. doi: 10.52202/075280-0943. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/43e9d647ccd3e4b7b5baab53f0368686-Paper-Conference.pdf.

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, 2026. URL https://arxiv.org/abs/2608.24814.

Ilya Loshchilov and Boris Ginsburg. Training nGPT, 2026. URL https://arxiv.org/abs/2608.01284.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Liliang Ren, Yang Liu, Yelong Shen, and Weizhu Chen. Rethinking Language Model Scaling under Transferable Hypersphere Optimization, 2026. URL https://arxiv.org/abs/2603.28743.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An Adversarial Winograd Schema Challenge at Scale. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8732–8740, April 2020. ISSN 2159-5399. doi: 10.1609/aaai.v34i05.6399. URL http://dx.doi.org/10.1609/aaai.v34i05.6399.

Tim Salimans and Durk P Kingma. Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/ed265bc903a5a097f61d3ec064d96d2e-Paper.pdf.

Fabian Schaipp, Alexander Hägele, Adrien Taylor, Umut Simsekli, and Francis Bach. The Surprising Agreement Between Convex Optimization Theory and Learning-Rate Scheduling for Large Model Training. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 53267–53294. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/schaipp25a.html.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *International Conference on Learning Representations*, 2017. URL https://mlanthology.org/iclr/2017/shazeer2017iclr-outrageously/.

Jianlin Su. Steepest Descent on Manifolds: 1. SGD + Hypersphere, Aug 2025a. URL https://kexue.fm/archives/11196.

Jianlin Su. Steepest Descent on Manifolds: 3. Muon + Stiefel, Aug 2025b. URL https://kexue.fm/archives/11221.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.824. URL https://aclanthology.org/2023.findings-acl.824/.

Twan van Laarhoven. L2 Regularization versus Batch and Weight Normalization, 2017. URL https://arxiv.org/abs/1706.05350.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

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 A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 95266–95290. Curran Associates, Inc., 2024. doi: 10.52202/079017-3018. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ad236edc564f3e3156e1b2feafb99a24-Paper-Datasets_and_Benchmarks_Track.pdf.

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 Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 42840–42885, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/6a1fe80a9e2dcda0b3e5fd0fd87eb097-Paper-Conference.pdf.

Kaiyue Wen, Xingyu Dang, Kaifeng Lyu, Tengyu Ma, and Percy Liang. Fantastic Pretraining Optimizers and Where to Find Them II: Hyperball Optimization, 2026. URL https://arxiv.org/abs/2606.16899.

Yihao Xiao, Jialong Sun, Zitian Gao, Zeming Wei, Chutian Wang, Ran Tao, Jiaye Teng, and Bryan Dai. Hyperball May Not Be a Free Lunch, 2026. URL https://arxiv.org/abs/2607.22444.

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.

Ruihan Xu, Jiajin Li, and Yiping Lu. On the Width Scaling of Neural Optimizers Under Matrix Operator Norms I: Row/Column Normalization and Hyperparameter Transfer, 2026. URL https://arxiv.org/abs/2603.09952.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. AGIEval: A Human-Centric Benchmark for Evaluating Foundation Models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 2299–2314, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.149. URL https://aclanthology.org/2024.findings-naacl.149/.

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 Directional Geometry and ELR Matching

We derive the geometric interpretation of ELR in Section 2.1 and the blockwise matching rule used in Section 2.2.

A.1 Exact normalized updates

For a nonzero parameter block, retain $r_k = \|\boldsymbol{\theta}_k\|$ and $\eta_k^{\text{eff}} = \eta_k / r_k$ from Section 2.1, and write $\hat{\boldsymbol{\theta}}_k = \boldsymbol{\theta}_k / r_k$ for its unit direction. Throughout, $\eta_k \geq 0$, and norms are Euclidean on vectorized blocks. Under fixed-norm WN, $r_k = R$. Dividing the updates in Eqs. (1) and (2) by their pre-update parameter norms and normalizing gives

$$\text{WD:} \quad \hat{\boldsymbol{\theta}}_{k+1} = \frac{(1 - \lambda\eta_k)\hat{\boldsymbol{\theta}}_k + \eta_k^{\text{eff}} \boldsymbol{u}_k}{\|(1 - \lambda\eta_k)\hat{\boldsymbol{\theta}}_k + \eta_k^{\text{eff}} \boldsymbol{u}_k\|}, \tag{18}$$

$$\text{WN:} \quad \hat{\boldsymbol{\theta}}_{k+1} = \frac{\hat{\boldsymbol{\theta}}_k + \eta_k^{\text{eff}} \boldsymbol{u}_k}{\|\hat{\boldsymbol{\theta}}_k + \eta_k^{\text{eff}} \boldsymbol{u}_k\|}, \qquad \eta_k^{\text{eff}} = \eta_k / R. \tag{19}$$

These identities hold whenever the vector being normalized is nonzero. They keep the realized optimizer update explicit. The ELR multiplies that update relative to the current parameter norm, while WD also contributes radial shrinkage.

A.2 Radial and tangential components

Define

$$u_{\parallel,k} = \hat{\boldsymbol{\theta}}_k^\top \boldsymbol{u}_k, \qquad \boldsymbol{u}_k^\perp = \boldsymbol{u}_k - u_{\parallel,k} \hat{\boldsymbol{\theta}}_k.$$

The scalar $u_{\parallel,k}$ is the radial component, and $\boldsymbol{u}_k^\perp$ is the component orthogonal to the parameter direction. Under WD, the updated vector is

$$\left[(1 - \lambda\eta_k) r_k + \eta_k u_{\parallel,k}\right] \hat{\boldsymbol{\theta}}_k + \eta_k \boldsymbol{u}_k^\perp.$$

Orthogonality gives its squared norm as the sum of the squared radial and tangential terms. The unsigned angle $\phi_k \in [0, \pi]$ between the old and updated directions is therefore

$$\phi_k = \operatorname{atan2}\big(\eta_k \|\boldsymbol{u}_k^\perp\|, (1 - \lambda\eta_k) r_k + \eta_k u_{\parallel,k}\big)\,. \tag{20}$$

For WN, the same expression applies with $\lambda = 0$ and $r_k = R$, because the final normalization preserves direction.

To obtain the first-order change, the derivative of $\boldsymbol{x} \mapsto \boldsymbol{x}/\|\boldsymbol{x}\|$ at a unit vector $\hat{\boldsymbol{\theta}}_k$ is $\boldsymbol{I} - \hat{\boldsymbol{\theta}}_k \hat{\boldsymbol{\theta}}_k^\top$. Applied to the increment $-\lambda\eta_k \hat{\boldsymbol{\theta}}_k + \eta_k^{\text{eff}} \boldsymbol{u}_k$, it removes the radial term and gives

$$\hat{\boldsymbol{\theta}}_{k+1} - \hat{\boldsymbol{\theta}}_k = \eta_k^{\text{eff}} \boldsymbol{u}_k^\perp + O\big((\lambda\eta_k + \eta_k^{\text{eff}} \|\boldsymbol{u}_k\|)^2\big)\,. \tag{21}$$

The expansion requires $\lambda\eta_k + \eta_k^{\text{eff}} \|\boldsymbol{u}_k\| \ll 1$. The WN expression again sets $\lambda = 0$. For a given tangential optimizer update, the ELR thus sets the first-order scale of the directional change.

A.3 Blockwise ELR transfer

For a target trajectory $\eta_k^{\text{eff,tar}}$, choosing $\eta_k^{\text{WD}} = r_k^{\text{WD}} \eta_k^{\text{eff,tar}}$ gives $\eta_k^{\text{WD}} / r_k^{\text{WD}} = \eta_k^{\text{eff,tar}}$ by substitution. A WN schedule supplies the target $\eta_k^{\text{eff,tar}} = \eta_k^{\text{WN}} / R$, giving

$$\eta_k^{\text{WD}} = \frac{r_k^{\text{WD}}}{R} \eta_k^{\text{WN}}. \tag{22}$$

This is the reverse direction of the coefficient matching in Eq. (4). We apply this rule separately to each controlled block, using the recorded WD norm r_k^{WD} replayed at step k in the reverse-matching experiments.

B WN SCHEDULE DESIGN AND CALIBRATION

We first derive the alignment dynamics and their unconstrained optimum from Section 3. We then establish the affine endpoint construction and discrete LR sum calibration used for finite-horizon schedules.

B.1 FROM THE WN UPDATE TO ALIGNMENT DYNAMICS

Under the assumptions in Section 3.1, $\|\boldsymbol{\theta}_k\| = R$, $\boldsymbol{\theta}_k^\top \boldsymbol{u}_k = 0$, and $\|\boldsymbol{u}_k\| = U$. Hence

$$\|\boldsymbol{\theta}_k + \eta_k \boldsymbol{u}_k\|^2 = R^2 + \eta_k^2 U^2.$$

Taking the inner product of Eq. (2) with $\boldsymbol{v}/R$ gives

$$\frac{\boldsymbol{v}^\top \boldsymbol{\theta}_{k+1}}{R} = \frac{\boldsymbol{v}^\top \boldsymbol{\theta}_k / R + \eta_k \boldsymbol{v}^\top \boldsymbol{u}_k / R}{\sqrt{1 + 2b\eta_k^2}}.$$

We condition on a fixed initial history $\mathcal{F}_0$ and a prescribed LR schedule, so η_k and the denominator are deterministic under this conditioning. We average over the first k updates that generate $\boldsymbol{\theta}_k$ and the randomness in $\boldsymbol{u}_k$. No independence between $\boldsymbol{\theta}_k$ and $\boldsymbol{u}_k$ is needed. For an LR selected adaptively from later random observations, this factorization would in general fail. The definitions of m_k and a then give Eq. (6). For $2b\eta_k^2 \ll 1$, the binomial expansion gives

$$(1 + 2b\eta_k^2)^{-1/2} = 1 - b\eta_k^2 + O(b^2\eta_k^4).$$

Multiplying by $m_k + a\eta_k$ and subtracting m_k yields

$$m_{k+1} - m_k = a\eta_k - bm_k\eta_k^2 + O\big(ab\eta_k^3 + b^2|m_k|\eta_k^4\big)\,. \tag{23}$$

Approximating this increment by a derivative with one optimizer step as the time unit gives

$$\dot{m}(t) = a\eta(t) - b\eta(t)^2 m(t).$$

This establishes the second-order discrete increment underlying Eq. (8). Replacing that increment by a derivative is a further continuous approximation, requiring small relative updates, slow variation of the LR and alignment over individual optimizer steps, and approximately constant update statistics.

B.2 DERIVATION OF THE INTEGRAL SOLUTION

To solve Eq. (8) on $[0, h]$ with $m(0) = m_0 > 0$, take a nonnegative $\eta \in L^2([0, h])$. Both η and η^2 are integrable on the finite interval, so the initial-value problem has a unique absolutely continuous solution. Multiplying the ODE by $\exp[b\int_0^t \eta(s)^2\,\mathrm{d}s]$ gives

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(m(t)\exp\left[b\int_0^t \eta(s)^2\,\mathrm{d}s\right]\right) = a\eta(t)\exp\left[b\int_0^t \eta(s)^2\,\mathrm{d}s\right].$$

Integration from 0 to h, followed by division by the terminal integrating factor, multiplies each contribution introduced at time t by

$$\frac{\exp[b\int_0^t \eta(s)^2\,\mathrm{d}s]}{\exp[b\int_0^h \eta(s)^2\,\mathrm{d}s]} = \exp\left[-b\int_t^h \eta(s)^2\,\mathrm{d}s\right] = H(h, t).$$

This is the fraction retained from t to h, as introduced in Eq. (9). Combining the initial and accumulated contributions gives

$$m(h) = m_0 H(h, 0) + a\int_0^h H(h, t)\eta(t)\,\mathrm{d}t. \tag{24}$$

This proves Eq. (9). Since $m_0 > 0$, $H(h, t) > 0$, and the integral term is nonnegative, the solution is strictly positive at every time in the interval.

B.3 Proof of the unconstrained optimal schedule

For any nonnegative $\eta \in L^2([0,h])$, with fixed $a, b, m_0, h > 0$, the positive solution satisfies

$$\frac{\mathrm{d}}{\mathrm{d}t}m(t)^2 = 2am(t)\eta(t) - 2bm(t)^2\eta(t)^2 = \frac{a^2}{2b} - 2b\left(m(t)\eta(t) - \frac{a}{2b}\right)^2$$

almost everywhere. Integrating establishes $m(h)^2 \le m_0^2 + a^2h/(2b)$. Equality requires $m(t)\eta(t) = a/(2b)$ almost everywhere. Under this condition, $\mathrm{d}(m^2)/\mathrm{d}t = a^2/(2b)$, so

$$m^\star(t) = \sqrt{m_0^2 + \frac{a^2}{2b}t}, \qquad \eta^\star(t) = \frac{a}{2b\sqrt{m_0^2 + a^2t/(2b)}}.$$

The resulting LR is nonnegative, continuous, and square-integrable on any finite interval. With $\eta_0 = a/(2bm_0)$, it is exactly Eq. (11). Substitution verifies that it solves the ODE with the prescribed initial state and attains the bound. For every candidate, the precise terminal gap is

$$m^\star(h)^2 - m(h)^2 = 2b\int_0^h \left(m(t)\eta(t) - \frac{a}{2b}\right)^2 \mathrm{d}t. \tag{25}$$

The nonnegative right-hand side proves global optimality. A zero gap uniquely determines $m^\star$ at every time and $\eta^\star$ almost everywhere. Since the endpoint h was arbitrary and the optimal curve does not depend on it, this same curve attains the alignment bound at every time. This completes the proof of the result in Section 3.2. The squared states here are squares of expected alignments, not expectations of squared alignments.

Exact discrete counterpart. For the recurrence in Eq. (6), direct algebra gives, for $m_k > 0$,

$$m_k^2 + \frac{a^2}{2b} - m_{k+1}^2 = \frac{2b\left(m_k\eta_k - a/(2b)\right)^2}{1 + 2b\eta_k^2} \ge 0.$$

After N updates, this implies $m_N^2 \le m_0^2 + Na^2/(2b)$. Choosing $\eta_k = a/(2bm_k)$ makes every single-step gap zero and yields

$$m_k^\star = \sqrt{m_0^2 + \frac{ka^2}{2b}}, \qquad \eta_k^\star = \frac{\eta_0}{\sqrt{1 + 2b\eta_0^2 k}}. \tag{26}$$

Thus the integer samples of the continuous optimum exactly maximize terminal alignment in this discrete recurrence. The N updates use $\eta_0, \ldots, \eta_{N-1}$ and produce m_N.

B.4 Geometric range of the constant-coefficient assumptions

The scalar recurrence and its continuous approximation have a broader algebraic domain than an expected cosine alignment. To identify the necessary geometric restriction, tangency and the Cauchy–Schwarz inequality give, pathwise,

$$\left|\frac{\boldsymbol{v}^\top \boldsymbol{u}_k}{R}\right| = \left|\frac{\left[\boldsymbol{v} - (\boldsymbol{v}^\top\boldsymbol{\theta}_k/R)\boldsymbol{\theta}_k/R\right]^\top \boldsymbol{u}_k}{R}\right|$$
$$\le \frac{U}{R}\sqrt{1 - \left(\frac{\boldsymbol{v}^\top\boldsymbol{\theta}_k}{R}\right)^2}.$$

The square-root function here is concave as a function of the cosine on $[-1, 1]$. Taking the conditional expectation and using Jensen's inequality therefore yields

$$a \le \sqrt{2b}\,\mathbb{E}\left[\sqrt{1 - \left(\frac{\boldsymbol{v}^\top\boldsymbol{\theta}_k}{R}\right)^2}\,\middle|\,\mathcal{F}_0\right] \le \sqrt{2b}\sqrt{1 - m_k^2}. \tag{27}$$

Thus $m_k^2 + a^2/(2b) \le 1$ is necessary at every state from which an update with the assumed positive drift is taken. For the exact discrete optimum over N updates, applying it at $k = N - 1$ requires

$$m_0^2 + \frac{Na^2}{2b} \le 1.$$

A constant positive drift cannot therefore describe an indefinitely growing physical alignment. The optimization proof is global over the prescribed nonnegative schedules of the scalar dynamics, while its interpretation as WN alignment is restricted to intervals on which the geometric assumptions remain realizable. Likewise, the continuous optimum need not satisfy the small-step condition automatically: its largest relative squared step is $2b\eta_0^2 = a^2/(2bm_0^2)$. This ratio must be small when the ODE optimum is used through the continuous approximation, although the discrete optimality identity itself does not require it.

B.5 Affine endpoint construction

Let $K \geq 1$ be the integer decay horizon from Section 3.3. For $\alpha > 0$, the inverse-square-root curve in Eq. (12) satisfies $q(0) = 1 > q(K) > 0$. Write an affine transformation of this curve as $sq(k)+d$, where s sets its scale and d its offset. An initial decay LR p and final LR $\eta_{\min}$ require

$$s + d = p, \qquad sq(K) + d = \eta_{\min}.$$

Since $q(K) < 1$, these equations uniquely determine

$$s = \frac{p - \eta_{\min}}{1 - q(K)}, \qquad d = \eta_{\min} - sq(K).$$

Substitution gives the decay sequence

$$\eta_{\min} + (p - \eta_{\min})\frac{q(k) - q(K)}{1 - q(K)}.$$

For $p > \eta_{\min}$, we have $s > 0$, so the schedule decreases monotonically between its prescribed endpoints. For any $0 \leq i < j \leq K$, subtracting the two LR values gives

$$\eta_{\mathrm{AIR}}(i) - \eta_{\mathrm{AIR}}(j) = s\big(q(i) - q(j)\big).$$

Since $p - \eta_{\min} = s[1 - q(K)]$, it follows that

$$\frac{\eta_{\mathrm{AIR}}(i) - \eta_{\mathrm{AIR}}(j)}{p - \eta_{\min}} = \frac{q(i) - q(j)}{1 - q(K)}. \tag{28}$$

Thus, each interval accounts for the same fraction of the total LR reduction as in the original curve. In particular, all per-step decreases are scaled by the same positive factor. For the continuous interpolation, $\eta'_{\mathrm{AIR}}(t) = sq'(t)$, so all decay rates are scaled by the same positive factor and their relative magnitudes are preserved. These properties establish the endpoint calibration and preservation of the relative magnitudes of decay rates. The unconstrained optimum in Section 3.2 thereby inspires the practical affine schedule.

B.6 Matching the discrete LR sum

Fix the warmup rule, decay horizon K, time scale α, terminal LR $\eta_{\min}$, and the indices at which LR values are executed. Write the warmup values as $c_i p + d_i$, where the coefficients c_i, d_i are fixed independently of the peak p. This includes linear warmup and fixed multiplicative adjustments to it. Let $\mathcal{I} \subseteq \{0, \ldots, K\}$ be the executed decay indices, and let $n_{\mathrm{d}} = |\mathcal{I}|$. For a target sum S_{d} over all executed LR values, define

$$C_W := \sum_{i \in \mathrm{warmup}} c_i, \qquad B_W := \sum_{i \in \mathrm{warmup}} d_i, \qquad F_{\mathcal{I}} := \sum_{k \in \mathcal{I}} f(k).$$

The first two quantities collect the peak-dependent and constant warmup contributions, and the third collects the peak coefficient from decay. Summing the warmup and Eq. (13) gives

$$S_{\mathrm{d}} = B_W + p(C_W + F_{\mathcal{I}}) + \eta_{\min}(n_{\mathrm{d}} - F_{\mathcal{I}}).$$

If $C_W + F_{\mathcal{I}} > 0$, the peak has the unique solution

$$p = \frac{S_{\mathrm{d}} - B_W - \eta_{\min}(n_{\mathrm{d}} - F_{\mathcal{I}})}{C_W + F_{\mathcal{I}}}. \tag{29}$$

The calibrated peak must still exceed $\eta_{\min}$ and satisfy the prescribed warmup constraints. Under the positive-coefficient condition above, a prescribed peak fixes the sum, whereas a prescribed feasible

sum determines the peak. The endpoint $k = K$ contributes $\eta_{\min}$ exactly when it belongs to $\mathcal{I}$. Thus the sum must use the executed LR values rather than infer endpoint inclusion from a logging convention.

For example, consider $W \geq 1$ linear warmup values $\eta_{\text{init}} + (p - \eta_{\text{init}})i/W$ at $i = 0, \ldots, W-1$, followed by all decay values $k = 0, \ldots, K$. For this convention,

$$C_W = \frac{W-1}{2}, \qquad B_W = \eta_{\text{init}}(W - C_W), \qquad n_{\text{d}} = K + 1.$$

There are $W + K + 1$ values, the peak occurs at the first decay step, and both decay endpoints are included. The warmup and decay regime requires $p \geq \eta_{\text{init}}$ and $p > \eta_{\min}$. For any other execution convention, the same general formula applies with its actual coefficients and index set.

C Experimental Settings and Evaluation Protocol

This appendix details the experimental setup in Sections 4 and 5.

C.1 Training settings and comparison controls

Table 3 summarizes the training settings for the smaller-scale scheduler experiments. The larger-scale comparison uses 790 M activated and 35.7 B total parameters, 500 B training tokens, and 168 hours per run on 32 GPUs. Within each fixed-norm scheduler comparison, the training setting is held fixed and the LR schedule is varied.

Table 3: Training settings for the smaller-scale scheduler comparisons. The peak LRs specify the main comparison between cosine and AIR; other schedule peaks are calibrated to the same cumulative LR.

Configuration	Value
Activated parameters	110 M
Total parameters	4.7 B
Training tokens	128 B
Optimizer	Muon
Global batch size	128
Normalization	Frobenius norm
Normalization epsilon	10^{-8}
Scheduled optimizer updates	32,047
Warmup updates	3,040
Minimum LR	10^{-5}
Cosine peak LR	1.07×10^{-3}
AIR peak LR	1.946367×10^{-3}
AIR decay time scale	$\alpha = K/8$
GPUs per run	16
Wall-clock per complete run	48 hours

The LR curves in Figure 3(a) show a linear warmup illustration; the comparison controls concern the complete training schedule, including its batch-size ramp. The observed terminal training loss step for these runs is 32,046. Logging indices and scheduled endpoints are therefore distinguished when pairing losses or evaluating a schedule sum.

Within a fixed-norm comparison, the model, token budget, optimizer settings, batch-size schedule, constrained parameter blocks, and their radii are held fixed. The complete optimizer-step LR sequence is the object being compared. Warmup and the last executed update are included in its LR sum. For matched-sum comparisons with cosine, peak calibration follows Appendix B.6. This leaves peak LR free to differ between schedules while holding the specified sum, horizon, warmup, and terminal LR fixed.

Stage replacements require a separate accounting of the resulting sequence. Replacing only one interval can alter its total LR sum or create a jump at the boundary. These properties follow from the actual combined curve and cannot be inferred from the controls of the original full schedules. A shared late schedule means equal LR values at every shared optimizer step, not merely equal endpoints or equal integrated LR. Likewise, varying an entire decay exponent is a different intervention from fixing the early schedule and replacing only the late part.

C.2 Compared decay shapes

Let $k = 0, \dots, K$ index the decay stage as in Eq. (12), and write $\tau = k/K$. Thus $\tau \in [0, 1]$ is the fraction of the decay stage completed. Table 4 gives the raw shape $q(k)$ for each schedule. All shapes satisfy $q(0) = 1$ and use the same affine endpoint transformation and LR interpolation in Eqs. (12)–(13), with their own q substituted for the inverse-square-root curve. The peak is then calibrated to match the cosine baseline's cumulative LR, including warmup, following Appendix B.6. This holds the warmup duration and terminal LR fixed while allowing the peak to differ. The raw inverse-square-root reference in Figure 3(a) is excluded from these controls.

Table 4: Raw decay shapes before the common affine transformation, and the shape parameter selected for the smaller-scale rows of Table 1. The time scale $\alpha > 0$, decay exponent $\beta > 0$, which is fixed for the two power families and rises to β_{end} for variable power, the terminal angle $0 < \theta_1 \le \pi$, and the exponential decay parameter $\rho > 0$ control the respective curve shapes. A dash marks a shape with no free parameter.

Shape	$q(k)$, with $\tau = k/K$	Selected
Cosine	$\frac{1}{2}[1 + \cos(\pi\tau)]$	–
Linear	$1 - \tau$	–
Half-cosine	$\cos(\pi\tau/2)$	–
Rear-cosine	$1 - \sin(\pi\tau/2)$	–
Cycloid	$\cos\theta, \quad \theta - \sin\theta = \tau(\theta_1 - \sin\theta_1)$	$\theta_1 = 3\pi/4$
Double-cosine	$q_{\text{dc}}(\tau)$, defined below	–
Power of $1 - \tau$	$(1 - \tau)^\beta$	$\beta = 2$
Power of τ	$1 - \tau^\beta$	$\beta = 0.2$
Logarithmic	$1 - \log(1 + k/\alpha)$	$\alpha = K/32$
Exponential	$\exp(-\rho\tau)$	$\rho = 3.594$
Inverse square	$(1 + k/\alpha)^{-2}$	$\alpha = K/2$
Inverse time	$(1 + k/\alpha)^{-1}$	$\alpha = 0.243K$
Variable power	$(1 + k/\alpha)^{-[1/2 + (\beta_{\text{end}} - 1/2)\tau]}$	$\alpha = K/8$, $\beta_{\text{end}} = 1.771$
Inverse square root (AIR)	$(1 + k/\alpha)^{-1/2}$	$\alpha = K/8$

The square-root shape used outside Table 1 is $1 - \tau^\beta$ with $\beta = 1/2$, and the quadratic shape is $(1 - \tau)^\beta$ with $\beta = 2$. In Table 1 each schedule is named by the shape in this table, written up to the affine transformation: the time variable t absorbs an affine rescaling, so $t = \tau$ for the polynomial shapes and $t = 1 + k/\alpha$ for the inverse-power and logarithmic ones, and $t = \rho\tau$ for the exponential shape, and the endpoint normalization is left implicit. Cosine, Half-cosine, Rear-cosine, and Cycloid are not separated by such a formula and keep their names there.

For the symmetric double-cosine schedule, the two cosine segments join at $\tau = 1/2$ and $q_{\text{dc}} = 1/2$:

$$q_{\text{dc}}(\tau) = \begin{cases} \frac{3}{4} + \frac{1}{4}\cos(2\pi\tau), & 0 \le \tau \le \frac{1}{2}, \\ \frac{1}{4} + \frac{1}{4}\cos(2\pi\tau - \pi), & \frac{1}{2} < \tau \le 1. \end{cases}$$

For Cycloid, $\theta \in [0, \theta_1]$ is the unique solution of the relation in the table. For variable power, the exponent varies linearly from $1/2$ to β_{end}.

C.3 Loss summaries

Tables 1 and 2, and the robustness comparison, use the same terminal loss convention. For training loss, take the last 100 finite logged values in step order, denoted $x_1, \dots, x_{100}$, initialize $z_1 = x_1$,

and recursively compute

$$z_j = 0.99 z_{j-1} + 0.01 x_j, \qquad j = 2, \ldots, 100. \tag{30}$$

The reported value is z_{100}; evaluation loss is the last finite logged evaluation value without smoothing. The last 100 observations need not span exactly 100 optimizer steps if a log has gaps. The smaller-scale comparison ends with training loss at step 32,046 and evaluation loss at step 32,000; the corresponding larger-scale steps are 62,602 and 62,000. These are terminal summaries of individual trajectories, not averages across independent training seeds.

Training restarts are stitched by optimizer step, with the later-started segment taking precedence at duplicate steps. For reference loss curves, we first pair raw losses at common optimizer steps and subtract the specified reference, then apply a debiased EMA with decay 0.99 to the complete paired sequence. For paired differences $d_1, \ldots, d_j$, the plotted value is

$$\frac{\sum_{i=1}^{j} 0.99^{j-i} d_i}{\sum_{i=1}^{j} 0.99^{j-i}}.$$

Smoothing advances once per retained paired observation; missing steps are not filled by interpolation. The terminal reference loss points in Figure 6 use the final value of this curve. Figure 5(c) does not use this curve: for each of the eight runs it reports the terminal training loss summary of Eq. (30), and its stage-swap gaps are differences of these terminal values. No independent-seed confidence interval is inferred from adjacent training steps.

C.4 Downstream benchmarks

Table 1 reports performance on MMLU (Hendrycks et al., 2021a), MMLU-Pro (Wang et al., 2024), AGIEval (Zhong et al., 2024), WinoGrande (Sakaguchi et al., 2020), BBH (Suzgun et al., 2023), DROP (Dua et al., 2019), GSM8k (Cobbe et al., 2021), MATH (Hendrycks et al., 2021b), HumanEval (Chen et al., 2021), MBPP+ (Liu et al., 2023), and TriviaQA (Joshi et al., 2017).

C.5 Hyperparameter sweeps and LR sensitivity

For the robustness sweep in Section 4.3, we report each family's LR sensitivity. For a family with configurations $i = 1, \ldots, n$ of terminal loss L_i and peak LR η_i, the LR sensitivity is

$$S = \frac{1}{W_{\log}} \left(\frac{1}{n} \sum_{i=1}^{n} L_i - \min_i L_i \right), \qquad W_{\log} = \ln \frac{\max_i \eta_i}{\min_i \eta_i}, \tag{31}$$

the average loss above the family's best configuration per e-fold of peak LR: the numerator is the mean terminal loss above the family's best measured configuration, and dividing by the log-span $W_{\log}$ of the swept peak LR normalizes for search width. This is a descriptive statistic over each family's measured range and is not extrapolated beyond it. Table 5 reports S on training and evaluation loss.

Table 5 gives the configuration counts and logarithmic spans. Search ranges and configuration counts differ across families; the normalization by $W_{\log}$ does not make their tuning budgets identical. The fitted curves visualize the measured points, whereas S is computed directly from those points. For a family with n configurations, the fit has degree $\min(3, n-1)$; fits with at least three points are constrained to be convex over the observed peak-LR interval.

C.6 Stage swaps

Figure 5 compares AIR with Cycloid at $\theta_1 = \pi$, which shares its peak LR, warmup, terminal LR, and full-schedule cumulative LR. This comparator differs from the evaluation-selected Cycloid at $\theta_1 = 3\pi/4$ in Table 1. The two intersections occur at approximately global steps 4,346 and 16,219 and define the early, middle, and late intervals. Each replacement follows the donor schedule only within the named interval and the recipient schedule outside it. Matching the cumulative LR of the two original schedules does not imply that an individual stage replacement preserves that sum. The terminal summary and reference loss smoothing are specified in Appendix C.3.

Table 5: **AIR is the least sensitive to the peak LR on training loss.** Sensitivity is the mean loss above a family's best per e-fold of peak LR (Eq. (31)).

Schedule	Configs	LR range	LR sensitivity (↓)	
		(e-folds)	Train	Eval
Cycloid	3	0.3461	0.0022	**0.0017**
$(1-t)^\beta$	3	0.4219	0.0160	0.0126
$1-t^\beta$	3	0.5630	0.0023	0.0039
$1-\log t$	3	0.2182	0.0047	0.0054
e^{-t}	4	0.7081	0.0113	0.0110
$1/t^2$	3	0.5630	0.0032	0.0021
$1/t$	5	0.5874	0.0019	0.0027
$t^{-\beta(t)}$	2	0.0427	0.0230	0.0234
$1/\sqrt{t}$ (AIR)	5	0.7506	**0.0018**	0.0018

Table 6: Measured shape settings and peak-LR ranges for the smaller-scale sweep. Decimal shape settings are rounded to three places; the other values are exact. Counts and logarithmic spans appear in Table 5.

Family	Shape settings	Peak LR (10^{-3})
Cycloid	$\theta_1 \in \{\pi/2, 3\pi/4, \pi\}$	1.377–1.946
$(1-t)^\beta$	$\beta \in \{2, 3, 4\}$	1.528–2.330
$1-t^\beta$	$\beta \in \{1/2, 1/3, 1/5\}$	1.528–2.683
Logarithmic	$\alpha/K \in \{1/8, 1/16, 1/32\}$	1.541–1.916
Exponential	$\rho \in \{2, 3.594, 5.549, 7\}$	1.489–3.023
Inverse square	$\alpha/K \in \{1, 1/2, 1/4\}$	1.528–2.683
Inverse time	$\alpha/K \in \{0.864, 0.400, 0.243, 0.166, 0.125\}$	1.382–2.487
Variable power	$\beta_{\text{end}} \in \{1.771, 2\},\ \alpha = K/8$	1.946–2.031
AIR	$\alpha/K \in \{0.557, 1/4, 1/8, 1/16, 1/32\}$	1.382–2.928

C.7 LATE DECAY EXPONENTS

The exponent interventions in Eq. (14) switch at global step 25,955, corresponding to $K_s/K \simeq 0.79$ of the decay stage after warmup. They use $\gamma \in \{0.5, 0.75, 1, 2, 3\}$, with the same earlier schedule, junction LR, and configured terminal LR. Their cumulative LR is not exactly matched. Trapezoidal integration of the recorded LR traces gives deviations below 1 % relative to the AIR reference. Figure 6 reports same-step loss differences to the full-run AIR reference with the full-sequence EMA convention in Appendix C.3.

D BOUNDS FOR EARLY- AND LATE-SCHEDULE DIFFERENCES

D.1 COMMON SETTING

We use the continuous relaxation of the decay-step coordinate $k \in [0, K]$, with a fixed split $0 \le K_s < K$. For $i \in \{1, 2\}$, the expected alignment m_i follows the dynamics from Section 3,

$$\dot{m}_i(k) = a\eta_i(k) - b\eta_i(k)^2 m_i(k), \qquad m_i(0) = m_0 \ge 0,$$

with common $a, b > 0$ and bounded, measurable, nonnegative schedules η_i. Dots denote derivatives with respect to this continuous step coordinate. The expectations retain the same initial conditioning $\mathcal{F}_0$ as Eq. (5); the conditioning is not reset at K_s. Write $\Delta m = m_1 - m_2$ and $\Delta\eta = \eta_1 - \eta_2$. A shared early schedule and the common initial alignment give $\Delta m(K_s) = 0$ by uniqueness of the linear ODE.

For each schedule, the retention factor from Eq. (9) is

$$\begin{aligned} H_i(K, k) &= \exp\left[-b \int_k^K \eta_i(s)^2 \, \mathrm{d}s\right], \\ H_{12}(K, k) &= \sqrt{H_1(K, k) H_2(K, k)}. \end{aligned} \tag{32}$$

The second factor retains a difference under the average retraction coefficient $b(\eta_1^2 + \eta_2^2)/2$. When the late schedules coincide, $H_1 = H_2 = H_{12} = H$ on that interval.

D.2 VARYING EARLY-STAGE SCHEDULES

If $\eta_1 = \eta_2 = \eta$ on $[K_s, K]$, subtracting the state equations gives

$$\frac{\mathrm{d}\Delta m(k)}{\mathrm{d}k} = -b\eta(k)^2 \Delta m(k).$$

Multiplication by $\exp[b \int_{K_s}^k \eta(s)^2 \, \mathrm{d}s]$ makes the derivative of the product zero. Evaluating this constant at K_s and K yields

$$\Delta m(K) = H(K, K_s) \Delta m(K_s). \tag{33}$$

The argument never divides by Δm, so it also covers zero or negative entering differences. Since $0 < H(K, K_s) \le 1$, the sign is preserved and the magnitude cannot increase, proving Eq. (15). Ignoring the effects of stochastic noise, the expected-alignment ordering is therefore preserved along the shared tail.

Early schedules with equal integrated LR. Assume additionally $0 \le m_i(k) \le 1$ on $[0, K_s]$ and $\int_0^{K_s} \eta_1(k) \, \mathrm{d}k = \int_0^{K_s} \eta_2(k) \, \mathrm{d}k$. Integrating the two state equations cancels the common initial alignment and the cumulative new-alignment terms:

$$\Delta m(K_s) = -b \int_0^{K_s} \left[\eta_1(k)^2 m_1(k) - \eta_2(k)^2 m_2(k)\right] \mathrm{d}k.$$

Each retraction integral is nonnegative and at most $\int_0^{K_s} \eta_i(k)^2 \, \mathrm{d}k$. Their absolute difference is therefore bounded by the larger integral, giving

$$|\Delta m(K)| \le bH(K, K_s) \max_{i \in \{1,2\}} \int_0^{K_s} \eta_i(k)^2 \, \mathrm{d}k. \tag{34}$$

Equal integrated LR cancels the cumulative new-alignment difference but does not equate the retraction corrections.

Discrete counterpart. Take K_s, K to be integers and let $\Delta m_k = m_{1,k} - m_{2,k}$ for the recurrences in Eq. (6). If they share η_k for $k = K_s, \dots, K-1$, their difference obeys $\Delta m_{k+1} = \Delta m_k / \sqrt{1 + 2b\eta_k^2}$. Iteration gives

$$\Delta m_K = \Delta m_{K_s} \prod_{k=K_s}^{K-1} (1 + 2b\eta_k^2)^{-1/2}.$$

This exact discrete relation has the same sign-preserving attenuation property without replacing the product by a continuous exponential.

D.3 Varying late-stage schedules

On $[K_s, K]$, the algebraic identity

$$\eta_1^2 m_1 - \eta_2^2 m_2 = \frac{\eta_1^2 + \eta_2^2}{2} \Delta m + \frac{(\eta_1 + \eta_2)(m_1 + m_2)}{2} \Delta\eta$$

gives the exact difference equation

$$\frac{\mathrm{d}\Delta m}{\mathrm{d}t} = -\frac{b}{2}(\eta_1^2 + \eta_2^2)\Delta m + \left[a - \frac{b}{2}(\eta_1 + \eta_2)(m_1 + m_2)\right] \Delta\eta. \tag{35}$$

Using the integrating factor associated with the first term yields

$$\begin{aligned}\Delta m(K) = {} & H_{12}(K, K_s)\Delta m(K_s) \\ & + \int_{K_s}^{K} H_{12}(K, k) \left[a - \frac{b}{2}(\eta_1(k) + \eta_2(k))(m_1(k) + m_2(k))\right] \Delta\eta(k)\, \mathrm{d}k.\end{aligned} \tag{36}$$

The first term propagates the entering difference, and the second accumulates differences caused by the later LR choices. This identity is exact within the ODE and does not assume a small schedule perturbation.

Choose common bounds $0 \le \eta_i \le P$ and $0 \le m_i \le M$ on the late interval. Here P bounds both schedules; it does not require them to have equal peaks. The integral solution preserves nonnegative states, and $\dot{m}_i \le aP$, so $M = \max_i m_i(K_s) + aP(K - K_s)$ is a valid finite choice. When the alignments stay in their physical range, $M = 1$ also suffices. Since

$$0 \le \frac{b}{2}(\eta_1 + \eta_2)(m_1 + m_2) \le 2bPM,$$

the response coefficient in brackets lies between $a - 2bPM$ and a. Its absolute value is bounded by

$$L_{12} = \max\{a, |a - 2bPM|\}.$$

This preserves the cancellation between alignment gain and retraction before taking absolute values. When only the late-stage schedule varies, the shared early schedule gives $\Delta m(K_s) = 0$. Applying the triangle inequality to Eq. (36) therefore gives

$$|\Delta m(K)| \le L_{12} \int_{K_s}^{K} H_{12}(K, k) |\Delta\eta(k)|\, \mathrm{d}k,$$

which proves Eq. (16).

If both stages vary, retain the first term in Eq. (36) and apply the same bound to its integral. Since $H_{12} > 0$, this gives

$$|\Delta m(K)| \le H_{12}(K, K_s)|\Delta m(K_s)| + L_{12} \int_{K_s}^{K} H_{12}(K, k) |\Delta\eta(k)|\, \mathrm{d}k,$$

proving Eq. (17) without requiring a common entering alignment. Together, these comparisons characterize how LR schedules promote expected directional alignment under common coefficients, thereby providing a mechanism for improved training loss.

D.4 Slope bounds and endpoint information

Assume $\Delta m(K_s) = 0$, absolutely continuous late schedules, and a common terminal LR $\eta_1(K) = \eta_2(K)$. Let $\varepsilon_1 = \operatorname{ess\,sup}_{[K_s,K]} |\dot\eta_1 - \dot\eta_2| < \infty$. Subtracting the terminal values and integrating the derivative difference gives

$$\Delta\eta(k) = -\int_k^K (\dot\eta_1(s) - \dot\eta_2(s))\,\mathrm{d}s, \qquad |\Delta\eta(k)| \le \varepsilon_1(K-k).$$

Substitution into Eq. (16), followed by $H_{12} \le 1$, yields

$$|\Delta m(K)| \le L_{12}\varepsilon_1 \int_{K_s}^K H_{12}(K,k)(K-k)\,\mathrm{d}k \tag{37}$$

$$\le \frac{L_{12}\varepsilon_1(K-K_s)^2}{2}. \tag{38}$$

If the schedules also share their LR at K_s, integration from the left endpoint gives $|\Delta\eta(k)| \le \varepsilon_1 \min\{k-K_s, K-k\}$. The integral of this envelope is $\varepsilon_1(K-K_s)^2/4$, hence

$$|\Delta m(K)| \le \frac{L_{12}\varepsilon_1(K-K_s)^2}{4}.$$

A shared early schedule alone does not impose this extra endpoint condition if a jump is allowed at the switch.

Applicability to the late-stage ablation. The schedules in Eq. (14) are bounded and absolutely continuous for every $\gamma > 0$, so the LR-difference bound in Eq. (16) applies under the stated alignment dynamics. For $0 < \gamma < 1$, however, the continuous terminal derivative is unbounded. Indeed, writing the positive junction-to-floor difference directly, differentiation gives

$$\dot\eta_\gamma(k) = -\frac{\gamma[\eta_{\mathrm{AIR}}(K_s) - \eta_{\min}]}{K-K_s}\left(\frac{K-k}{K-K_s}\right)^{\gamma-1}, \qquad k < K.$$

Its magnitude diverges at K when $\gamma < 1$, is constant when $\gamma = 1$, and vanishes there when $\gamma > 1$. A comparison with the smooth AIR tail, or with a distinct exponent, therefore need not satisfy the finite ε_1 condition. Accordingly, comparisons involving $0 < \gamma < 1$ are governed by the general LR-difference bound rather than its slope-based corollary. This bound connects the exponent intervention to expected directional alignment, complementing the empirical loss comparison in Section 5.3.

Knowing only the terminal slope. For twice continuously differentiable schedules with a common terminal LR and $\Delta m(K_s) = 0$, define $\delta_K = |\dot\eta_1(K) - \dot\eta_2(K)|$ and $C_2 = \sup_{[K_s,K]} |\ddot\eta_1 - \ddot\eta_2|$. These measure the terminal slope difference and the curvature difference over the full interval. Taylor expansion with integral remainder gives

$$|\Delta\eta(k)| \le \delta_K(K-k) + \frac{C_2}{2}(K-k)^2.$$

Substitution into the LR-difference bound yields

$$|\Delta m(K)| \le L_{12}\left(\frac{\delta_K(K-K_s)^2}{2} + \frac{C_2(K-K_s)^3}{6}\right).$$

Equal terminal slopes remove the first contribution, while the curvature term captures differences across the interior of the schedules.

D.5 Affine power curves and the scope of the bounds

As an auxiliary example on $[K_s, K]$, fix a common initial LR $P > \eta_{\min} \ge 0$ and a time scale $\alpha_{\mathrm{tail}} > 0$. For exponent $\beta > 0$, define

$$q_\beta(k) = \left(1 + \frac{k-K_s}{\alpha_{\mathrm{tail}}}\right)^{-\beta}, \qquad \eta_\beta(k) = \eta_{\min} + (P-\eta_{\min})\frac{q_\beta(k) - q_\beta(K)}{1 - q_\beta(K)}.$$

This affine inverse-power family shares its initial and terminal LRs; $\beta = 1/2$ gives an inverse-square-root shape. It is distinct from the terminal-power variants indexed by γ in Eq. (14). Direct differentiation gives

$$|\dot{\eta}_\beta(k)| = \frac{P - \eta_{\min}}{\alpha_{\text{tail}}} \frac{\beta\big(1 + (k - K_s)/\alpha_{\text{tail}}\big)^{-\beta-1}}{1 - \big(1 + (K - K_s)/\alpha_{\text{tail}}\big)^{-\beta}}.$$

The terminal slope follows by setting $k = K$. On a compact interval of positive exponents, the slope is jointly continuous in β and k, so close exponents give uniformly close slopes. Combined with the preceding bounds, this continuity shows that nearby exponents produce nearby terminal expected alignments from a common entering state. When the early stage also varies, Eq. (36) incorporates the resulting entering-state difference.

D.6 Infinitesimal sensitivity to the time of an LR change

For an admissible family $\eta_\varepsilon = \eta + \varepsilon\delta\eta$ with bounded perturbation $\delta\eta$ and a common initial state, let $\delta m = \partial_\varepsilon m_\varepsilon|_{\varepsilon=0}$. It is the first-order change in expected alignment induced by the LR perturbation. Differentiating the state equation gives

$$\frac{\mathrm{d}\delta m}{\mathrm{d}t} = -b\eta^2\delta m + (a - 2b\eta m)\delta\eta, \qquad \delta m(0) = 0.$$

This differentiation is justified directly from the integral solution: bounded η and $\delta\eta$ on the finite interval uniformly bound both the integrand and its derivative for sufficiently small admissible ε. Differentiation under the integral sign therefore gives an absolutely continuous δm satisfying this linear equation. Applying the same integrating factor gives

$$\delta m(K) = \int_0^K H(K, k)\big[a - 2b\eta(k)m(k)\big]\delta\eta(k)\,\mathrm{d}k.$$

Later perturbations are retained more strongly through $H(K, k)$, while the factor $a - 2b\eta(k)m(k)$ determines their local alignment gain. The resulting sensitivity kernel identifies when and where an LR change most strongly influences terminal alignment.