K-SCALING: CRITICAL BATCH SIZE GOVERNS LEARNING RATE SCHEDULING ACROSS TOKEN BUDGETS IN LLM PRETRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

We study learning rate (LR) scheduling in large language model pretraining by jointly tuning peak LR and decay ratio, the fraction of post-warmup training allocated to terminal decay. Using the functional scaling law framework with an admissibility constraint on peak LR, we establish *K-scaling collapse*: when expressed through the normalized batch size (BS) $x = B/B_c(D)$, both optimal hyperparameters follow budget-independent scaling profiles, where D is the token budget and $B_c(D)$ is a shared critical batch size (CBS). The CBS marks a *phase transition* from pure decay with increasing peak LR in the *subcritical* regime ($x < 1$) to warmup–stable–decay (WSD) with saturated peak LR and decreasing decay ratio in the *supercritical* regime ($x > 1$). Experiments on dense and MoE models spanning 0.1B–1B parameters, trained with MuonH, confirm approximate K-scaling collapse and support a practical four-parameter model for jointly predicting peak LR and decay ratio. Fitted on small token budgets, the model extrapolates to budgets up to $10\times$ the largest fitting budget, yielding schedules with final losses close to grid-search optima without further tuning. The theory also explains why longer training favors larger decay ratios at fixed-BS training, helping reconcile differing schedule preferences in prior studies.

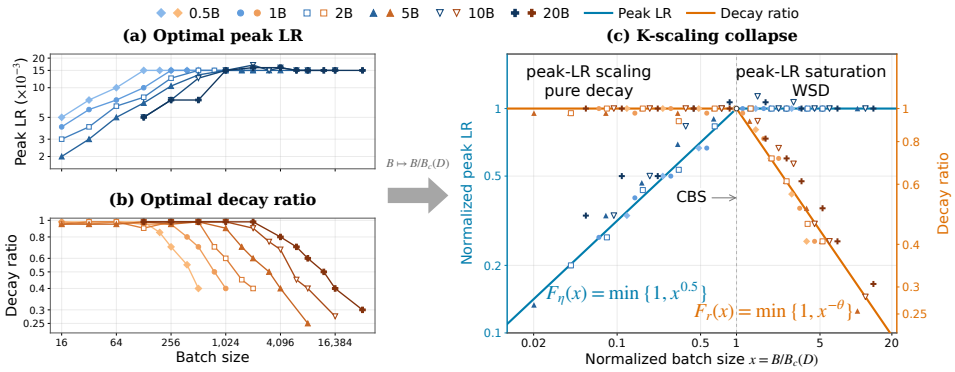


Figure 1: K-scaling collapse in LLM pretraining. Jointly tuned peak LRs and decay ratios for Llama-120M trained with MuonH at token budgets $D \in \{0.5, 1, 2, 5, 10, 20\}$B. **(a,b)** Both schedule parameters exhibit budget-dependent transitions. **(c)** A shared CBS $B_c(D) = CD^b$ approximately aligns these transitions and collapses the optima onto their respective scaling profiles. Blue denotes peak LR and orange denotes decay ratio; marker shapes and shades distinguish budgets. Solid lines show the joint fit.

1 INTRODUCTION

Learning rate (LR) scheduling is critical to efficient large language model (LLM) pretraining, affecting the loss achievable within a given training budget. Common choices include pure decay schedules, such as cosine (Loshchilov & Hutter, 2016; Touvron et al., 2023) and linear decay (Defazio et al., 2023; Bergsma et al., 2025b), which decrease the LR throughout the post-warmup phase,

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.

and warmup–stable–decay (WSD) schedules (Hu et al., 2024), which maintain a constant LR after warmup before a short terminal decay phase. However, existing studies reach different conclusions about their relative performance. For instance, Semenov et al. (2025) report that Muon favors WSD, whereas Kimi K3 reports lower loss with pure cosine decay after separately optimizing each schedule's hyperparameters (Team et al., 2026).

We identify batch size (BS) as a key factor governing this preference. Holding the model, token budget, and optimizer fixed, we observe that pure decay achieves lower loss at smaller BS, whereas WSD performs better at larger BS, even when peak LR is independently tuned for each schedule. Figure 2 illustrates this reversal for Llama-1B. This reversal motivates treating peak LR and decay duration as joint design variables that depend on the training budget and BS.

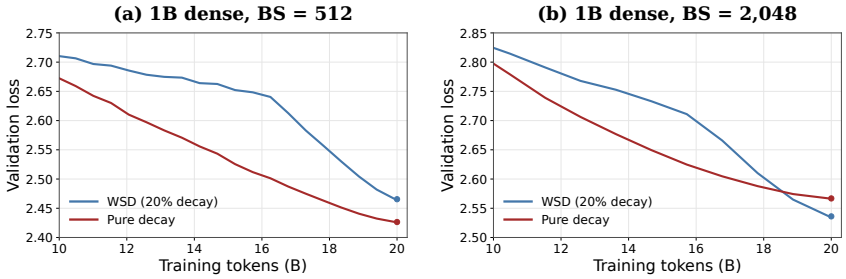


Figure 2: Batch size changes schedule preference. Llama-1B trained on FineWeb-Edu with MuonH. Peak LR is tuned independently for each schedule.

We fix the terminal decay profile, such as linear decay, and parameterize the schedule by its peak LR η_{peak} and decay ratio r. Omitting warmup for this formulation, $r \in [0, 1]$ is the fraction of training steps allocated to terminal decay: $r = 1$ gives pure decay, while $0 < r < 1$ corresponds to WSD. For a fixed model and optimizer configuration, we seek to characterize

$$(\eta^{\star}_{\text{peak}}(D, B), r^{\star}(D, B)) = \underset{\substack{0<\eta_{\text{peak}}\leq\eta_{\text{crit}} \\ r\in[0,1]}}{\arg\min} \mathcal{L}_{D,B}(\eta_{\text{peak}}, r), \tag{1.1}$$

where $\mathcal{L}_{D,B}$ denotes the final loss after training with BS B for a token budget D. The admissibility constraint $\eta_{\text{peak}} \leq \eta_{\text{crit}}$ accounts for limits on usable peak LRs, such as instability at excessively large values.

To characterize these joint optima, we use the functional scaling law (FSL) framework (Li et al., 2025; 2026) to model how $\mathcal{L}_{D,B}$ depends on the LR schedule. At a fixed token budget, increasing BS reduces gradient noise but also causes fewer optimizer updates. FSL captures these competing effects through signal learning and noise accumulation, allowing us to analyze how their balance shapes the optimal peak LR and decay ratio. Originally derived for kernel regression, FSL has also demonstrated empirical accuracy in modeling LR-dependent loss dynamics in LLM pretraining.

Theoretical results: K-scaling collapse. Our finding is that optimal peak LR and decay ratio across data budgets can be organized by a shared, data-dependent *critical batch size* (CBS), $B_c(D) = CD^b$ with $b \in (0, 1)$. Defining the *normalized BS* $x = B/B_c(D)$, both optimal hyperparameters collapse onto scaling profiles that depend *only* on x:

$$\eta^{\star}_{\text{peak}}(D, B) = F_{\eta}(x), \qquad r^{\star}(D, B) = F_r(x). \tag{1.2}$$

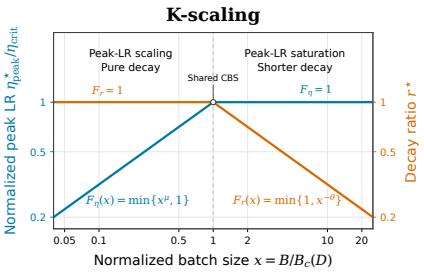
Specifically, the peak-LR scaling profile takes the form

$$F_{\eta}(x) = \eta_{\text{crit}} \min\{1, x^{\mu}\}, \qquad \mu > 0,$$

where η_{crit} is the critical LR imposed by the training-stability constraint. The decay-ratio scaling profile satisfies $F_r(x) = 1$ for $x \leq 1$ and decreases continuously toward zero for $x > 1$. Thus, the two optimal hyperparameters undergo a shared phase transition at the critical BS, i.e., $x = 1$:

- For $x < 1$, pure decay is optimal, and optimal peak LR increases as a power of x.
- For $x > 1$, WSD is optimal, and optimal peak LR reaches the critical LR, while the optimal decay ratio decreases with x.

This shared transition provides a mechanism consistent with the batch-dependent schedule preference observed in Figure 2. The precise exponent and decay-ratio profile follow from the FSL analysis; the central structural prediction is that critical BS governs the tuning of both peak LR and decay ratio. The two scaling profiles form a sideways K-like pattern in log–log coordinates, as illustrated in the figure on the right. We therefore call this joint structure *K-scaling collapse*.



Empirical results: From K-scaling to practical tuning. We test K-scaling in LLM pretraining on FineWeb-Edu with MuonH, using Llama-120M, QwenMoE-250M, and Llama-1B. Across token budgets and batch sizes, we jointly search over peak LR and decay ratio to evaluate the predicted scaling structure and its practical use for schedule tuning. Our main empirical findings are:

- **Practical parameterization.** The scaling collapse predicted by FSL also emerges in LLM pretraining (Figure 1). Empirical calibration across the dense and MoE models shows that a square-root LR profile ($\mu = 1/2$) and a clipped-power decay-ratio profile, $F_r(x) \approx \min\{1, x^{-\theta}\}$, closely capture the scaling behavior of LLM pretraining. These findings turn the theoretical structure into a simple tuning rule, without requiring the implicit FSL profile to be solved. Moreover, allowing separate CBSs further improves collapse accuracy, revealing the tradeoff between a shared-CBS description and a more flexible fit.
- **Schedule transfer across budgets.** Fitting the K-scaling model on small-budget runs predicts both peak LR and decay ratio at larger budgets without further tuning, with predictions close to joint grid-search optima. The resulting schedules achieve near-optimal final losses and outperform both pure-decay and vanilla WSD (20% decay) baselines.
- **Interpreting existing tuning practices.** We revisit prior findings on LR tuning and schedule selection through the lens of K-scaling, showing how it connects empirical observations across training configurations and helps explain seemingly conflicting schedule preferences.

In summary, we find it sufficient to use the following K-scaling model:

K-scaling model: $(D, B) \mapsto (\eta_{\text{peak}}, r^*)$**.** Let $x = B/B_c$ with $B_c = CD^b$. Then,

$$\eta^\star_{\text{peak}}(D, B) = \eta_{\text{crit}} \min\{1, x^{0.5}\}, \qquad r^\star(D, B) = \min\{1, x^{-\theta}\}. \tag{1.3}$$

Here $(C, b, \eta_{\text{crit}}, \theta)$ are the parameters to be fitted.

Remark 1.1 (A roofline perspective). K-scaling admits an interpretation analogous to the roofline model (Williams et al., 2009): an active constraint changes how further progress can be achieved. Below the CBS, fewer optimizer updates are compensated by increasing peak LR. Once the critical LR η_{crit} is reached, further adaptation occurs through shortening decay, allowing more accumulated LR within the available updates. The two scaling profiles thus reflect coupled responses to the same constraint, giving rise to oppositely oriented roofline-like structures.

2 Related Work

Functional scaling laws. FSL describes population loss as a functional of the learning rate schedule through signal learning and noise forgetting (Li et al., 2025). Originally developed for SGD in power-law kernel regression, it has enabled analyses of optimal learning rate schedules (Li et al., 2026), batch-size schedules (Wang et al., 2026a), and joint data-quality and batch-size scheduling (Zhu et al., 2026b). Related analyses also inform optimizer design, providing a theoretical explanation for the benefits of multiscale momentum in MuonM (Zhu et al., 2026a). Building on FSL, we jointly optimize peak LR and decay ratio under a peak-LR stability constraint, deriving a shared critical batch scale and the associated schedule transition. Our Muon experiments assess this structure using empirically determined scaling profiles.

Critical batch size. Prior work studies critical batch size (CBS) and its connection to learning-rate scaling and diminishing returns from data parallelism (McCandlish et al., 2018; Li et al., 2024;

Merrill et al., 2025). Studies of LLM pretraining further find that CBS depends primarily on data size rather than model size (Zhang et al., 2025; Bergsma et al., 2025a). Our CBS instead marks a joint transition in schedule design: peak LR reaches its ceiling as the optimal decay ratio departs from pure decay. We characterize how this CBS varies with token budget and organizes both schedule parameters, without assuming equivalence to efficiency-based definitions of CBS.

3 Experimental Setup and Functional Scaling Laws

3.1 Experimental Setup

We pretrain three models on FineWeb-Edu (Penedo et al., 2024): two Llama-style dense models (Touvron et al., 2023), denoted Llama-120M and Llama-1B, and a QwenMoE model (Yang et al., 2024) with approximately 250M total and 100M active parameters, denoted QwenMoE-250M. All models use Muon (Jordan et al., 2024; Liu et al., 2025) with Hyperball (MuonH) (Wen et al., 2026). Table 1 summarizes the training configurations. Within each schedule comparison, we hold the model and all other optimizer settings fixed.

All runs allocate 2% of training steps to warmup. The decay ratio r denotes the fraction of post-warmup steps allocated to terminal decay, so $r = 1$ corresponds to pure decay. Our main experiments use linear terminal decay, with other decay profiles examined separately. For Llama-120M and QwenMoE-250M, we jointly search over peak LR and decay ratio at each (D, B), refine the grid around low-loss configurations, and select the evaluated pair with the lowest final validation loss. To limit computational cost, the Llama-1B transfer experiments fix peak LR at 0.01 and search only the decay ratio. Architecture details, optimizer settings, and search grids are provided in Appendix F.

Remark 3.1 (Why norm control?). Recent work identifies effective LR (ELR), rather than nominal LR, as an intrinsic LR governing LLM training dynamics (Liu et al., 2026). Our preliminary experiments likewise show that K-scaling describes optimal ELR more accurately than optimal nominal LR. We therefore focus on ELR scaling and use MuonH to keep ELR proportional to nominal LR through norm control. Under this setting, tuning nominal LR directly controls ELR.

Table 1: Training configurations. BS is measured in sequences per update.

Model	Token budgets (billions)	BS range	Sequence length
Llama-120M	$0.5, 1, 2, 5, 10, 20$	16–32,768	1,024
QwenMoE-250M	$0.5, 1, 2, 5, 10, 20$	16–16,384	1,024
Llama-1B	$1, 2, 5, 20$	16–2,048	4,096

3.2 Intrinsic Time and Functional Scaling Laws

We analyze LR schedules in a continuous-step model. Let $\eta(z)$ denote the LR at training step z, and define the intrinsic time

$$t(z) = \int_0^z \eta(v)\,\mathrm{d}v, \qquad T = t(N),$$

where N is the total number of updates. Intrinsic time measures accumulated LR, the continuous analogue of $\sum_{j=1}^{k} \eta_j$. Writing the same schedule in intrinsic time as $\varphi(t) = \eta(z(t))$, the functional scaling law (FSL) (Li et al., 2025; 2026) models the final excess loss as

$$\mathcal{L}_{N,B}[\eta] = S(T) + \frac{1}{B}\int_0^T K(T-t)\varphi(t)\,\mathrm{d}t. \tag{3.1}$$

Here $S(T)$ is the remaining signal error, while $K(T-t)$ quantifies how much noise injected at intrinsic time t persists until training ends. We assume long-time power-law behavior, $S(t) \sim C_S t^{-s}$ and $K(t) \sim C_K t^{-\nu}$, where $s > 0$ is the learning exponent and $\nu > 0$ is the forgetting exponent. These exponents are determined by the learning problem and admit explicit spectral interpretations in kernel regression. FSL thus separates the benefit of accumulating intrinsic time from the residual noise induced by the LR schedule.

4 K-SCALING UNDER FUNCTIONAL SCALING LAW

Our theoretical analysis omits warmup and considers schedules with a stable phase followed by terminal decay:

$$\eta(z) = \begin{cases} \eta_{\text{peak}}, & 0 \le z \le (1-r)N, \\ \eta_{\text{peak}}\, \xi\Big(\frac{z-(1-r)N}{rN}\Big), & (1-r)N < z \le N. \end{cases} \tag{4.1}$$

Here $\xi : [0,1] \to [0,1]$ is a fixed decreasing profile with $\xi(0) = 1$ and $\xi(1) = 0$. We optimize jointly the peak LR η_{peak} and decay ratio r. With $D = BN$, the joint design problem is

$$\min_{\substack{0<\eta_{\text{peak}}\le\eta_{\text{crit}} \\ r\in[0,1]}} \mathcal{F}_{D,B}(\eta_{\text{peak}}, r), \qquad \mathcal{F}_{D,B}(\eta_{\text{peak}}, r) := \mathcal{L}_{D/B,B}[\eta]. \tag{4.2}$$

Here η_{crit} denotes the critical LR, the peak-LR ceiling imposed by the training-stability constraint, treated as fixed for a given model and optimizer configuration.

Theoretical setting. We consider power decay $\xi(u) = (1-u)^p$ with $p > 0$ and write $\alpha = p/(1+p)$. Assume $s > \alpha$ and $\nu > 1 + \alpha$, together with the regularity conditions in Appendix B.

The optimization is simpler in intrinsic time. Let T denote the total intrinsic time and ρ the fraction allocated to decay, related to the training-step decay ratio by $r = (1+p)\rho/(1+p\rho)$. Substituting the schedule into FSL and retaining the dominant terms in the large-budget limit gives

$$\mathcal{J}_D(T,\rho) = C_S T^{-s} + \frac{M}{D} T^{1-\alpha} G(\rho), \qquad G(\rho) = (1+p\rho)\rho^{-\alpha}, \tag{4.3}$$

where $C_S, M > 0$ are problem-dependent constants. Optimizing this leading-order FSL objective subject to the critical-LR constraint $\eta_{\text{peak}} \le \eta_{\text{crit}}$ yields the following result. The coordinate conversion, asymptotic reduction, and proof are provided in Appendices B–C.

Theorem 1 (K-scaling collapse)**.** *There exists a critical BS $B_c(D) = CD^{(s-\alpha)/(s+1-\alpha)}$, with $C > 0$ independent of D and B, such that the optimal peak LR and decay ratio satisfy*

$$\eta^\star_{\text{peak}} = F_\eta(x), \qquad r^\star = F_r(x), \qquad x = B/B_c(D), \tag{4.4}$$

where

$$F_\eta(x) = \eta_{\text{crit}} \begin{cases} x, & x \le 1, \\ 1, & x > 1, \end{cases} \qquad F_r(x) = \begin{cases} 1, & x \le 1, \\ R(x), & x > 1. \end{cases}$$

For each $x > 1$, $R(x)$ is the unique solution $r \in (0,1)$ of

$$r^{1+\alpha}\left(\frac{1-\alpha}{1-\alpha r}\right)^{s+1} = x^{-(s+1-\alpha)}. \tag{4.5}$$

The profile F_r is continuous, with a kink at $x = 1$, and strictly decreases toward zero for $x > 1$.

A shared transition from LR growth to decay shortening. The shared CBS arises because increasing peak LR and shortening decay offer two different ways to accumulate intrinsic time. At a fixed total intrinsic time, pure decay minimizes noise effect. As BS increases and fewer updates remain, the optimal schedule therefore retains pure decay and raises peak LR to preserve the preferred total intrinsic time. This response remains feasible until peak LR reaches η_{crit}. Beyond this point, retaining pure decay would reduce total intrinsic time and leave more signal unlearned. Shortening decay raises the average LR at the same peak, recovering some of that learning progress at the cost of greater residual noise. The FSL optimum begins making this tradeoff precisely when peak LR reaches its ceiling, so LR saturation and the departure from pure decay share the same CBS.

Scaling collapse at a fixed peak LR. In practice, peak LR may already be prescribed, leaving only the decay ratio to tune. The optimal decay ratio still follows the profile F_r from Theorem 1, with the CBS rescaled according to the prescribed peak LR.

Corollary 4.1. *Under the same theoretical setting, fix $\eta_{\text{peak}} = \bar\eta \in (0, \eta_{\text{crit}}]$ independently of D and B. The optimal decay ratio satisfies*

$$r^\star_{\bar\eta}(D,B) = F_r\left(\frac{B}{B_c(D;\bar\eta)}\right), \qquad B_c(D;\bar\eta) = \frac{\bar\eta}{\eta_{\text{crit}}} B_c(D), \tag{4.6}$$

where F_r is the profile in Theorem 1.

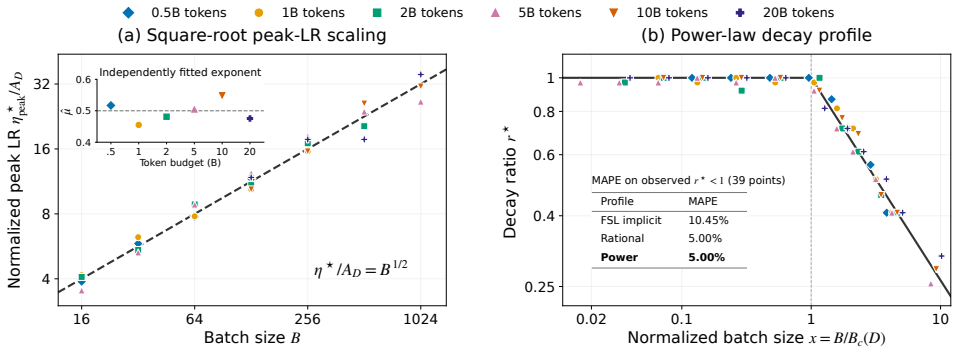


Figure 3: Empirical calibration of the K-scaling profiles on Llama-120M. (a) Rising-branch optima collapse after rescaling peak LR by the budget-specific coefficient A_D fitted in $\eta^\star_{\text{peak}} = A_D B^\mu$. The dashed line shows square-root scaling with $\mu = 1/2$; the inset shows independently fitted exponents. **(b)** Calibration of the decay-ratio profile as a function of BS normalized by the fitted CBS. We compare the FSL-implicit profile, a rational profile, and a clipped power law. Each candidate independently fits its CBS and shape parameters to all observed optima. The table reports MAPE on the decay branch.

5 Calibrating K-Scaling in LLM Pretraining

Figure 1(a,b) shows striking agreement with the qualitative structure predicted by FSL: optimal peak LR increases toward a plateau, while the optimal decay ratio stays near pure decay at small BS and decreases beyond a transition. The two transitions occur at similar batch sizes and shift together toward larger BS as the token budget increases, suggesting a shared, budget-dependent CBS.

Motivated by Theorem 1, we consider the following empirical K-scaling family: $B_c(D) = CD^b$ denotes the shared CBS and

$$\widehat{\eta}_{\text{peak}}(D,B) = \eta_{\text{crit}} \min\{x^\mu, 1\}, \quad \widehat{r}(D,B) = \min\{1, R(x)\}, \quad x = \frac{B}{B_c(D)}. \tag{5.1}$$

Here, FSL motivates the shared scaling variable $x = B/B_c(D)$ and the increase-to-plateau versus plateau-to-decay structure, while the detailed profiles will be calibrated empirically. This distinction is important because the precise scaling exponents predicted by FSL need not transfer quantitatively to practical LLM training.

We use the jointly searched optima of Llama-120M to calibrate the two profiles.

- **Peak LR: square-root scaling.** For each token budget, we fit $\eta^\star_{\text{peak}} = A_D B^\mu$ in log space over the rising branch, including the first sampled plateau point. The fitted exponents range from 0.455 to 0.549. Fixing $\mu = 1/2$ and refitting only A_D collapses the rescaled optima onto $B^{1/2}$ (Figure 3(a)), supporting $\mu = 0.5$. While Theorem 1 predicts a linear rising branch under the FSL assumptions, the experiments preserve the increase-to-plateau structure but favor square-root scaling, consistent with recent analyses of sign-based optimizers and batch-size experiments with Muon (Shulgin et al., 2026; Cesista & Wen, 2025).
- **Decay ratio: power law.** For $R(x)$, we compare the FSL-implicit profile in Eq. (4.5), the rational form $(a+1)/(a+x^\theta)$, and the power law $x^{-\theta}$. Each profile independently fits $B_c(D) = CD^b$ and its shape parameters using decay-ratio data alone by minimizing the mean absolute percentage error (MAPE):

$$\text{MAPE}(\mathcal{Q}) = \frac{100}{|\mathcal{Q}|} \sum_{j \in \mathcal{Q}} \frac{|\widehat{r}_j - r^\star_j|}{r^\star_j}.$$

  Fitting all observed optima across six token budgets and evaluating on the same 39 decay-branch points ($r^\star_j < 1$), the three profiles achieve MAPEs of 10.45%, 5.00%, and 5.00%, respectively; see Figure 3(b). Since the rational form gives no improvement over the power law, we adopt the simpler choice $R(x) = x^{-\theta}$.

Calibrated K-scaling model. The above profile calibration reduces Eq. (5.1) to the four-parameter model (1.3), with $\mu = 0.5$ and $R(x) = x^{-\theta}$. We jointly fit Eq. (1.3) across all token budgets

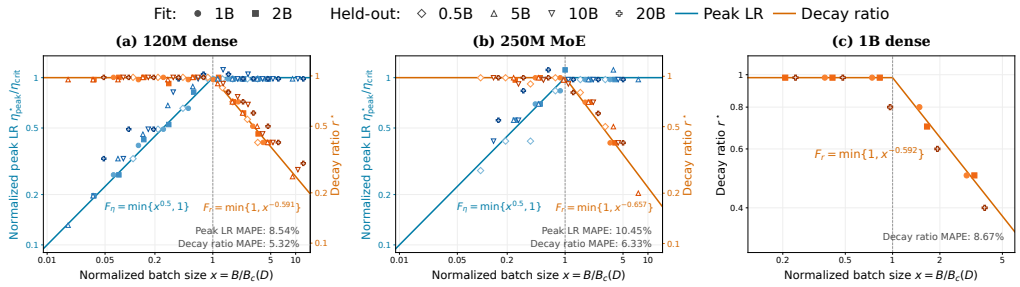


Figure 4: Schedule transfer across token budgets. Curves are calibrated on 1B- and 2B-token optima (filled markers); open markers show held-out optima, with up to $10\times$ budget extrapolation. **(a,b)** Joint peak-LR and decay-ratio transfer. **(c)** Decay-ratio transfer at fixed peak LR 10^{-2}. All normalization uses calibration data only; dashed lines mark $B = B_c(D)$.

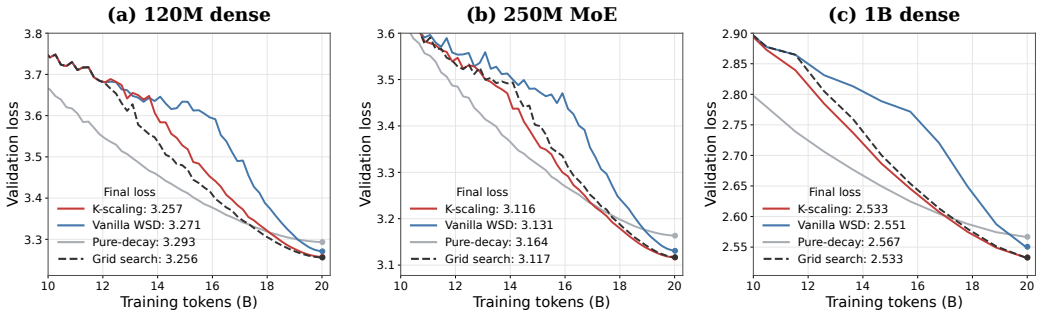


Figure 5: Large-batch training at 20B tokens. K-scaling uses 1B/2B-token calibration. In (a,b), pure linear decay and WSD with 20% decay have independently tuned peak LRs; dashed curves show the best joint peak-LR and decay-ratio grid-search results. In (c), all schedules use a fixed peak LR of 10^{-2}, and the dashed curve shows the best decay-ratio grid-search.

by minimizing the MAPE of the predicted optimal peak LR and decay ratio. Figure 1(c) shows a clear scaling collapse after normalizing BS by the resulting shared CBS, which aligns both transitions across budgets. This all-budget fit validates the joint K-scaling and establishes its descriptive accuracy. Unless stated otherwise, all subsequent experiments use this calibrated K-scaling model.

6 Transferring Learning Rate Schedules Across Token Budgets

We fit K-scaling on the 1B- and 2B-token runs and test on larger budgets up to 20B tokens without refitting, corresponding to a $10\times$ **extrapolation**. We predict both peak LR and decay ratio for Llama-120M and QwenMoE-250M, and only the decay ratio for Llama-1B with peak LR fixed at 10^{-2} (Corollary 4.1).

Figure 4(a,b) compares the K-scaling predictions with the observed optima for Llama-120M and QwenMoE-250M. Across the held-out token budgets, the model captures the transition of peak LR from the increasing branch to the plateau, as well as the decrease of the optimal decay ratio once the normalized BS $x > 1$. The extrapolated predictions track the observed optima well overall, with the main discrepancy being an underestimation of peak LR for some subcritical configurations. These results indicate that the CBS rescaling learned from small budgets transfers the joint dependence of peak LR and decay ratio to substantially larger token budgets.

We further test decay-ratio transfer on Llama-1B, where peak LR is fixed at 10^{-2}. Figure 4(c) shows that the small-budget fit captures the transition from pure decay to shorter decay at the held-out 20B-token budget. This agrees with the fixed-peak-LR scaling structure in Corollary 4.1 and supports transferring the decay ratio when peak LR is prescribed rather than jointly tuned.

Performance and data efficiency in large-batch training. We evaluate schedule transfer at 20B tokens, $10\times$ the largest calibration budget, using batch sizes of 16,384 for Llama-120M and QwenMoE-250M and 2,048 for Llama-1B. K-scaling predicts both peak LR and decay ratio for the first two models, and only the decay ratio for Llama-1B. We compare against pure linear decay

and vanilla WSD (20% decay), with peak LR independently tuned for each baseline in the first two models and fixed at 10^{-2} for Llama-1B. We use the best joint-search configurations as references for the first two models and the best decay-ratio search configuration for Llama-1B. As shown in Figure 5, K-scaling reduces final validation loss by 0.014, 0.015, and 0.018 relative to WSD, and by 0.036, 0.047, and 0.034 relative to pure decay, across the three models. The predicted schedules perform comparably to the corresponding grid-search references while consistently outperforming both fixed-ratio baselines.

Improving data efficiency when scaling batch size. For Llama-120M, we fix the number of training updates and increase batch size from 4,096 to 8,192 and 16,384, yielding token budgets of 5B, 10B, and 20B, respectively. K-scaling adapts both peak LR and decay ratio using the same small-budget calibration, while each baseline's peak LR is tuned independently at each configuration. Figure 6 shows that K-scaling achieves lower final losses along this scaling path. The fitted loss–budget curves estimate that it requires 47% fewer tokens to match the fitted 20B-token loss of WSD with 20% decay, and 73% fewer tokens relative to pure linear decay. These results show that adapting both schedule parameters improves data efficiency as batch size increases.

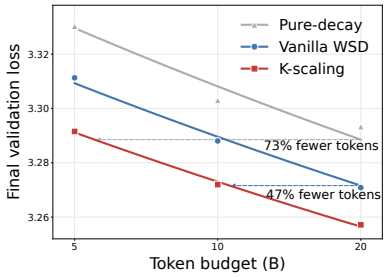


Figure 6: Data efficiency when scaling batch size.

7 Revisiting Learning Rate Schedule Design

We now use K-scaling to provide a unified view of prior empirical observations on LR scheduling through their dependence on batch size and token budget.

WSD favors a larger decay ratio at higher TPP with fixed BS. Previous studies report that longer training at fixed batch size favors a larger decay fraction (Wen et al., 2025; Luo et al., 2026). Related evidence shows that pure linear decay outperforms WSD with a fixed cooldown fraction in a high-TPP setting (Bergsma et al., 2025b). Figure 7(a,b) reproduces the former trend for Llama-120M at a fixed BS of 1,024, with peak LR tuned independently for each decay ratio at each token budget. As the token budget increases from 0.5B to 20B, TPP rises from approximately 4.2 to 167, and the optimal decay ratio increases from 0.2 to 1.

K-scaling gives a direct explanation. Since $B_c(D) = CD^b$ with $0 < b < 1$, increasing D at fixed B decreases the normalized BS $x = B/B_c(D)$, moving the optimum toward the pure-decay plateau. Panel (c) shows the same observations in this coordinate: larger token budgets shift the operating point leftward along the decay-ratio profile. This interpretation also offers a possible explanation for the advantage of pure linear decay over fixed-fraction WSD at high TPP. However, increasing TPP through larger batches at fixed training updates increases $x \propto D^{1-b}$, favoring shorter decay. In Figure 6, K-scaling selects decay ratios of 0.40, 0.36, and 0.33 at 5B, 10B, and 20B tokens, respectively. Thus, the preferred decay ratio depends on normalized BS, rather than TPP alone.

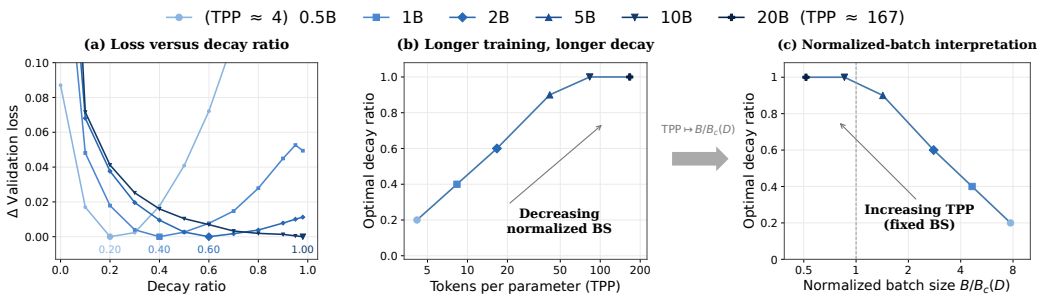


Figure 7: Higher TPP favors longer decay at fixed BS. Results for Llama-120M with $B = 1{,}024$. **(a)** Validation loss relative to each budget's minimum. **(b,c)** Optimal decay ratio versus token budget (and hence TPP) and normalized BS $x = B/B_c(D)$, respectively. $B_c(D)$ uses the all-budget fit; lines guide the eye.

The preference between cosine and WSD reverses with batch size. Prior studies report different preferences for Muon: Semenov et al. (2025) favor WSD, whereas Kimi K3 (Team et al., 2026)

adopts cosine decay after independent tuning. For Llama-120M at 2B tokens, pure cosine performs better at small BS, while WSD with 20% decay becomes better at large BS after independently tuning peak LR (Figure 8(a)). K-scaling provides an interpretation: increasing BS favors a smaller decay ratio, shifting the preferred schedule from prolonged decay toward a longer stable phase. Consistently, linear, cosine, and $1-\sqrt{\cdot}$ (Hägele et al., 2024) shapes all favor shorter decay at larger BS when peak LR and decay ratio are jointly tuned (panel (b)). This trend persists when decay shape is also optimized (panel (c)), supporting a batch-dependent preference for decay duration beyond any single decay shape.

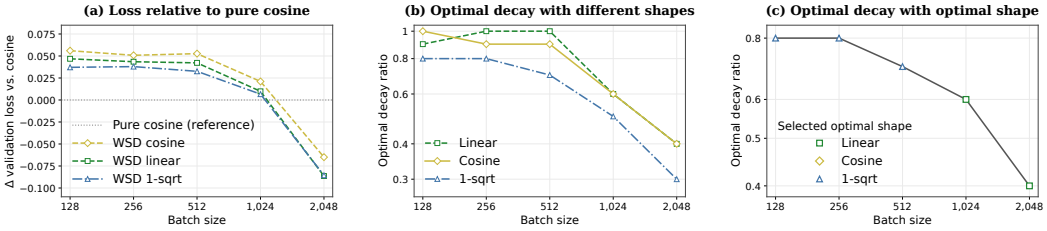


Figure 8: Batch-dependent schedule preferences. Results for Llama-120M at 2B tokens. **(a)** Validation-loss differences relative to pure cosine for WSD with 20% decay, with peak LR tuned independently. **(b)** Optimal decay ratios for each decay shape, jointly tuned with peak LR. **(c)** Optimal decay ratios jointly tuned with decay shape and peak LR.

8 Conclusion and Discussion

In this paper, we study how batch size shapes the optimal LR schedule through two practical tuning variables: peak LR and decay ratio. Building on the functional scaling law (FSL) framework, originally derived for kernel regression, we show that both quantities are organized by a shared CBS. This leads to a simple K-scaling rule, which we calibrate and validate in LLM pretraining across token budgets, model architectures, and model scales. The resulting batch-dependent picture also unifies several observations from prior work, including the changing preference between pure decay and WSD and the dependence of the optimal decay ratio on tokens per parameter.

Relaxing the shared-CBS constraint. A natural next question is whether the peak-LR and decay-ratio transitions must share exactly the same CBS, as predicted by the current FSL theory. As a preliminary exploration, we allow separate scales $B_{c,\eta}(D) = C_\eta D^{b_\eta}$ and $B_{c,r}(D) = C_r D^{b_r}$ while keeping the same scaling profiles. This modest extension improves the all-budget fit for both parameters and held-out peak-LR prediction, but slightly worsens held-out decay-ratio prediction (Figure 9). The result suggests that the shared-CBS structure captures the leading trend, while the two transitions may have distinct higher-order dependence not represented in the current FSL theory. One possibility is that this separation is controlled by tokens per parameter, rather than token budget alone. Testing this hypothesis across model scales is a natural next step toward a more complete scaling description.

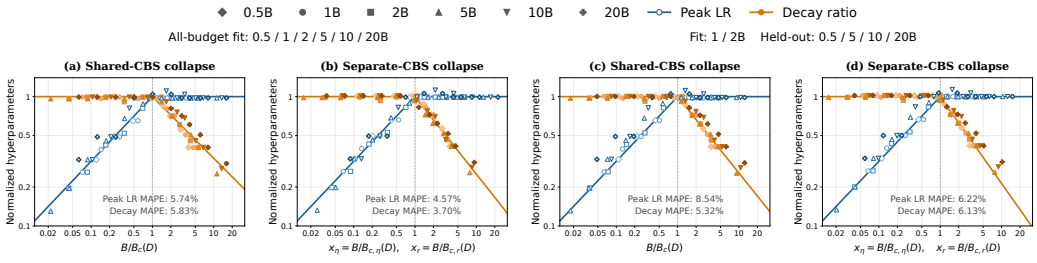


Figure 9: Shared versus separate critical batch scales. Results for Llama-120M. **(a,b)** All-budget fits. **(c,d)** Calibration on 1B/2B tokens, with MAPE evaluated on held-out budgets. Separate models normalize peak LR and decay ratio by their respective CBSs.

Our current theory captures the leading batch-size structure, but not all quantitative details observed in LLM training. In particular, the square-root peak-LR branch is calibrated empirically for Muon, and the two transitions need not coincide exactly. Developing an FSL theory that explains these effects is a natural next step. Beyond this, K-scaling is currently calibrated separately at each model size. A more complete scaling law should characterize how its parameters vary with model size, enabling transfer across both training budgets and model scales.

AI USE STATEMENT

In this work, we used generative AI tools to assist with modifying scientific figures and tables, improving manuscript readability, checking grammar, editing software code, and searching for information and relevant literature. We reviewed all AI-assisted work, including verifying that the revised figures and tables accurately represented the underlying data, ensuring that language edits preserved the intended scientific meaning, reviewing AI-assisted code for correctness, and checking the cited literature and associated claims against the original sources. We take full responsibility for the final content of this work, including all text, claims, code, and figures produced with the assistance of generative AI.

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. In *Advances in Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=bFXbLQzRoZ. (cited on page 4)

Shane Bergsma, Nolan Dey, Gurpreet Gosal, Gavia Gray, Daria Soboleva, and Joel Hestness. Straight to zero: Why linearly decaying the learning rate to zero works best for llms. In *International Conference on Learning Representations*, volume 2025, pp. 22102–22133, 2025b. (cited on pages 1, 8, and 14)

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024. (cited on page 14)

Johan Bjorck, Alon Benhaim, Vishrav Chaudhary, Furu Wei, and Xia Song. Scaling optimal lr across token horizons. In *International Conference on Learning Representations*, volume 2025, pp. 83640–83657, 2025. (cited on page 14)

Franz Louis Cesista and Kaiyue Wen. Critical batch size for steepest descent under arbitrary norms. Online technical article, November 2025. URL https://leloykun.github.io/ponder/steepest-descent-crit-bz/. (cited on page 6)

Aaron Defazio, Ashok Cutkosky, Harsh Mehta, and Konstantin Mishchenko. Optimal linear decay learning rate schedules and further refinements. *arXiv preprint arXiv:2310.07831*, 2023. (cited on pages 1 and 14)

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. (cited on page 14)

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. *Advances in Neural Information Processing Systems*, 37:76232–76264, 2024. (cited on pages 9 and 14)

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024. (cited on pages 2 and 14)

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/. (cited on pages 4, 14, and 20)

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. (cited on page 20)

Binghui Li, Fengling Chen, Zixun Huang, Lean Wang, and Lei Wu. Functional scaling laws in kernel regression: Loss dynamics and learning rate schedules. *Advances in Neural Information Processing Systems*, 38:101211–101269, 2025. (cited on pages 2, 3, and 4)

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. (cited on pages 2, 3, and 4)

Shuaipeng Li, Penghao Zhao, Hailin Zhang, Xingwu Sun, Hao Wu, Dian Jiao, Weiyan Wang, Chengjun Liu, Zheng Fang, Jinbao Xue, et al. Surge phenomenon in optimal learning rate and batch size scaling. *Advances in Neural Information Processing Systems*, 37:132722–132746, 2024. (cited on pages 3 and 14)

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for llm training. *arXiv preprint arXiv:2502.16982*, 2025. (cited on pages 4, 14, and 20)

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. URL https://arxiv.org/abs/2608.24814. (cited on pages 4 and 14)

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. (cited on page 1)

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. (cited on page 20)

Kairong Luo, Jiarui Cui, Yaorui Yin, Shengqi Chen, Yiming Yang, Linxiang Gao, Yanmohan Wang, Chengxia Li, Mingzhe Zhang, Kaifeng Lyu, and Wenguang Chen. Puro-2B: Poor lab's Qwen2-1.5B trained on RTX 5090 within $5090. *arXiv preprint arXiv:2608.27370*, 2026. URL https://arxiv.org/abs/2608.27370. (cited on page 8)

Sadhika Malladi, Kaifeng Lyu, Abhishek Panigrahi, and Sanjeev Arora. On the sdes and scaling rules for adaptive gradient algorithms. *Advances in Neural Information Processing Systems*, 35: 7697–7711, 2022. (cited on page 14)

Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018. (cited on pages 3 and 14)

Will Merrill, Shane Arora, Dirk Groeneveld, and Hanna Hajishirzi. Critical batch size revisited: A simple empirical approach to large-batch language model training. *Advances in Neural Information Processing Systems*, 38:116936–116959, 2025. (cited on pages 4 and 14)

Hiroki Naganuma, Atish Agarwala, Priya Kasimbeg, and George E Dahl. What do near-optimal learning rate schedules look like? *arXiv preprint arXiv:2603.10301*, 2026. (cited on page 14)

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2406.17557. (cited on pages 4 and 19)

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. (cited on page 19)

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 *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 53267–53294. PMLR, 2025. URL https://proceedings.mlr.press/v267/schaipp25a.html. (cited on page 14)

Andrei Semenov, Matteo Pagliardini, and Martin Jaggi. Benchmarking optimizers for large language model pretraining. *arXiv preprint arXiv:2509.01440*, 2025. (cited on pages 2, 8, and 14)

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020. (cited on page 19)

Yikang Shen, Matthew Stallone, Mayank Mishra, Gaoyuan Zhang, Shawn Tan, Aditya Prasad, Adriana Meza Soria, David D Cox, and Rameswar Panda. Power scheduler: A batch size and token number agnostic learning rate scheduler. *arXiv preprint arXiv:2408.13359*, 2024. (cited on page 14)

Egor Shulgin, Dimitri von Rütte, Tianyue H. Zhang, Niccolò Ajroldi, Bernhard Schölkopf, and Antonio Orvieto. Deriving hyperparameter scaling laws via modern optimization theory. *arXiv preprint arXiv:2603.15958*, 2026. doi: 10.48550/arXiv.2603.15958. URL https://arxiv.org/abs/2603.15958. (cited on page 6)

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, M. C., Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026. (cited on pages 2, 8, and 14)

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. (cited on pages 1, 4, and 19)

Jinbo Wang, Binghui Li, Zhanpeng Zhou, Mingze Wang, Yuxuan Sun, Jiaqi Zhang, Xunliang Cai, and Lei Wu. Fast catch-up, late switching: Optimal batch size scheduling via functional scaling laws. In *International Conference on Learning Representations*, 2026a. URL https://arxiv.org/abs/2602.14208. (cited on page 3)

Siqi Wang, Zhengyu Chen, Teng Xiao, Zheqi Lv, Jinluan Yang, Xunliang Cai, Jingang Wang, and Xiaomeng Li. Scaling and transferability of annealing strategies in large language model training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33639–33647, 2026b. (cited on page 14)

Kaiyue Wen, Zhiyuan Li, Jason Wang, David Hall, Percy Liang, and Tengyu Ma. Understanding warmup-stable-decay learning rates: A river valley loss landscape perspective. *arXiv preprint arXiv:2410.05192*, 2024. (cited on page 14)

Kaiyue Wen, David Hall, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them. *arXiv preprint arXiv:2509.02046*, 2025. URL https://arxiv.org/abs/2509.02046. (cited on page 8)

Kaiyue Wen, Xingyu Dang, Kaifeng Lyu, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them II: Hyperball optimization. *arXiv preprint arXiv:2606.16899*, 2026. URL https://arxiv.org/abs/2606.16899. (cited on pages 4, 14, and 20)

Samuel Williams, Andrew Waterman, and David Patterson. Roofline: An insightful visual performance model for multicore architectures. *Communications of the ACM*, 52(4):65–76, 2009. doi: 10.1145/1498765.1498785. (cited on page 3)

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. URL https://arxiv.org/abs/2407.10671. (cited on pages 4 and 19)

Greg Yang, Edward Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tuning large neural networks via zero-shot hyperparameter transfer. *Advances in Neural Information Processing Systems*, 34:17084–17097, 2021. (cited on page 14)

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019. (cited on page 19)

Hanlin Zhang, Depen Morwani, Nikhil Vyas, Jingfeng Wu, Difan Zou, Udaya Ghai, Dean Foster, and Sham Kakade. How does critical batch size scale in pre-training? In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.21676. (cited on page 4)

Shuchen Zhu, Yuxin Fang, Mingze Wang, and Kun Yuan. Curvature-conditioned multiscale momentum with sphere constraints for LLM pretraining. *arXiv preprint arXiv:2608.28442*, 2026a. URL https://arxiv.org/abs/2608.28442. (cited on page 3)

Zhitao Zhu, Xili Wang, Shizhe Wu, Jiawei Fu, and Xiaoqing Liu. How should LLMs consume high-quality data? optimal data scheduling via quality-aware functional scaling laws. *arXiv preprint arXiv:2605.25698*, 2026b. URL https://arxiv.org/abs/2605.25698. (cited on page 3)

A Additional Related Work

Schedule selection and decay duration. Pure linear decay has theoretical motivation and empirical support in language-model training (Defazio et al., 2023; Bergsma et al., 2025b). WSD and constant-LR schedules with terminal cooldown support flexible training horizons and checkpoint reuse (Hu et al., 2024; Hägele et al., 2024). Analyses based on river-valley loss landscapes (Wen et al., 2024) and convex optimization bounds (Schaipp et al., 2025) explain aspects of their behavior and inform LR transfer across schedules. Empirical comparisons report different preferences, including for Muon (Semenov et al., 2025; Team et al., 2026); our controlled comparisons show that changing batch size can reverse schedule rankings. Schedule-shape searches also highlight the need to separate shape from LR magnitude and account for other optimization hyperparameters (Naganuma et al., 2026). Closely related work studies annealing-ratio scaling and transfer across dense and MoE configurations (Wang et al., 2026b). Our focus is the joint dependence of optimal peak LR and decay duration on batch size and data budget, and the shared critical scale linking them.

LR scaling and critical batch size. Classical batch-size rules include linear LR scaling for SGD (Goyal et al., 2017) and square-root scaling for adaptive optimizers derived through SDE approximations (Malladi et al., 2022). Beyond these regimes, Li et al. (2024) identify a nonmonotonic dependence of optimal LR on batch size. Critical batch size has also been studied through diminishing returns from data parallelism (McCandlish et al., 2018; Merrill et al., 2025). We examine a critical scale defined by a joint schedule transition: peak LR reaches a plateau as the optimal decay ratio leaves pure decay. Its equivalence to critical batch sizes defined through training efficiency is not assumed.

Hyperparameter transfer across scales. Prior work studies hyperparameter transfer across model widths (Yang et al., 2021), optimal LR across token horizons (Bjorck et al., 2025), and schedule design across batch sizes and training budgets (Shen et al., 2024); LLM technical reports also use budget-dependent LR choices (Bi et al., 2024). Our transfer problem holds the model and optimizer configuration fixed and varies token budget and batch size. Larger-model and MoE experiments test the applicability of the scaling structure with separate calibration, rather than transferring fitted parameters across model sizes.

Normalized updates and effective learning rate. Muon uses matrix-based update normalization and has been developed for scalable LLM training (Jordan et al., 2024; Liu et al., 2025). Hyperball controls weight and update norms (Wen et al., 2026), while recent work relates effective LR to nominal LR, parameter norms, and pretraining loss dynamics (Liu et al., 2026). We use Muon with Hyperball to reduce the influence of parameter-norm dynamics on schedule comparisons, and test whether the shared-scale structure derived within FSL extends to this optimizer.

B Coordinate conversion and the intrinsic-time FSL reduction

Throughout, $D = BN$ is the total data budget. We write $h = \eta_{\mathrm{crit}}$. Assume that S is continuous, positive, and nonincreasing, with $S(T) \sim C_S T^{-s}$; that K is continuous and nonnegative, with $K(0) > 0$ and $K(v) \sim C_K v^{-\nu}$; and that

$$p > 0, \qquad \alpha = \frac{p}{1+p} < \nu - 1, \qquad s > \alpha, \qquad k = s + 1 - \alpha > 1. \tag{B.1}$$

Here $C_S, C_K > 0$, and the effective peak-LR cap $h > 0$ is independent of D, B, and the schedule. The minibatch factor $1/B$ is an assumption of the FSL model.

From intrinsic time back to training steps. For an intrinsic-time profile ψ satisfying $\int_0^1 \psi(u)^{-1}\, du = 1$, define

$$z(t) = \frac{N}{T} \int_0^t \frac{\mathrm{d}v}{\psi(v/T)}, \qquad \eta(z(t)) = \frac{T}{N} \psi(t/T). \tag{B.2}$$

Then $z(T) = N$ and $dt/dz = \eta(z)$, so this reconstructs the training-step schedule without changing its LR values or their order. Profiles are bounded and positive before the endpoint, and reciprocal

integrals at a vanishing endpoint are understood as improper integrals. Conversely, a training-step fractional schedule can be written as $\eta(z) = (T/N)\zeta(z/N)$ with $\int_0^1 \zeta(u)\,du = 1$. Writing $P(u) = \int_0^u \zeta(v)\,dv$ gives

$$t(z) = TP(z/N), \qquad \psi = \zeta \circ P^{-1}, \qquad \int_0^1 \frac{du}{\psi(u)} = 1, \qquad \eta_{\text{peak}} = \frac{T}{N}\|\psi\|_\infty. \tag{B.3}$$

Thus a unit-area profile in training steps becomes a reciprocal-normalized profile in intrinsic time. The original FSL, with intrinsic-time LR $\eta(z(t))$ and minibatch noise factor $1/B$, is

$$S(T) + \frac{1}{B}\int_0^T K(T-t)\eta(z(t))\,dt.$$

Substitution and $D = BN$ give Equation (3.1).

Decay shape and decay ratio in the two coordinates. The main text uses only the intrinsic-time exponent α. For the conversion, define the training-step exponent $p = \alpha/(1-\alpha)$, equivalently $\alpha = p/(1+p)$. A stable phase followed by $(1-v)^p$ decay, with v the fraction of the decay phase completed in training steps, becomes a power profile with exponent α in remaining intrinsic time. Indeed, integrating the LR from any point in the decay phase to termination gives remaining intrinsic time proportional to $(1-v)^{p+1}$, so LR is proportional to its $p/(p+1)$ power. In particular, linear decay ($p = 1$) corresponds to $\alpha = 1/2$. For the power-decay family, let r be the fraction of training steps spent decaying and let ρ be the fraction of intrinsic time spent decaying. Direct integration in training-step time yields

$$\begin{aligned} T &= \eta_{\text{peak}} N(1-\alpha r), \quad T_{\text{decay}} = \eta_{\text{peak}} N(1-\alpha) r, \\ \rho &= \frac{r}{1+p-pr}, \qquad\qquad r = \frac{(1+p)\rho}{1+p\rho}. \end{aligned} \tag{B.4}$$

In particular, $r = 1$ and $\rho = 1$ both denote pure decay. Writing $c_\rho = 1 + p\rho = 1 + \alpha\rho/(1-\alpha)$, the normalized intrinsic-time profile is

$$\psi_\rho(u) = c_\rho \begin{cases} 1, & 0 \le u \le 1-\rho, \\ \big((1-u)/\rho\big)^\alpha, & 1-\rho < u \le 1. \end{cases} \tag{B.5}$$

The peak and cap are

$$\eta_{\text{peak}} = \frac{BTc_\rho}{D}, \qquad Tc_\rho \le \frac{hD}{B}. \tag{B.6}$$

Indeed, $\int_0^1 du/\psi_\rho(u) = c_\rho^{-1}(1-\rho+\rho/(1-\alpha)) = 1$. Thus these coordinates describe exactly the same schedules.

Exact noise decomposition. After subtracting the schedule-independent irreducible loss, the assumed FSL becomes

$$\mathcal{F}_D(T,\rho) = S(T) + \frac{Tc_\rho}{D} W(T,\rho T), \tag{B.7}$$

$$W(T,q) = \int_q^T K(v)\,dv + q^{-\alpha}\int_0^q v^\alpha K(v)\,dv. \tag{B.8}$$

The two integrals are the stable-phase and decay-phase contributions, expressed using the remaining intrinsic time v. At $\rho = 0$, use the continuous extension $W(T,0) = \int_0^T K(v)\,dv$.

The condition $\alpha < \nu - 1$ makes $M = \int_0^\infty v^\alpha K(v)\,dv$ finite and positive. As $q \to \infty$, uniformly over $T \ge q$,

$$0 \le \int_q^T K(v)\,dv = O(q^{1-\nu}) = o(q^{-\alpha}), \qquad \int_0^q v^\alpha K(v)\,dv = M + o(1).$$

Consequently, for $T \to \infty$ and $\rho T \to \infty$,

$$\mathcal{F}_D(T,\rho) = C_S T^{-s}[1+o(1)] + \frac{M}{D} T^{1-\alpha} c_\rho \rho^{-\alpha}[1+o(1)]. \tag{B.9}$$

The relative errors are uniform when the displayed intrinsic-time scales diverge uniformly. In particular, this identifies the leading objective

$$J_D(T,\rho) = C_S T^{-s} + \frac{T^{1-\alpha}}{D} G(\rho), \qquad G(\rho) = M(1+p\rho)\rho^{-\alpha}. \tag{B.10}$$

Retaining $G(\rho)$, rather than only the order of the noise term, is essential for determining the optimal schedule.

C Proof of Theorem 1

Proof. We minimize (B.10) over $T > 0$ and $0 < \rho \le 1$, subject to (B.6).

Unconstrained optimum and the critical batch. Since $p(1-\alpha) = \alpha$,

$$G'(\rho) = M\alpha(\rho - 1)\rho^{-1-\alpha} < 0 \qquad (0 < \rho < 1).$$

At fixed T, increasing ρ strictly reduces the objective until pure decay is reached. Without the cap, the unique optimal ratio is therefore $\rho = 1$. Since $G(1) = M/(1-\alpha)$,

$$\frac{\partial}{\partial T} J_D(T,1) = T^{-s-1}\left(\frac{M}{D}T^k - sC_S\right).$$

The unique unconstrained optimum is $(T_0, 1)$, where

$$T_0 = \left(\frac{sC_S D}{M}\right)^{1/k}, \qquad B_c(D) = \frac{hD}{(1+p)T_0} = C_B D^{b^\star}, \qquad b^\star = 1 - \frac{1}{k}, \tag{C.1}$$

with $C_B = \frac{h}{1+p}(\frac{M}{sC_S})^{1/k}$. This pair is feasible exactly when $x = B/B_c(D) \le 1$, and its peak LR is hx.

The constrained branch. An optimum with $\rho < 1$ and a strict cap margin is impossible: increasing ρ slightly at fixed T remains feasible and reduces G. If $x > 1$, the largest feasible pure-decay time is $T_0/x < T_0$, where the pure-decay objective is still decreasing. Thus a pure-decay point with a strict margin is also suboptimal. Every optimum for $x > 1$ lies on the cap,

$$T = \frac{hD}{Bc_\rho} = \frac{T_0(1+p)}{xc_\rho}, \qquad \eta_{\text{peak}} = h.$$

On this boundary the objective is

$$J^{\text{cap}}(\rho) = C_S\left(\frac{B}{hD}\right)^s c_\rho^s + \frac{M}{D}\left(\frac{hD}{B}\right)^{1-\alpha} c_\rho^\alpha \rho^{-\alpha}.$$

Its derivative has the sign of $\Phi(\rho) - x^{-k}$, where

$$\Phi(\rho) = \rho^{1+\alpha}\left(\frac{1+p\rho}{1+p}\right)^{s-\alpha}, \qquad \frac{d}{d\rho}\log\Phi(\rho) = \frac{1+\alpha}{\rho} + \frac{p(s-\alpha)}{1+p\rho} > 0. \tag{C.2}$$

Because $\Phi(0+) = 0$ and $\Phi(1) = 1$, there is a unique root $\rho_\star(x) \in (0,1)$ of $\Phi(\rho) = x^{-k}$ for $x > 1$. The derivative changes from negative to positive at this root, so it is the unique global minimum on the cap. Together with the unconstrained branch, this proves uniqueness and

$$\eta^\star_{\text{peak}} = h\min\{x, 1\}, \qquad \rho_\star(x) = 1 \quad (x \le 1), \qquad \Phi(\rho_\star(x)) = x^{-k} \quad (x > 1).$$

Peak-LR saturation and departure from pure decay therefore occur at the same $x = 1$ in the leading objective.

Returning to training steps. The map $r = (1+p)\rho/(1+p\rho)$ is strictly increasing and takes 0 and 1 to themselves. Substituting its inverse into (C.2) gives

$$r^{1+\alpha}\left(\frac{1-\alpha}{1-\alpha r}\right)^{s+1} = x^{-k}. \tag{C.3}$$

This is the training-step counterpart of Equation (C.2), with $k = s+1-\alpha$. It determines the continuous, strictly decreasing right branch $r_\star(x)$, with $r_\star(x) = 1$ for $x \le 1$. The right derivatives at the transition are

$$\rho'_\star(1+) = -\frac{k}{1+\alpha k}, \qquad r'_\star(1+) = -\frac{(1-\alpha)k}{1+\alpha k},$$

whereas both left derivatives vanish. Thus the decay curve is continuous with a kink, and is smooth for $x > 1$. Finally, (C.3) gives

$$r_\star(x) \sim (1-\alpha)^{-(s+1)/(1+\alpha)} x^{-\theta}, \qquad \theta = \frac{k}{1+\alpha}, \qquad x \to \infty. \tag{C.4}$$

The intrinsic-time curve satisfies

$$\rho_\star(x) \sim (1+p)^{(s-\alpha)/(1+\alpha)} x^{-\theta}.$$

The two ratios have the same tail exponent because $r_\star/\rho_\star \to 1+p$, but different crossover shapes and tail amplitudes. The monotone conversion preserves the optimum, the pure-decay boundary, and its shared critical batch size. □

D Proof of Corollary 4.1

Proof. We use the notation and assumptions of Appendix B. Fixing $\eta_{\text{peak}} = \bar\eta$ changes (B.6) into the equality constraint

$$Tc_\rho = \frac{\bar\eta D}{B}, \qquad c_\rho = 1 + p\rho.$$

Thus T is determined by ρ, and we minimize (B.10) along this boundary.

The fixed-LR objective and critical batch. Substituting $T = \bar\eta D/(Bc_\rho)$ gives

$$J^{\bar\eta}_{D,B}(\rho) = C_S\left(\frac{B}{\bar\eta D}\right)^s c_\rho^s + \frac{M}{D}\left(\frac{\bar\eta D}{B}\right)^{1-\alpha} c_\rho^\alpha \rho^{-\alpha}.$$

Using T_0 from (C.1), define

$$B_c(D;\bar\eta) = \frac{\bar\eta D}{(1+p)T_0} = \frac{\bar\eta}{h} B_c(D), \qquad x_{\bar\eta} = \frac{B}{B_c(D;\bar\eta)}. \tag{D.1}$$

In particular, the budget exponent remains $b^\star = 1 - 1/k$. The fixed-LR objective has the same form as the cap-boundary objective in the proof of Theorem 1, with h replaced by $\bar\eta$.

The optimal intrinsic-time ratio. Its derivative has the sign of $\Phi(\rho) - x_{\bar\eta}^{-k}$, where

$$\Phi(\rho) = \rho^{1+\alpha}\left(\frac{1+p\rho}{1+p}\right)^{s-\alpha}$$

is the strictly increasing function in (C.2). Recall that $\Phi(0+) = 0$ and $\Phi(1) = 1$.

If $x_{\bar\eta} \le 1$, then $\Phi(\rho) < x_{\bar\eta}^{-k}$ for every $0 < \rho < 1$. The objective is strictly decreasing on this interval, so its unique minimum is attained at $\rho = 1$. If $x_{\bar\eta} > 1$, there is a unique root $\rho \in (0,1)$ of

$$\Phi(\rho) = x_{\bar\eta}^{-k}.$$

The derivative changes from negative to positive at this root, making it the unique global minimum. Hence the optimal intrinsic-time ratio is precisely $\rho_\star(x_{\bar\eta})$ from the main theorem.

Returning to training steps. Applying the strictly increasing conversion $r = (1+p)\rho/(1+p\rho)$ yields

$$r^\star_{\bar\eta}(D,B) = F_r(x_{\bar\eta}).$$

For $x_{\bar\eta} \le 1$, this ratio equals one. For $x_{\bar\eta} > 1$, it is the unique solution of

$$r^{1+\alpha}\left(\frac{1-\alpha}{1-\alpha r}\right)^{s+1} = x_{\bar\eta}^{-k},$$

and therefore decreases strictly toward zero. The same profile and tail exponent follow because this is exactly (C.3) evaluated at $x_{\bar\eta}$. The critical LR does not enter this conditional optimization beyond requiring $\bar\eta \le h$. □

E Connection to the full FSL optimization

We justify passing from (B.7) to the leading optimum for each fixed $x > 0$. Follow $B = xC_B D^{b^\star}$ as $D \to \infty$, with C_B as in Equation (C.1). Write $\tau = T/D^{1/k}$. The feasible set becomes

$$\tau c_\rho \le \frac{h}{xC_B}, \qquad 0 \le \rho \le 1.$$

On compact subsets with $\tau, \rho > 0$, (B.9) implies locally uniform convergence

$$D^{s/k}\mathcal{F}_D(D^{1/k}\tau, \rho) \longrightarrow E_x(\tau,\rho) := C_S\tau^{-s} + M\tau^{1-\alpha}c_\rho\rho^{-\alpha}. \tag{E.1}$$

The preceding proof gives a unique minimizer of this limiting problem. To obtain convergence of minimizers, it remains to rule out escape to the boundaries omitted from local convergence.

Compactness of minimizing schedules. For each D, the exact objective extends continuously to the compact feasible set with $T \ge 0$, including $\rho = 0$; at $T = 0$ it equals $S(0)$. Hence it attains its minimum. A fixed feasible pair $\bar\tau, \bar\rho > 0$ gives a comparison value $O(D^{-s/k})$. Positivity and the signal asymptotics imply the global bound $S(T) \ge c(1+T)^{-s}$. Every sequence of exact minimizers $(T_\star, \rho_\star^D)$ consequently satisfies

$$T_\star \to \infty, \qquad 0 < \epsilon \le \tau_\star := T_\star/D^{1/k} \le \frac{h}{xC_B}$$

for all sufficiently large D.

For large T, uniformly over $0 \le q \le T$,

$$W(T,q) \ge c(1+q)^{-\alpha}. \tag{E.2}$$

For large q, this follows by retaining a fixed positive part of the moment $\int_0^q v^\alpha K(v)\,dv$. For bounded q, retain a fixed positive kernel mass above that bounded interval in the stable integral; such a mass exists by the positive kernel-tail asymptotic, and $T \to \infty$ includes it. The two bounds cover all q with constants independent of T, q. Using $q = \rho T_\star$ and the bounds on $\tau_\star$, the rescaled noise is therefore bounded below by

$$D^{s/k}\frac{T_\star c_\rho}{D}W(T_\star, \rho T_\star) \ge c'\big(\rho + D^{-1/k}\big)^{-\alpha}.$$

This would diverge if $\rho_\star^D \to 0$, contradicting the bounded rescaled comparison value. Thus the minimizing ratios are bounded away from zero.

All exact minimizers now lie in a compact region where (E.1) is uniform. Every subsequential limit minimizes E_x; uniqueness implies convergence of the whole sequence, of the minimum values, and of the corresponding peak LR and training-step ratio. Explicitly,

$$\eta^D_{\mathrm{peak},\star} \to h\min\{x,1\}, \qquad \rho_\star^D \to \rho_\star(x), \qquad r_\star^D \to r_\star(x).$$

For fixed x, the optimal decay intrinsic time is of order $D^{1/k}$, validating the expansion.

Table 2: Model configurations. For QwenMoE-250M, FFN widths refer to the routed and shared experts, respectively.

Configuration	Llama-120M	QwenMoE-250M	Llama-1B
Total parameters	125.35M	249.21M	1.038B
Active parameters	125.35M	100.57M	1.038B
Layers	12	12	22
Hidden width	640	576	1,792
FFN width	1,792	512/256	4,864
Attention heads	10	9	28
KV heads	10	9	28
Sequence length	1,024	1,024	4,096
Routed/shared experts per layer	–	16/1	–
Selected routed experts per token	–	2	–

Scope of the asymptotics. The shared transition is exact for the leading objective and describes the fixed-x limit of the full FSL. It does not assert an exactly pure-decay plateau or exactly coincident transition points at every finite budget. Equation (C.4) is a further $x \to \infty$ limit of the limiting curve. It is not a uniform statement for arbitrary growing $x(D)$: applying the long-time reduction along such a path additionally requires $\rho T \to \infty$. If that window remains short, the full kernel must be retained.

F Experimental Settings

F.1 Model Architectures

We use two Llama-style dense models (Touvron et al., 2023), denoted Llama-120M and Llama-1B, and a Qwen2-MoE-style sparse model (Yang et al., 2024), denoted QwenMoE-250M. All three are decoder-only Transformers. Table 2 summarizes their configurations. Parameter counts include the untied input embedding and output head. For QwenMoE-250M, the active parameter count includes the two selected routed experts and the shared expert in each layer, as well as both full embedding and output matrices.

All models use pre-normalization with RMSNorm (Zhang & Sennrich, 2019), SwiGLU activations (Shazeer, 2020), and rotary positional embeddings with base 10,000. We use neither dropout nor embedding–head weight tying. The RMSNorm ϵ is 10^{-5} for the dense models and 10^{-6} for QwenMoE-250M. Dense-model linear layers are bias-free; QwenMoE-250M uses biases only in the query, key, and value projections.

Each MoE layer routes tokens to the top two of 16 experts using softmax probabilities without renormalizing the selected weights, and includes one sigmoid-gated shared expert. The load-balancing auxiliary-loss coefficient is 10^{-3}. We use neither router z-loss nor capacity-based token dropping.

Initialization. All linear and embedding weights are independently initialized from $\mathcal{N}(0, 0.02^2)$, including the output head, MoE router, expert projections, and shared-expert gate. Biases are initialized to zero and RMSNorm scales to one. No additional depth-dependent scaling is applied to residual-projection weights.

F.2 Dataset and Tokenization

We use the `sample/100BT` subset of FineWeb-Edu (Penedo et al., 2024) and the GPT-2 tokenizer (Radford et al., 2019), with the model vocabulary padded to 50,304 entries. We reserve approximately 0.005% of documents for validation and use the remainder for training. Within each split, documents are shuffled, separated by end-of-text tokens, and concatenated into a token stream. All experiments use the same split, and each training run reads the training stream sequentially without reusing data.

Table 3: Optimizer hyperparameters shared across models.

Hyperparameter	Value
MuonH momentum	0.95 (Nesterov)
Newton–Schulz iterations	5
AdamW learning rate	$0.1\eta(z)$
AdamW (β_1, β_2)	$(0.9, 0.95)$
AdamW ϵ	10^{-8}
Weight decay	0
Global gradient-norm clipping	1.0
Autocast dtype	bfloat16

Table 4: Training token budgets and global batch sizes. Token budgets are expressed in billions of tokens.

Model	D (tokens)	B (sequences/update)
Llama-120M	0.5B	16, 32, 64, 128, 192, 256, 384, 512, 1,024
	1B	16, 32, 64, 128, 256, 384, 512, 768, 1,024
	2B	16, 32, 64, 128, 256, 512, 768, 1,024, 1,536, 2,048
	5B	16, 32, 64, 128, 256, 512, 1,024, 1,536, 2,048, 3,072, 4,096, 8,192
	10B	128, 256, 512, 1,024, 2,048, 3,072, 4,096, 6,144, 8,192, 16,384
	20B	128, 256, 512, 1,024, 2,048, 4,096, 6,144, 8,192, 12,288, 16,384, 32,768
QwenMoE-250M	0.5B	16, 32, 64, 128, 256, 512
	1B	128, 256, 512, 1,024
	2B	128, 256, 512, 1,024, 2,048
	5B	256, 512, 1,024, 2,048, 4,096, 8,192
	10B	256, 512, 1,024, 2,048, 4,096, 8,192
	20B	512, 1,024, 2,048, 4,096, 8,192, 16,384
Llama-1B	1B	16, 32, 64, 128, 256, 512
	2B	16, 32, 64, 128, 256, 512
	5B	512
	20B	128, 256, 512, 1,024, 2,048

F.3 Optimizer Configuration

We use Muon (Jordan et al., 2024; Liu et al., 2025) with Hyperball (MuonH) (Wen et al., 2026) for all parameter tensors with at least two dimensions, including the input embedding, output head, and MoE router and expert matrices. Vector and scalar parameters, including normalization weights and biases, are updated with AdamW (Kingma & Ba, 2014; Loshchilov & Hutter, 2019). The reported and tuned peak learning rate η_{peak} refers to the MuonH parameter group. The AdamW learning rate remains at 0.1 times the MuonH learning rate throughout training.

For each matrix W, MuonH fixes the radius $R_W = \|W_0\|_F$ at initialization. Following the momentum and Newton–Schulz transformations, the update is rescaled to norm $\eta(z)R_W$, where $\eta(z)$ is the MuonH learning rate at update z. The updated weight is then projected onto the sphere $\|W\|_F = R_W$. Thus, $\eta(z)$ specifies the update-to-weight norm ratio before projection. Table 3 lists the shared hyperparameters.

We use bfloat16 autocasting, while model parameters, parameter gradients, optimizer states, and Newton–Schulz computations remain in float32.

F.4 Training Budgets and Batch Sizes

Table 4 summarizes the training token budgets D and global batch sizes B. Batch size is measured in sequences per optimizer update, aggregated across all devices and gradient-accumulation steps. For sequence length L, each update processes LB tokens. Given a target token budget D, we run $N = \lceil D/(LB) \rceil$ optimizer updates.

Table 5: Evaluated hyperparameter values pooled across token budgets and batch sizes in the linear-decay experiments.

Model	Peak LR values	Decay ratio values
Llama-120M	1e-4, 2e-4, 3e-4, 5e-4, 1e-3, 1.1e-3, 1.2e-3, 1.3e-3, 1.4e-3, 2e-3, 2.5e-3, 3e-3, 3.25e-3, 3.5e-3, 4e-3, 4.25e-3, 4.5e-3, 5e-3, 5.5e-3, 6e-3, 6.2e-3, 6.4e-3, 6.5e-3, 6.6e-3, 6.8e-3, 7e-3, 7.5e-3, 8e-3, 8.25e-3, 8.5e-3, 8.75e-3, 9e-3, 9.5e-3, 1e-2, 1.02e-2, 1.04e-2, 1.05e-2, 1.06e-2, 1.08e-2, 1.1e-2, 1.2e-2, 1.225e-2, 1.25e-2, 1.275e-2, 1.3e-2, 1.325e-2, 1.35e-2, 1.375e-2, 1.4e-2, 1.5e-2, 1.6e-2, 1.7e-2, 1.75e-2, 1.8e-2, 1.9e-2, 2e-2, 2.5e-2, 3e-2, 5e-2, 1e-1	0, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2, 0.225, 0.25, 0.275, 0.3, 0.33, 0.36, 0.4, 0.432, 0.45, 0.467, 0.5, 0.55, 0.6, 0.625, 0.65, 0.675, 0.7, 0.75, 0.8, 0.825, 0.85, 0.875, 0.9, 0.925, 0.95, 1
QwenMoE-250M	5e-4, 1e-3, 4e-3, 5e-3, 7.5e-3, 1e-2, 1.25e-2, 1.5e-2, 1.75e-2, 2e-2, 2.5e-2, 3e-2, 3.5e-2, 5e-2	0, 0.1, 0.2, 0.3, 0.34, 0.4, 0.5, 0.54, 0.6, 0.7, 0.8, 0.85, 0.9, 0.95, 1
Llama-1B	1e-3, 2.5e-3, 5e-3, 7.5e-3, 1e-2, 1.5e-2	0.1, 0.2, 0.3, 0.4, 0.44, 0.5, 0.6, 0.7, 0.8, 0.9, 1

F.5 Hyperparameter Search

For Llama-120M and QwenMoE-250M, we jointly tune the peak learning rate η_{peak} and decay ratio r using configuration-dependent grids, with local refinement around low-loss configurations. We select the evaluated configuration with the lowest final validation loss, without interpolation. Runs sharing $(D, B, \eta_{\text{peak}})$ reuse a common warmup–stable trajectory and branch from checkpoints at their respective decay starts. Each branch is trained to the same target horizon. For Llama-120M at the 2B-token budget, cosine and $1 - \sqrt{\cdot}$ decay use the same peak-learning-rate and decay-ratio search ranges as linear decay.

For Llama-1B, we study decay-ratio calibration and transfer. At the 1B- and 2B-token budgets, we search $r \in \{0.1, 0.2, \ldots, 0.9, 1\}$ for calibration. The grid-search reference is selected by minimizing final validation loss over the evaluated decay ratios. Table 8 lists the evaluated configurations.

Table 5 summarizes the evaluated hyperparameter values across the linear-decay experiments. Tables 6, 7, and 8 provide coverage by model, token budget, and batch size. Intervals indicate the minimum and maximum evaluated values. The count n_{eval} denotes the number of distinct (η_{peak}, r) pairs evaluated at each (D, B); repeated runs of the same pair are counted once.

Table 6: Evaluated hyperparameter ranges for Llama-120M with linear terminal decay. Configurations with identical reported ranges are grouped. Evaluation counts are listed in batch-size order; a single count applies to every batch size in that row.

D (tokens)	B	Peak LR range	Decay ratio range	n_{eval}
0.5B	16, 32, 64	[1e-3, 1.5e-2]	[0, 1]	84
0.5B	128, 256, 512, 1,024	[1e-2, 2e-2]	[0, 1]	48
0.5B	192	[5e-3, 3e-2]	[0.7, 1]	18
0.5B	384	[5e-3, 3e-2]	[0.4, 0.7]	18
1B	16, 32	[2.5e-3, 1e-2]	[0, 1]	84
1B	64	[1e-3, 1.5e-2]	[0, 1]	132
1B	128	[5e-3, 2e-2]	[0, 1]	132
1B	256	[1e-2, 2.5e-2]	[0, 1]	48
1B	384	[5e-3, 3e-2]	[0.6, 0.9]	18
1B	512	[7e-3, 2.5e-2]	[0, 1]	84
1B	768	[5e-3, 3e-2]	[0.4, 0.7]	18
1B	1,024	[1.2e-2, 3e-2]	[0, 1]	84

Continued on next page

Table 6 (continued)

D (tokens)	B	Peak LR range	Decay ratio range	n_{eval}
2B	16, 32, 64, 128	[1e-4, 5e-2]	[0, 1]	144, 144, 216, 240
2B	256	[5e-3, 5e-2]	[0, 1]	84
2B	512, 1,024	[1e-2, 1e-1]	[0, 1]	84
2B	768	[5e-3, 3e-2]	[0.6, 0.9]	15
2B	1,536	[5e-3, 3e-2]	[0.4, 0.6]	15
2B	2,048	[1e-2, 2e-2]	[0, 1]	60
5B	16	[1e-3, 1e-2]	[0.5, 1]	54
5B	32, 64, 128	[1e-3, 1e-2]	[0, 1]	120, 122, 122
5B	256	[1e-3, 1.4e-2]	[0, 1]	173
5B	512	[1e-3, 1.5e-2]	[0, 1]	184
5B	1,024, 2,048	[1e-3, 2e-2]	[0, 1]	128, 60
5B	1,536	[5e-3, 3e-2]	[0.6, 0.9]	18
5B	3,072	[5e-3, 3e-2]	[0.4, 0.6]	15
5B	4,096	[1e-3, 5e-2]	[0, 1]	63
5B	8,192	[1e-2, 2e-2]	[0, 1]	39
10B	128, 256	[1e-3, 1.5e-2]	[0, 1]	11, 9
10B	512	[5e-3, 2e-2]	[0, 1]	51
10B	1,024, 2,048, 4,096, 16,384	[1e-2, 2e-2]	[0, 1]	52, 96, 102, 39
10B	3,072	[5e-3, 3e-2]	[0.6, 0.9]	15
10B	6,144	[5e-3, 3e-2]	[0.4, 0.7]	15
10B	8,192	[1e-3, 5e-2]	[0, 1]	54
20B	128, 256, 512	[1e-3, 1.5e-2]	[0, 1]	10, 11, 10
20B	1,024, 2,048	[5e-3, 2e-2]	[0, 1]	120, 112
20B	4,096, 8,192, 32,768	[1e-2, 2e-2]	[0, 1]	84, 36, 39
20B	6,144	[5e-3, 3e-2]	[0.6, 0.8]	15
20B	12,288	[5e-3, 3e-2]	[0.4, 0.6]	15
20B	16,384	[1e-3, 5e-2]	[0, 1]	42

Table 7: Evaluated hyperparameter ranges for QwenMoE-250M with linear terminal decay. Configurations with identical reported ranges are grouped. n_{eval} denotes the number of distinct evaluated parameter pairs for each (D, B). Counts are listed in batch-size order; a single count applies to every batch size in that row.

D (tokens)	B	Peak LR range	Decay ratio range	n_{eval}
0.5B	16	[4e-3, 7.5e-3]	[0.9, 1]	6
0.5B	32	[5e-3, 1e-2]	[0.9, 1]	6
0.5B	64	[7.5e-3, 1.5e-2]	[0.9, 1]	6
0.5B	128	[1e-2, 2e-2]	[0.7, 1]	12
0.5B	256	[1.25e-2, 2.5e-2]	[0.5, 1]	18
0.5B	512	[1.25e-2, 2.5e-2]	[0.2, 1]	21
1B	128, 256, 512, 1,024	[1e-3, 5e-2]	[0, 1]	108
2B	128, 256, 512, 1,024, 2,048	[5e-4, 5e-2]	[0, 1]	132
5B	256, 512, 1,024, 2,048, 4,096, 8,192	[5e-4, 5e-2]	[0, 1]	111, 130, 129, 132, 132, 132
10B	256	[5e-3, 1e-2]	[0.9, 1]	6
10B	512	[7.5e-3, 1.5e-2]	[0.9, 1]	6
10B	1,024	[1e-2, 1.75e-2]	[0.7, 1]	12
10B	2,048	[1.25e-2, 2.5e-2]	[0.6, 1]	18
10B	4,096	[1.25e-2, 2.5e-2]	[0.3, 1]	21
10B	8,192	[1.25e-2, 2.5e-2]	[0.2, 1]	18
20B	512	[5e-3, 1e-2]	[0.9, 1]	6
20B	1,024	[7.5e-3, 1.5e-2]	[0.9, 1]	6

Continued on next page

Table 7 (continued)

D (tokens)	B	Peak LR range	Decay ratio range	n_{eval}
20B	2,048	`[1e-2, 2e-2]`	$[0.7, 1]$	12
20B	4,096	`[1.25e-2, 2.5e-2]`	$[0.5, 1]$	21
20B	8,192, 16,384	`[1.25e-2, 5e-2]`	$[0.2, 1]$	28, 22

Table 8: Evaluated decay ratios for the reported Llama-1B experiments with linear terminal decay. $\mathcal{R} = \{0.1, 0.2, \ldots, 0.9, 1\}$. n_{eval} denotes the number of distinct evaluated parameter pairs for each (D, B). A single count applies to every batch size in that row.

D (tokens)	B	Peak LR values	Decay ratio values	n_{eval}
1B	16, 32, 64, 128, 256, 512	`1e-2`	$\mathcal{R}$	10
2B	16, 32, 64, 128, 256, 512	`1e-2`	$\mathcal{R}$	10
5B	512	`1e-2`	$\{0.2, 1\}$	2
20B	128, 256	`1e-2`	$\{0.2, 1\}$	2
20B	512	`1e-3`, `2.5e-3`, `1.5e-2`	$\{0.2, 1\}$	6
20B	512	`5e-3`, `7.5e-3`, `1e-2`	$\mathcal{R}$	30
20B	1,024	`1e-2`	$\mathcal{R}$	10
20B	2,048	`1e-3`, `2.5e-3`, `5e-3`	$\{0.2, 1\}$	6
20B	2,048	`7.5e-3`, `1.5e-2`	$\mathcal{R}$	20
20B	2,048	`1e-2`	$\mathcal{R} \cup \{0.44\}$	11