How Should LLMs Consume High-Quality Data? Optimal Data Scheduling via Quality-Aware Functional Scaling Laws

Anonymous ACL submission

Abstract

High-quality data is scarce in large language model (LLM) training, yet how to schedule its use jointly with training dynamics lacks theoretical guidance. We extend functional scaling laws by incorporating a data-quality dimension, and solve the joint data-quality and batch-size scheduling problem in asymptotic closed form. The solution reveals two regimes and a dual role of high-quality data. In the noise-limited regime, high-quality data acts as a signal amplifier: lowering the batch size converts cleaner data into more signal without amplifying noise. In the signal-limited regime, it acts as a noise suppressor: late placement reduces terminal noise without sacrificing signal accumulation. Existing curriculum-style pipelines primarily exploit the second role by placing cleaner data late, but miss the first role because conventional decay schedules reduce update intensity exactly when high-quality data becomes available. Guided by this, we propose Drop-Stable-Rampup for LLM midtraining: upon the quality transition, drop the batch size, hold it stable to accumulate signal, then ramp up to suppress terminal noise. On a 15B Mixture-of-Experts model midtrained on 108B tokens, Drop-Stable-Rampup improves average accuracy over Warmup-Stable-Decay (WSD) by +1.70 and over Cosine-decay by +2.98, with particularly large gains on mathematical reasoning benchmarks such as GSM8K (+4.23) and MATH (+2.80).

1 Introduction

High-quality data, such as textbooks, mathematical proofs and expert-curated code, is the scarcest resource in LLM training. Multi-stage training has become the standard paradigm: models are pretrained on massive noisy web data, then midtrained on a limited curated corpus to sharpen capabilities such as mathematical reasoning, code generation and agentic capabilities (AI@Meta, 2024; Qwen Team, 2025; Kimi Team, 2026). Given a fixed budget of high-quality tokens, a fundamental question arises: **how should LLMs consume high-quality data?** Specifically, at what training stage should it be introduced, and what batch-size or learning-rate schedule should accompany the transition?

A natural intuition from curriculum learning (Bengio et al., 2009) suggests placing high-quality data late in training, when the model has sufficient capacity to exploit it. However, Luo et al. (2026) identified a key conflict: conventional schedules simultaneously decay the learning rate in later stages, so the model encounters its best data precisely when its learning intensity (effective update magnitude per token) is weakest. This reveals that data scheduling and training dynamics cannot be optimized in isolation—they must be considered jointly. Yet existing theoretical frameworks do not characterize this coupled scheduling problem.

We fill this gap by developing a *quality-aware functional scaling law* (Section 3) that extends Li et al. (2026) to time-varying data quality. In our framework, the excess risk decomposes into a signal-learning term and a noise-accumulation integral that is jointly controlled by the data-quality schedule $p(t)$ and the batch-size schedule $b(t)$. We formulate the joint optimization as a variational problem and derive asymptotic closed-form optimal schedules (Theorem 4.3). The solution exhibits two regimes—*noise-limited* and *signal-limited*—depending on whether the learning bottleneck is noise accumulation or signal acquisition. Together, the two regimes reveal two complementary uses of high-quality data: late placement and terminal batch-size ramp-up use it as a *noise suppressor*, while pairing it with low batch size (or higher learning intensity) uses it as a *signal amplifier*. Existing curriculum-style schedules mainly exploit the former and miss the latter.

Translating these theoretical insights into practice, we propose **Drop-Stable-Rampup** (DSR)

for LLM midtraining (Section 5): upon the quality transition, drop the batch size; hold it stable at a low level to accumulate signal; then ramp up to suppress noise. Midtraining is a natural testbed because its limited corpus size makes low batch sizes feasible—a prerequisite that pretraining's hardware-utilization constraints typically prevent. On a 15B Mixture-of-Experts model with 108B tokens, DSR improves average accuracy over WSD (Hu et al., 2024) by +1.70 and over Cosine-decay (Loshchilov and Hutter, 2017) by +2.98.

Our contributions are threefold: we derive a quality-aware functional scaling law for time-varying data quality; solve the resulting joint data-quality and batch-size scheduling problem in asymptotic closed form, revealing a noise/signal dichotomy that resolves the curriculum-versus-decay conflict; and translate the theory into Drop-Stable-Rampup, which improves midtraining of a 15B MoE model across 14 benchmarks.

2 Preliminaries

Notation. For two nonnegative functions f, g of an index variable, we write $f \asymp g$ if there exist constants $C_1, C_2 > 0$, independent of the index variable, such that $C_1 f \le g \le C_2 f$ in the relevant range.

2.1 Feature-space linear regression

Let $\boldsymbol{x} \in \mathcal{X}$ be sampled from a distribution $\mathcal{D}$. Let $\boldsymbol{\phi} : \mathcal{X} \to \mathbb{R}^N$ be a feature map such that

$$\boldsymbol{\phi}(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{H}),$$

where $\boldsymbol{H} \in \mathbb{R}^{N \times N}$ is the feature covariance matrix. We consider the feature-space linear regression model

$$f^\star(\boldsymbol{x}) = \langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\theta}^\star \rangle, \quad f(\boldsymbol{x}; \boldsymbol{\theta}) = \langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\theta} \rangle.$$

Without loss of generality, we assume that $\boldsymbol{H}$ is diagonalized. Let $\lambda_1 \ge \lambda_2 \ge \cdots \ge 0$ denote its eigenvalues, ordered non-increasingly. This assumption is due to the rotation equivariance of stochastic gradient descent (SGD).

Assumption 2.1 (Power-law capacity and source conditions)**.** *There exist $\beta > 1$ and $s > 0$ such that*

$$\lambda_j \asymp j^{-\beta}, \quad |\theta_j^\star|^2 \asymp j^{-1} \lambda_j^{s-1}.$$

The capacity exponent β controls the decay rate of the eigenvalues. The source exponent s measures the alignment of the target function with the eigenstructure: smaller s corresponds to harder learning problems. These capacity and source conditions are standard in the analysis of kernel methods and have recently been adopted in scaling-law studies (Paquette et al., 2024; Lin et al., 2024; Bordelon et al., 2025; Li et al., 2026).

2.2 Two-level data-quality model

We assume that the label is generated as

$$y = f^\star(\boldsymbol{x}) + \epsilon,$$

where ϵ is a zero-mean noise term that captures label corruption. In practice, training data are drawn from heterogeneous sources of varying reliability. For instance, in large-scale language model training, curated corpora (e.g., textbooks, encyclopedias) produce labels that are closely aligned with the target distribution, whereas noisy web-crawled text introduces substantially larger deviations. We model this quality heterogeneity through the variance of the label noise: high-quality data correspond to small noise variance, and low-quality data correspond to large noise variance.

Concretely, the label noise ϵ comes from a two-component mixture:

$$\epsilon \sim \begin{cases} \mathcal{N}(0, \sigma_{\text{good}}^2), & \text{high-quality data,} \\ \mathcal{N}(0, \sigma_{\text{bad}}^2), & \text{low-quality data,} \end{cases}$$

where

$$0 < \sigma_{\text{good}}^2 < \sigma_{\text{bad}}^2.$$

Let $p(t) \in [0, 1]$ be the proportion of high-quality data used at continuous training time t. Then the effective label-noise variance at time t is

$$\sigma_{\text{eff}}^2(t) := p(t)\sigma_{\text{good}}^2 + (1 - p(t))\sigma_{\text{bad}}^2.$$

2.3 One-pass SGD

The parameter $\boldsymbol{\theta}$ is trained by minimizing the population risk

$$R(\boldsymbol{\theta}) := \frac{1}{2}\mathbb{E}_{\boldsymbol{x},y}\left[(f(\boldsymbol{x}; \boldsymbol{\theta}) - y)^2\right]$$

via SGD in the one-pass regime: each data point is seen exactly once. We fix a constant learning rate $\eta > 0$ and allow the batch size to vary across iterations. Denote the *batch-size schedule* by $(B_1, B_2, \ldots, B_K)$. We write $t = k\eta$ for the continuous training time and $b(t)$ for the continuous interpolation of B_k, i.e., $b(k\eta) = B_k$. At

iteration k, a mini-batch of B_k samples is drawn and the parameter is updated as

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \frac{\eta}{B_k} \sum_{i=1}^{B_k} \nabla_{\boldsymbol{\theta}} \frac{1}{2} \big(f(\boldsymbol{x}_{k,i}; \boldsymbol{\theta}_k) - y_{k,i} \big)^2. \tag{1}$$

This can be decomposed into a population gradient plus noise:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta \big(\nabla R(\boldsymbol{\theta}_k) + \boldsymbol{\xi}_k \big),$$

where the gradient noise $\boldsymbol{\xi}_k$ satisfies $\mathbb{E}[\boldsymbol{\xi}_k] = \mathbf{0}$ and $\mathbb{E}[\boldsymbol{\xi}_k \boldsymbol{\xi}_k^\top] = \boldsymbol{\Sigma}(\boldsymbol{\theta}_k)/B_k$. Here $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ denotes the per-sample gradient-noise covariance at $\boldsymbol{\theta}$. We measure performance by the excess risk

$$E(\boldsymbol{\theta}) := R(\boldsymbol{\theta}) - \frac{1}{2}\mathbb{E}[\epsilon^2] = \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}^\star\|_{\boldsymbol{H}}^2.$$

Beyond the batch-size schedule, our setting introduces a second control: the *data-quality schedule* $p(t) \in [0,1]\,(p_k = p(k\eta))$, governing the fraction of high-quality samples at each time in training. The joint schedule $(b(t), p(t))$ is the object we seek to optimize. The discrete update (1) can be modeled by the following Itô stochastic differential equation (SDE) (Li et al., 2019; Orvieto and Lucchi, 2019; Ankirchner and Perko, 2024):

$$\mathrm{d}\bar{\boldsymbol{\theta}}_t = -\nabla R(\bar{\boldsymbol{\theta}}_t)\,\mathrm{d}t + \sqrt{\frac{\eta}{b(t)}}\boldsymbol{\Sigma}(\bar{\boldsymbol{\theta}}_t)^{1/2}\,\mathrm{d}\boldsymbol{W}_t, \tag{2}$$

where $\boldsymbol{W}_t \in \mathbb{R}^N$ is a standard Brownian motion. The noise covariance satisfies $\boldsymbol{\Sigma}(\boldsymbol{\theta}) \asymp (E(\boldsymbol{\theta}) + \sigma_{\text{eff}}^2(t))\boldsymbol{H}$, i.e., it is proportional to $\boldsymbol{H}$ with a scalar prefactor depending only on the excess risk and the effective noise level (Lemma A.1 in Appendix A). This anisotropic noise alignment has been identified in (Mori et al., 2022; Wu et al., 2022; Wang and Wu, 2023); our contribution is the extension to the quality-mixture setting.

3 Quality-Aware Functional Scaling Law

3.1 Quality-aware functional scaling law

Theorem 3.1 (Quality-aware functional scaling law)**.** *Suppose Assumption 2.1 holds. Assume that $b(t) \geq B_{\min} \geq 1$, $p(t) \in [0,1]$ is measurable, and the learning rate $\eta > 0$ is sufficiently small. Define*

$$K(t) := (t+1)^{-(2-1/\beta)}.$$

Then, for sufficiently large t,

$$\mathbb{E}[E(\bar{\boldsymbol{\theta}}_t)] \asymp \underbrace{t^{-s}}_{\substack{\textit{signal}\\ \textit{learning}}} + \underbrace{\eta \int_0^t K(t-\tau) \frac{\sigma_{\text{eff}}^2(\tau)}{b(\tau)}\,\mathrm{d}\tau}_{\substack{\textit{noise}\\ \textit{accumulation}}}.$$

The proof is in Appendix B, and empirical verification is given in Appendix E.1.

Remark 3.2 (Relation to prior work)**.** *The above theorem extends the functional scaling law (FSL) of Li et al. (2026) to the data-quality scheduling setting. In their original formulation, the label-noise variance σ^2 is a fixed constant. Our contribution is to replace the constant σ^2 by the time-dependent effective variance $\sigma_{\text{eff}}^2(t)$, which couples the data-quality schedule $p(t)$ into the loss dynamics.*

3.2 Variational formulation

The quality-aware FSL (Theorem 3.1) suggests minimizing the end-of-training excess risk over the training horizon T, batch-size schedule $b(t)$, and data-quality schedule $p(t)$. Let D be the continuous-time total data budget, $\rho \in (0,1)$ the fraction of high-quality data, $r := \sigma_{\text{good}}^2/\sigma_{\text{bad}}^2 \in (0,1)$ the variance ratio, and $B_{\min} \geq 1$ a hardware lower bound on the batch size (e.g., the number of parallel devices). The fully joint scheduling problem is

$$\boxed{\begin{aligned} \min_{T>0,\,b,\,p} \quad & T^{-s} + \eta \int_0^T K(T-t) \frac{\sigma_{\text{eff}}^2(t)}{b(t)}\,\mathrm{d}t \\ \text{s.t.} \quad & \int_0^T b(t)\,\mathrm{d}t = D, \\ & \int_0^T p(t)b(t)\,\mathrm{d}t = \rho D, \\ & b(t) \geq B_{\min}, \quad 0 \leq p(t) \leq 1. \end{aligned}} \tag{3}$$

Define the critical exponent $s_{\text{crit}} := 1 - 1/\beta$. As we show below, the structure of the optimal schedule depends sharply on whether $s > s_{\text{crit}}$ (noise-limited regime) or $s < s_{\text{crit}}$ (signal-limited regime).

4 Optimal Data Schedule

4.1 Optimal data-quality schedule

We first optimize the data-quality schedule $p(t)$ with the training horizon T and batch-size schedule $b(t)$ fixed. Since $\sigma_{\text{eff}}^2(t) = p(t)\sigma_{\text{good}}^2 + (1 - p(t))\sigma_{\text{bad}}^2$ is affine in p and the p-independent part of the objective is constant, the problem reduces to

$$\begin{aligned} \max_{0 \leq p \leq 1} \quad & \int_0^T K(T-t) \frac{p(t)}{b(t)}\,\mathrm{d}t \\ \text{s.t.} \quad & \int_0^T p(t)b(t)\,\mathrm{d}t = \rho D. \end{aligned}$$

Theorem 4.1 (Optimal quality schedule for fixed batch-size schedule)**.** *Fix $T > 0$ and $b(t) \geq B_{\min}$. Define the quality-allocation score $S_b(t) := K(T-t)/b(t)^2$. Then an optimal data-quality schedule is bang-bang:*

$$p^\star(t) = \begin{cases} 1, & S_b(t) > \lambda, \\ 0, & S_b(t) < \lambda, \end{cases}$$

where λ is the unique threshold satisfying $\int_0^T p^\star(t) b(t)\,\mathrm{d}t = \rho D$ (fractional values arise only on the level set $\{t : S_b(t) = \lambda\}$). In words, high-quality data should be allocated to the time region where $K(T-t)/b(t)^2$ is largest.

The proof is given in Appendix C.

Corollary 4.2 (Quality placement depends on batch-size shape)**.** *The quality-allocation score $S_b(t) = K(T-t)/b(t)^2$ determines whether quality placement is* determinate *or* degenerate*:*

*(i) If $b(t) = D/T$ is constant, then $S_b(t) \propto K(T-t)$, which is largest near $t = T$. The optimal quality schedule is **late**: high-quality data should be saved for the end of training.*

*(ii) If $b(t) = C\sqrt{K(T-t)}$ with $C = D/\int_0^T \sqrt{K(T-t)}\,\mathrm{d}t$ (the optimal batch-size schedule of Wang et al. (2026) in the noise-limited regime), then $S_b(t) = 1/C^2$ is constant. The quality placement is **degenerate**: any allocation satisfying the budget constraint is optimal.*

Corollary 4.2(i) implies that under a constant batch size, high-quality data should be placed late—yet conventional training simultaneously decays the learning rate in this phase, reducing the model's ability to exploit the high-quality data. This is consistent with the empirical finding of Luo et al. (2026) that learning rate decay conflicts with curriculum-based data scheduling.

Empirical verification of Corollary 4.2 (constant batch and $b(t) = C\sqrt{K(T-t)}$) is provided in Appendix E.2.

4.2 Main theory

We now solve the fully joint problem (3), optimizing T, $b(t)$, and $p(t)$ simultaneously. Write $B := B_{\min}$, $\gamma := 2 - 1/\beta$, and define

$$\kappa := \frac{\sigma_{\text{good}}^2 \sigma_{\text{bad}}^2}{(1-\rho)\sigma_{\text{good}}^2 + \rho\sigma_{\text{bad}}^2}.$$

Let $I_T := \int_0^T \sqrt{K(u)}\,\mathrm{d}u$.

Theorem 4.3 (Asymptotically optimal joint schedule)**.** *Suppose Assumption 2.1 holds. The following schedules are asymptotically optimal as $D \to \infty$.*

***(I) Noise-limited regime** ($s > 1 - 1/\beta$)**.** The constraint $b(t) \geq B$ is asymptotically inactive. The optimal quality schedule is bang-bang:*

$$p^\star(t) = \mathbf{1}_S(t)$$

for a measurable set $S \subset [0, T^\star]$ satisfying

$$\int_S \sqrt{K(T^\star - t)}\,\mathrm{d}t = \frac{\rho\kappa}{\sigma_{\text{good}}^2} I_{T^\star}.$$

Any such S is equally optimal (quality placement is degenerate). The optimal batch-size schedule is

$$b^\star(t) = \begin{cases} C\sqrt{K(T^\star - t)}, & t \notin S, \\ rC\sqrt{K(T^\star - t)}, & t \in S, \end{cases}$$

where $C = D\sigma_{\text{bad}}^2/(\kappa\, I_{T^\star})$. The optimal training horizon satisfies

$$T^\star = \left(\frac{sD}{4\beta\eta\kappa}\right)^{\frac{1}{s+1/\beta}} (1 + o(1)),$$

and the optimal excess risk satisfies

$$\mathbb{E}[E(\bar{\boldsymbol{\theta}}_{T^\star})] \asymp D^{-\frac{s\beta}{1+s\beta}}.$$

***(II) Signal-limited regime** ($s < 1 - 1/\beta$)**.** The constraint $b(t) \geq B$ is active. The optimal schedule has the strict-switching form:*

1. ***Low-quality, flat**: $p^\star = 0$, $b^\star = B$ for $t \in [0, T_1^\star)$.*
2. ***High-quality, flat**: $p^\star = 1$, $b^\star = B$ for $t \in [T_1^\star, T_1^\star + T_3^\star)$.*
3. ***High-quality, ramp-up** (duration $T_4^\star := T^\star - T_1^\star - T_3^\star$): $p^\star = 1$, $b^\star(t) = B\left(\frac{T^\star - t + 1}{T_4^\star + 1}\right)^{1/(2\beta)-1}$ for $t \in [T_1^\star + T_3^\star, T^\star]$.*

The phase durations are determined as follows. Phase 1 exhausts all low-quality data:

$$T_1^\star = (1-\rho)\frac{D}{B}.$$

Phase 3 (the ramp-up) has duration

$$T_4^\star = \left(\frac{\eta\sigma_{\text{good}}^2}{sB^{s+2}}\right)^{1/\gamma} D^{\frac{s+1}{\gamma}} (1 + o(1)).$$

Phase 2 occupies the remainder: $T_3^\star = T^\star - T_1^\star - T_4^\star$, where the total horizon satisfies

$$T^\star = \frac{D}{B} - (2\beta - 1)T_4^\star + o(T_4^\star).$$

The optimal excess risk satisfies

$$\mathbb{E}[E(\bar{\boldsymbol{\theta}}_{T^\star})] \asymp B^s D^{-s}.$$

The proof is in Appendix D; the critical case $s = 1 - 1/\beta$, which lies outside the regime dichotomy above, is discussed separately in Remark D.2. We verify Theorem 4.3 by comparing four strategies that share the same total data budget D and high-quality fraction ρ:

- **Constant b + uniform p**: $b(t) = D/T$, $p(t) = \rho$ (naive baseline).
- **Constant b + late p**: $b(t) = D/T$, bang-bang quality placed late.
- **Uniform p + optimal b**: $p(t) = \rho$, $b(t) = \max(C\sqrt{K(T-t)}, B)$.
- **Joint optimal**: simultaneously optimal (b, p) from Theorem 4.3.

Figure 1 confirms that the joint optimal strategy achieves the lowest final risk in both regimes. In the noise-limited regime, the joint schedule outperforms batch-only optimization by exploiting the batch-size drop at the quality transition; in the signal-limited regime, it outperforms the other three strategies through the three-phase structure. Full experimental details are given in Appendix E.3.

Scaling-law verification. Theorem 4.3 predicts that the optimal excess risk scales as $\mathbb{E}[E(\bar{\boldsymbol{\theta}}_{T^\star})] \asymp D^{-s\beta/(1+s\beta)}$ in the noise-limited regime and $\mathbb{E}[E(\bar{\boldsymbol{\theta}}_{T^\star})] \asymp B^s D^{-s}$ in the signal-limited regime. To verify this, we sweep the total data budget D and measure the final excess risk achieved by each strategy. The signal-limited regime uses a smaller data range because its signal-constrained dynamics require longer per-sample training horizons ($T \propto D/B$), making large budgets computationally expensive. Figure 2 shows that the joint optimal strategy closely tracks the predicted power-law exponent in both regimes (-0.81 vs. theory -0.80 for noise-limited; -0.51 vs. theory -0.50 for signal-limited), while suboptimal strategies exhibit shallower scaling.

4.3 Interpreting the joint optimum

The key difference from original FSL is that the quality-aware FSL introduces data quality as an additional control variable. Instead of treating the noise level as fixed, the schedule can decide when to allocate high-quality, low-noise data through $p(t)$. This extra degree of freedom can be used in two dual ways: it can be converted into more optimization steps at a comparable noise scale, or into a lower noise scale at a comparable number of optimization steps.

Noise-limited regime ($s > 1 - 1/\beta$): quality as more signal. In this regime, the noise-accumulation integral is the bottleneck. The batch-only optimum (Wang et al., 2026) controls noise by ramping up the batch size, but it treats all data as having the same noise level. Once high-quality data is available, the joint optimum lowers the batch size during the low-noise phase. In particular, when entering the high-quality region S, the optimal batch size drops by a factor $r = \sigma^2_{\text{good}}/\sigma^2_{\text{bad}}$, so that

$$\sigma^2_{\text{good}}/b^\star|_S = \sigma^2_{\text{bad}}/b^\star|_{S^c}.$$

Thus, the local noise-injection rate remains comparable across quality phases, but the smaller batch size yields more optimization steps. In the noise-limited regime, high-quality data is therefore used to obtain more signal at roughly the same noise scale.

Signal-limited regime ($s < 1 - 1/\beta$): quality as lower noise. In this regime, the signal term t^{-s} is the bottleneck, so the optimal schedule must preserve as many optimization steps as possible. Both the batch-only and joint optima therefore spend most of training at the minimum batch size $B_{\min}$. Since the update count is already maximized, the joint optimum uses high-quality data in a different way: it places low-noise data late in training. This curriculum placement lowers the noise scale of late-stage updates without changing the leading-order signal-accumulation mechanism. In the signal-limited regime, high-quality data is therefore used to reduce noise at comparable signal accumulation.

Unifying principle. The two regimes show the same signal-noise trade-off from opposite sides. When noise is the bottleneck, high-quality data is converted into more signal without increasing noise. When signal is the bottleneck, high-quality data is converted into lower noise without sacrificing signal. As we will detail in Section 5, this duality motivates Drop-Stable-Rampup: drop the batch size after the quality transition to exploit low-noise data for more updates, keep it low to accumulate signal, and ramp up at the end to suppress terminal noise.

5 From Theory to Practice: Large-Scale LLM Experiments

5.1 The Drop-Stable-Rampup recipe

Building on Section 4.3, we instantiate the joint optimum at the pretraining-to-midtraining qual-

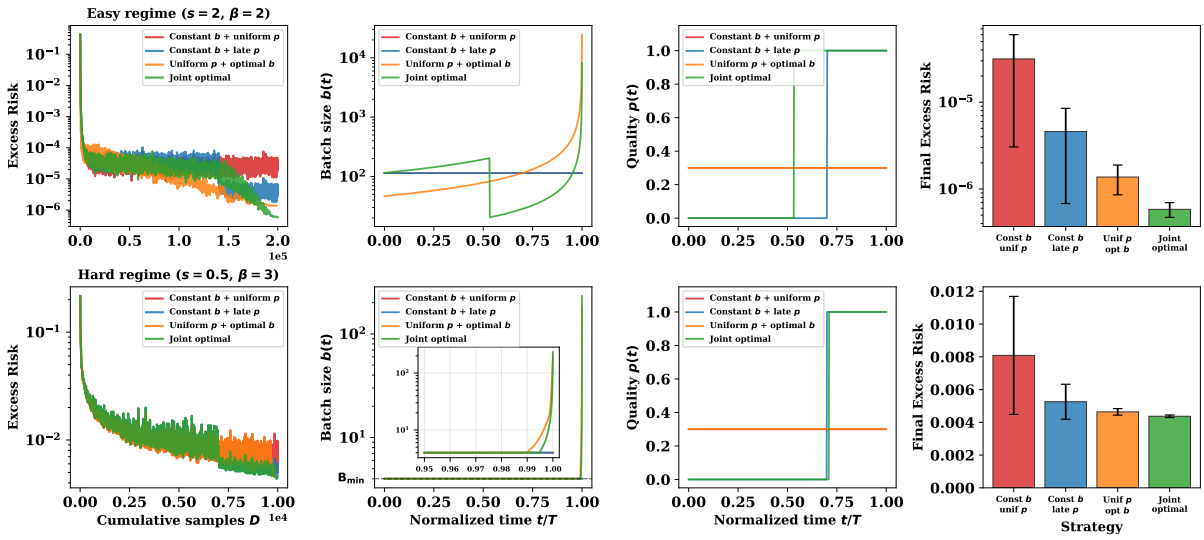

Figure 1: Verification of Theorem 4.3. Top: noise-limited ($s = 2$, $\beta = 2$); bottom: signal-limited ($s = 0.5$, $\beta = 3$). Four strategies share the same total data budget D and high-quality fraction ρ: (1) constant b + uniform p (baseline), (2) constant b + late p, (3) uniform p + optimal b, (4) joint optimal (Theorem 4.3). Columns show loss dynamics, batch-size schedules, data-quality schedules, and final excess risk. The joint optimal strategy achieves the lowest final risk in both regimes.

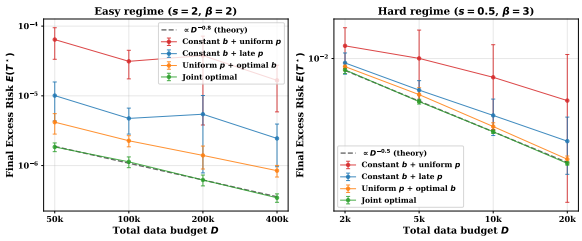

Figure 2: Scaling-law verification of Theorem 4.3. Final excess risk vs. total data budget D (log-log scale). Left: noise-limited regime ($s = 2$, $\beta = 2$, theory: $D^{-0.8}$); right: signal-limited regime ($s = 0.5$, $\beta = 3$, theory: $D^{-0.5}$). The joint optimal strategy matches the predicted power-law scaling in both regimes.

ity transition. A single LLM training run must serve capabilities that may fall into different empirical regimes: general knowledge and fluency often behave as noise-limited, whereas mathematical reasoning and some code-generation tasks exhibit signal-limited trajectories. Since midtraining uses a much smaller curated corpus than pretraining, low-batch training is feasible on existing hardware. This allows us to go beyond conventional curriculum learning with learning rate decay—which exploits high-quality data only for noise reduction. We additionally accumulate signal through increased update intensity. This leads to **Drop-Stable-Rampup**:

1. **Drop**: cut the batch size at the quality transition—accumulating signal via more gradient steps under low-noise conditions.
2. **Stable**: hold $B_{\min}$ for a period—maximizing signal acquisition for signal-limited capabilities.
3. **Rampup**: linearly grow the batch size (equivalently decay the learning rate (Smith et al., 2018))—suppressing accumulated noise for final convergence.

Setup. We evaluate on a Mixture-of-Experts (MoE) model with 15B total parameters and 2.4B active parameters, adopting the shortcut-connected MoE architecture of Cai et al. (2025). All experiments are initialized from a checkpoint pretrained on 1.2T tokens and conducted under a fixed budget of 108B tokens. All schedules are token-budget matched. Detailed model architecture, optimizer settings, and data composition are provided in Appendix F. We structure our evaluation into three parts: (1) verifying the batch-size drop effect across data corpora, (2) determining the optimal stable-phase ratio, (3) comparing Drop-Stable-Rampup (DSR) schedule against conventional baselines.

5.2 Verifying the effectiveness of dropping batch size

Theorem 4.3(I) predicts that when transitioning to low-noise data, the batch size should drop proportionally. To test whether this structural prediction transfers to LLMs, we compare three batch-size schedules: (1) Constant 4k, (2) Ramp-up 4k→8k,

and (3) Drop 1k→8k (i.e., dropping from the pretraining batch size of 4k down to 1k at the transition, then ramping to 8k). We test each on two corpora: pretraining data and midtraining data (see Appendix F.2 for details).

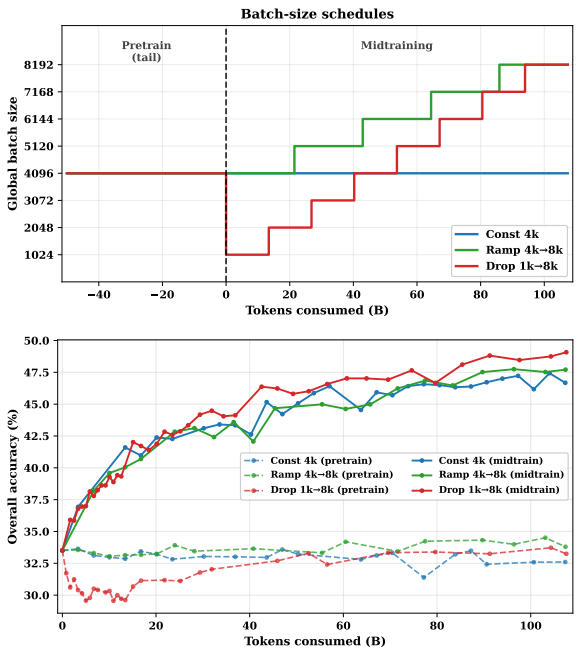

Figure 3: Batch-size schedules and overall accuracy curves for pretraining and midtraining corpora.

As shown in Table 1 and Figure 3, the results exhibit a sharp dichotomy that aligns with our theoretical predictions. On the pretraining corpus, the monotonic Ramp-up achieves the best overall performance (33.79), with Drop trailing slightly at 33.24. This is consistent with our theory: on pretraining data, the noise level σ^2_{eff} remains uniformly high, so the signal-accumulation benefit of smaller batch sizes cannot compensate for the increased noise injection. Without a quality transition, the Ramp-up remains preferable.

However, on the high-quality midtraining corpus—where the transition from pretraining to curated data creates precisely the noise drop our theory addresses—the trend reverses. Drop 1k→8k achieves the best overall score (49.07), outperforming the Ramp-up baseline by +1.36. The gains are broad-based, with notable improvements on GSM8K (+4.85), MATH (+3.86), and CRUX-O (+2.25), consistent with the prediction that dropping the batch size at a quality transition yields more gradient steps for signal accumulation under low-noise conditions.

5.3 Determining the optimal stable-phase ratio

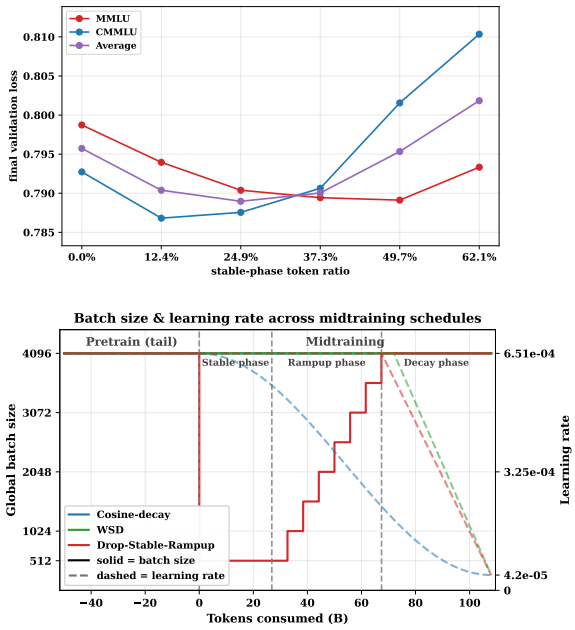

Figure 4: The final validation loss vs. stable-phase ratio (top) and schedule trajectories (bottom).

Theorem 4.3(II) predicts the existence of a flat low-batch-size phase before the terminal ramp-up, with an asymptotic duration that depends on problem-dependent constants (s, β, σ^2_{good}) not directly observable in LLM training. We therefore tune the stable-phase ratio empirically, scaling the batch-size bounds to [512, 4096] and varying the token ratio allocated to the stable phase (0%–62%) under the fixed 108B budget. The schedule has three phases: stable (batch size fixed at 512), ramp-up (linear increase 512→4096), and decay (learning rate linear decay equivalent to batch size linear growth (Smith et al., 2018)). We fix equal token allocation for the ramp-up and decay phases and sweep only the stable ratio.

We select the stable ratio using validation loss on MMLU and CMMLU. As shown in Figure 4, validation loss exhibits a U-shaped curve with optimal performance at a stable ratio of 24.9%. A too-short stable phase skips the signal-accumulation phase; a too-long stable phase leaves insufficient budget for the terminal ramp-up needed for noise suppression. The two validation sets also exhibit different preferences: MMLU benefits from a longer stable phase, suggesting stronger signal-accumulation sensitivity, whereas CMMLU degrades under overly long stable phases, suggesting greater reliance on noise suppression. Notably, every stable-ratio configura-

Table 1: Unified 15B MoE ablation results. **Bold** indicates the best result within each comparison block.

Category	Benchmark	Pretraining Corpus			Midtraining Corpus			Midtraining Schedule		
		4k	4k→8k	1k→8k	4k	4k→8k	1k→8k	Cos	WSD	DSR
Knowledge	MMLU	54.29	**56.05**	54.92	64.90	65.35	**66.40**	65.88	66.58	**67.27**
	MMLU-Pro	23.61	24.21	**26.29**	38.11	37.46	**39.07**	40.04	40.96	**41.89**
	CMMLU	59.25	**60.57**	60.00	66.02	66.07	**67.01**	67.43	67.61	**68.59**
	C-Eval	57.89	59.96	**59.99**	65.61	66.71	**67.37**	67.84	68.09	**68.24**
Reasoning	ARC-c	67.50	**69.50**	67.00	74.50	76.00	**77.50**	78.50	80.50	**81.00**
	BBH	36.70	38.03	**38.34**	**51.68**	50.30	49.60	48.13	52.19	**53.92**
Math	GSM8K	27.56	**29.72**	26.64	61.74	61.89	**66.74**	62.97	65.59	**69.82**
	MATH	9.20	9.38	**9.74**	30.48	32.22	**36.08**	33.98	36.86	**39.66**
Code	HE+	20.12	**21.95**	**21.95**	41.46	**45.12**	43.90	43.29	46.95	**49.39**
	BCB	21.23	**23.95**	22.46	35.09	37.02	**39.04**	**40.00**	35.79	38.60
	LCB	3.30	3.30	**3.74**	11.01	12.33	**12.78**	9.47	12.33	**13.88**
	CRUX-I	**30.38**	28.12	28.75	36.12	37.38	**38.50**	36.88	38.25	**39.75**
	CRUX-O	28.12	**28.62**	27.25	35.62	37.50	**39.75**	40.38	40.00	**40.88**
	MultiPL-E	17.26	**19.74**	18.34	41.11	42.54	**43.30**	43.95	44.87	**47.53**
Overall		32.60	**33.79**	33.24	46.68	47.71	**49.07**	48.48	49.76	**51.46**

tion in this sweep exceeds both WSD and Cosine-decay baselines on overall accuracy, demonstrating the robustness of the Drop-Stable-Rampup strategy (see Figure 8 for details).

5.4 Main results: schedule comparison

In this section, the Drop-Stable-Rampup schedule (with the optimal 24.9% stable ratio) is compared against two widely adopted baselines: Cosine-decay (Loshchilov and Hutter, 2017) and WSD (Hu et al., 2024). Both baselines use a constant batch size of 4096, consuming the same 108B token budget (see Figure 4 for schedule trajectories). Results are summarized in Table 1, with further trajectory details provided in Figure 9.

Drop-Stable-Rampup achieves the highest overall score of 51.46, outperforming WSD (49.76) and Cosine-decay (48.48). The largest gains appear on mathematical reasoning and algorithmic code generation: GSM8K (+4.23 over WSD), MATH (+2.80), HumanEval+ (+2.44), and MultiPL-E (+2.66). These benchmarks also exhibit the smallest early-phase gaps—DSR stays close to WSD during the stable phase and overtakes it during ramp-up—suggesting that the extended low-batch-size phase accumulates latent capacity that manifests after noise suppression. We provide a detailed per-benchmark analysis in Appendix G.

6 Conclusion

We study how LLMs should consume scarce high-quality data. By solving the joint data-quality and batch-size scheduling problem in asymptotic closed form, we show that optimal training schedules must jointly coordinate data quality and batch size. The resulting joint optimum reveals two regimes, but a shared principle: low-batch training after a quality transition accumulates signal, while late high-quality placement and terminal batch-size ramp-up suppress noise. This principle leads to Drop-Stable-Rampup, which improves LLM midtraining over standard schedules. More broadly, our results suggest that high-quality data should not merely be added to training; it should change how compute is allocated.

Limitations

Theoretical simplification. Our analysis is built on feature-space linear regression with Gaussian features and power-law spectral decay. While this modeling framework is standard in the scaling-law literature and has been useful for deriving qualitative predictions about training dynamics, it does not capture key aspects of LLM training: nonlinear feature learning, multi-layer compositional representations, and attention-based context dependence. Furthermore, the two-level data-quality model (σ^2_{good}

vs. σ_{bad}^2) is a stylized abstraction; real training corpora exhibit a continuous quality spectrum, and what constitutes "quality" is itself task-dependent. Extending the theory to richer noise models and nonlinear dynamics remains an important direction.

Multi-task heterogeneity. Our framework models data quality as a single scalar dimension (low-noise vs. high-noise), but LLM training is inherently multi-task: different data domains (math, code, knowledge) correspond to distinct subtasks with potentially different optimal schedules. As our empirical trajectory analysis shows (Appendix G), benchmarks differ substantially in their early-phase gaps and delayed gains, and a single schedule must compromise across these heterogeneous dynamics. A more fine-grained theory that jointly optimizes per-domain data mixtures $\{p_i(t)\}$ together with the batch-size $b(t)$ (or learning rate $\eta(t)$) schedule would better capture this heterogeneity, but remains an open problem.

Scale dependence of optimal hyperparameters. Our theory and experiments are conducted at a single model scale. As model size grows, the balance between signal accumulation and noise suppression may shift, potentially requiring different drop magnitudes, stable-phase durations, or ramp-up rates. Whether the specific hyperparameter choices (e.g., the batch-size drop factor, the 24.9% stable ratio) transfer to larger models, or how they should be adapted, remains an open question that our current framework does not address.

Experimental scope and overhead. We validate Drop-Stable-Rampup on a single architecture (15B MoE with 2.4B active parameters) under one midtraining setting (108B tokens from a fixed pretrain checkpoint); full details are in Appendix F. Whether the gains transfer to dense architectures, different model scales, or substantially different data budgets has not been tested. Additionally, the low-batch-size stable phase reduces hardware utilization in our setup, incurring a $1.1\times$ wall-clock overhead. This overhead is specific to our infrastructure configuration (Appendix F); different parallelism strategies or hardware may yield different ratios. In the midtraining regime, this trade-off may be acceptable when curated data is the binding constraint rather than compute, but its practicality depends on the available hardware and training infrastructure.

Ethical Considerations

Our work studies training schedules for LLM midtraining and does not release a new model, dataset, or user-facing system. The main risks are therefore indirect. First, the proposed schedule is evaluated through large-scale training experiments, which incur nontrivial computational cost. As discussed in the Limitations section, the low-batch-size stable phase reduces hardware utilization in our setup and leads to a $1.1\times$ wall-clock overhead; the practical and environmental cost of such schedules should be considered when scaling them further.

Second, our experiments use large-scale pretraining and midtraining corpora. Such corpora may inherit residual privacy, copyright, or offensive-content risks common to web-scale training data. For the proprietary corpora used in our experiments, data collection and processing follow the existing data-governance pipeline of the data provider, including automated filtering for sensitive or unsafe content. The corpora are used only for aggregate training experiments and benchmark evaluation; this work does not introduce, redistribute, or publish raw training data, nor does it release trained checkpoints. Since large-scale corpora cannot be manually audited in full, residual data-governance risks may remain.

Third, improving the efficiency of midtraining could lower the cost of capability improvement for both beneficial and potentially harmful downstream applications. We mitigate these risks by restricting the contribution to schedule-level analysis, reporting the relevant training setup and compute-efficiency trade-offs, and not releasing trained models or training corpora.

References

AI@Meta. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Stefan Ankirchner and Stefan Perko. 2024. A comparison of continuous-time approximations to stochastic gradient descent. *Journal of Machine Learning Research*, 25(13):1–55.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *International Conference on Machine Learning*, pages 41–48.

Blake Bordelon, Alexander Atanasov, and Cengiz Pehlevan. 2025. How feature learning can improve neural scaling laws. *Journal of Statistical Mechanics: Theory and Experiment*, 2025(8):084002.

Weilin Cai, Juyong Jiang, Le Qin, Junwei Cui, Sunghun Kim, and Jiayi Huang. 2025. Shortcut-connected expert parallelism for accelerating mixture of experts. In *International Conference on Machine Learning*.

Federico Cassano, John Gouwar, Daniel Nguyen, Sydney Nguyen, Luna Phipps-Costin, Donald Pinckney, Ming-Ho Yee, Yangtian Zi, Carolyn Jane Anderson, Molly Q. Feldman, Arjun Guha, Michael Greenberg, and Abhinav Jangda. 2023. MultiPL-E: A scalable and polyglot approach to benchmarking neural code generation. *IEEE Transactions on Software Engineering*, 49(7):3675–3691.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

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. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Alex Gu, Baptiste Roziere, Hugh James Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida Wang. 2024. CRUXEval: A benchmark for code reasoning, understanding and execution. In *International Conference on Machine Learning*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*.

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, and 5 others. 2024. MiniCPM: Unveiling the potential of small language models with scalable training strategies. In *Conference on Language Modeling*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, Yao Fu, Maosong Sun, and Junxian He. 2023. C-Eval: A multi-level multi-discipline chinese evaluation suite for foundation models. In *Advances in Neural Information Processing Systems*.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2025. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*.

Kimi Team. 2026. Kimi K2.5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*.

Binghui Li, Fengling Chen, Zixun Huang, Lean Wang, and Lei Wu. 2026. Functional scaling laws in kernel regression: Loss dynamics and learning rate schedules. *Advances in Neural Information Processing Systems*, 38:101211–101269.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024. CMMLU: Measuring massive multitask language understanding in chinese. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11260–11285.

Qianxiao Li, Cheng Tai, and Weinan E. 2019. Stochastic modified equations and dynamics of stochastic gradient algorithms I: Mathematical foundations. *Journal of Machine Learning Research*, 20(40):1–47.

Licong Lin, Jingfeng Wu, Sham M Kakade, Peter L Bartlett, and Jason D Lee. 2024. Scaling laws in linear regression: Compute, parameters, and data. *Advances in Neural Information Processing Systems*, 37:60556–60606.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2023. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems*, 36:21558–21572.

Ilya Loshchilov and Frank Hutter. 2017. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*.

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

Kairong Luo, Zhenbo Sun, Haodong Wen, Xinyu Shi, Jiarui Cui, Chenyi Dang, Kaifeng Lyu, and Wenguang Chen. 2026. How learning rate decay wastes your best data in curriculum-based LLM pretraining. In *International Conference on Learning Representations*.

Takashi Mori, Liu Ziyin, Kangqiao Liu, and Masahito Ueda. 2022. Power-law escape rate of SGD. In *International Conference on Machine Learning*, pages 15959–15975.

Antonio Orvieto and Aurelien Lucchi. 2019. Continuous-time models for stochastic optimization algorithms. *Advances in Neural Information Processing Systems*, 32.

Elliot Paquette, Courtney Paquette, Lechao Xiao, and Jeffrey Pennington. 2024. 4+3 phases of compute-optimal neural scaling laws. *Advances in Neural Information Processing Systems*, 37:16459–16537.

Qwen Team. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-LM: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*.

Samuel L. Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V. Le. 2018. Don't decay the learning rate, increase the batch size. In *International Conference on Learning Representations*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging Big-Bench tasks and whether Chain-of-Thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051.

Jinbo Wang, Binghui Li, Zhanpeng Zhou, Mingze Wang, Yuxuan Sun, Jiaqi Zhang, Xunliang Cai, and Lei Wu. 2026. Fast catch-up, late switching: Optimal batch size scheduling via functional scaling laws. In *International Conference on Learning Representations*.

Mingze Wang and Lei Wu. 2023. A theoretical analysis of noise geometry in stochastic gradient descent. *arXiv preprint arXiv:2310.00692*.

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. 2024. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.

Lei Wu, Mingze Wang, and Weijie Su. 2022. The alignment property of SGD noise and how it helps select flat minima: A stability analysis. *Advances in Neural Information Processing Systems*, 35:4680–4693.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, Simon Brunner, Chen Gong, Thong Hoang, Armel Zebaze, Xiaoheng Hong, Wen-Ding Li, Jean Kaddour, Ming Xu, Zhihan Zhang, and 14 others. 2025. BigCodeBench: Benchmarking code generation with diverse function calls and complex instructions. In *International Conference on Learning Representations*.

A Gradient-Noise Covariance

Let

$$\boldsymbol{e} = \boldsymbol{\theta} - \boldsymbol{\theta}^\star.$$

For one sample, the stochastic gradient is

$$\begin{aligned}\boldsymbol{g}(\boldsymbol{\theta}; \boldsymbol{x}, y) &= \big(\langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\theta}\rangle - y\big)\boldsymbol{\phi}(\boldsymbol{x}) \\ &= \big(\langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{e}\rangle - \epsilon\big)\boldsymbol{\phi}(\boldsymbol{x}).\end{aligned}$$

Taking expectation over $\boldsymbol{x}$ and ϵ, and using $\mathbb{E}[\boldsymbol{\phi}(\boldsymbol{x})] = \mathbf{0}$, we obtain $\mathbb{E}[\boldsymbol{g}(\boldsymbol{\theta}; \boldsymbol{x}, y)] = \boldsymbol{H}\boldsymbol{e}$, confirming that the stochastic gradient is unbiased.

The covariance of the single-sample gradient noise at time t is

$$\boldsymbol{\Sigma}(\boldsymbol{\theta}) = \mathrm{Cov}\big((\langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{e}\rangle - \epsilon)\boldsymbol{\phi}(\boldsymbol{x})\big).$$

Since the label noise is independent of $\boldsymbol{x}$,

$$\boldsymbol{\Sigma}(\boldsymbol{\theta}) = \mathrm{Cov}\big(\langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{e}\rangle \boldsymbol{\phi}(\boldsymbol{x})\big) + \sigma_{\text{eff}}^2(t)\boldsymbol{H}.$$

Lemma A.1 (Quality-aware anisotropic noise)**.** *For any $\boldsymbol{\theta} \in \mathbb{R}^N$ and any time t, the single-sample gradient-noise covariance satisfies*

$$\boldsymbol{v}_j^\top \boldsymbol{\Sigma}(\boldsymbol{\theta})\boldsymbol{v}_j \asymp \lambda_j\big(E(\boldsymbol{\theta}) + \sigma_{\text{eff}}^2(t)\big),$$

where $\boldsymbol{v}_j$ is the eigenvector of $\boldsymbol{H}$ with eigenvalue λ_j, and the constants are independent of j, t, and $\boldsymbol{\theta}$.

Proof. Since $\boldsymbol{\phi}(\boldsymbol{x}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{H})$, Wick's formula gives

$$\begin{aligned}&\mathbb{E}\big[\boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{e}\boldsymbol{e}^\top \boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{\phi}(\boldsymbol{x})^\top\big] \\ &\quad = 2\boldsymbol{H}\boldsymbol{e}\boldsymbol{e}^\top\boldsymbol{H} + (\boldsymbol{e}^\top\boldsymbol{H}\boldsymbol{e})\boldsymbol{H}.\end{aligned}$$

Therefore,

$$\begin{aligned}\mathrm{Cov}\big(\langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{e}\rangle \boldsymbol{\phi}(\boldsymbol{x})\big) =& \mathbb{E}\big[\boldsymbol{\phi}\boldsymbol{\phi}^\top \boldsymbol{e}\boldsymbol{e}^\top \boldsymbol{\phi}\boldsymbol{\phi}^\top\big] \\ &- \boldsymbol{H}\boldsymbol{e}\boldsymbol{e}^\top\boldsymbol{H} \\ =& \boldsymbol{H}\boldsymbol{e}\boldsymbol{e}^\top\boldsymbol{H} + (\boldsymbol{e}^\top\boldsymbol{H}\boldsymbol{e})\boldsymbol{H}.\end{aligned}$$

Since $\boldsymbol{e}^\top\boldsymbol{H}\boldsymbol{e} = 2E(\boldsymbol{\theta})$, we have

$$\boldsymbol{\Sigma}(\boldsymbol{\theta}) = \boldsymbol{H}\boldsymbol{e}\boldsymbol{e}^\top\boldsymbol{H} + 2E(\boldsymbol{\theta})\boldsymbol{H} + \sigma_{\text{eff}}^2(t)\boldsymbol{H}.$$

The lower bound follows from $\boldsymbol{H}\boldsymbol{e}\boldsymbol{e}^\top\boldsymbol{H} \succeq 0$. For the upper bound, we have

$$\boldsymbol{H}\boldsymbol{e}\boldsymbol{e}^\top\boldsymbol{H} \preceq (\boldsymbol{e}^\top\boldsymbol{H}\boldsymbol{e})\boldsymbol{H} = 2E(\boldsymbol{\theta})\boldsymbol{H}.$$

Hence

$$\boldsymbol{\Sigma}(\boldsymbol{\theta}) \preceq \big(4E(\boldsymbol{\theta}) + \sigma_{\text{eff}}^2(t)\big)\boldsymbol{H}.$$

This proves the lemma. □

B Proof of Theorem 3.1

Proof. Let $\boldsymbol{e}_t = \bar{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star$. Without loss of generality, assume zero initialization $\bar{\boldsymbol{\theta}}_0 = \mathbf{0}$, so that $\boldsymbol{e}_0 = -\boldsymbol{\theta}^\star$ and $e_{0,j}^2 = |\theta_j^\star|^2$. Since $\nabla R(\bar{\boldsymbol{\theta}}_t) = \boldsymbol{H}\boldsymbol{e}_t$, the SDE (2) decouples coordinate-wise: each $e_{t,j}$ satisfies a scalar Ornstein-Uhlenbeck equation with decay rate λ_j. Integrating via Duhamel's formula yields

$$\begin{aligned}e_{t,j} &= e^{-\lambda_j t} e_{0,j} \\ &\quad + \int_0^t e^{-\lambda_j(t-\tau)} \sqrt{\frac{\eta}{b(\tau)}}\, \big[\boldsymbol{\Sigma}(\bar{\boldsymbol{\theta}}_\tau)^{1/2}\, \mathrm{d}\boldsymbol{W}_\tau\big]_j.\end{aligned}$$

By Itô isometry,

$$\begin{aligned}\mathbb{E}[e_{t,j}^2] &= e^{-2\lambda_j t} e_{0,j}^2 \\ &\quad + \int_0^t e^{-2\lambda_j(t-\tau)} \frac{\eta}{b(\tau)} \mathbb{E}\big[q_j(\tau)\big]\, \mathrm{d}\tau,\end{aligned}$$

where

$$q_j(\tau) := \boldsymbol{v}_j^\top \boldsymbol{\Sigma}(\bar{\boldsymbol{\theta}}_\tau)\boldsymbol{v}_j.$$

Therefore

$$\begin{aligned}2\mathbb{E}[E(\bar{\boldsymbol{\theta}}_t)] &= \sum_{j\geq 1} \lambda_j e^{-2\lambda_j t} e_{0,j}^2 \\ &\quad + \sum_{j\geq 1} \lambda_j \int_0^t e^{-2\lambda_j(t-\tau)} \frac{\eta}{b(\tau)} \mathbb{E}[q_j(\tau)]\, \mathrm{d}\tau.\end{aligned}$$

By Lemma A.1,

$$\mathbb{E}[q_j(\tau)] \asymp \lambda_j\big(\mathbb{E}[E(\bar{\boldsymbol{\theta}}_\tau)] + \sigma_{\text{eff}}^2(\tau)\big).$$

Hence, if

$$f(t) := \mathbb{E}[E(\bar{\boldsymbol{\theta}}_t)],$$

then f satisfies the Volterra-type relation

$$f(t) \asymp e(t) + \int_0^t \frac{\eta\, K(t-\tau)}{b(\tau)} \big(f(\tau) + \sigma_{\text{eff}}^2(\tau)\big)\, \mathrm{d}\tau,$$

where

$$e(t) := \sum_{j\geq 1} \lambda_j |\theta_j^\star|^2 e^{-2\lambda_j t}, K(t) := \sum_{j\geq 1} \lambda_j^2 e^{-2\lambda_j t}.$$

Under the power-law assumptions $\lambda_j \asymp j^{-\beta}$ and $|\theta_j^\star|^2 \asymp j^{-1}\lambda_j^{s-1} \asymp j^{-(1+(s-1)\beta)}$, the sums defining $e(t)$ and $K(t)$ can be approximated by integrals via Euler-Maclaurin comparison. Substituting the continuous variable $x = \lambda_j t$ and using $j \asymp \lambda_j^{-1/\beta}$, one obtains

$$\begin{aligned}e(t) &= \sum_{j\geq 1} \lambda_j |\theta_j^\star|^2 e^{-2\lambda_j t} \\ &\asymp \int_0^\infty x^{s-1} e^{-2x}\, \mathrm{d}x \cdot t^{-s} \asymp t^{-s},\end{aligned}$$

and similarly

$$K(t) = \sum_{j\geq 1} \lambda_j^2 e^{-2\lambda_j t}$$
$$\asymp \int_0^\infty x^{1-1/\beta} e^{-2x}\,\mathrm{d}x \cdot t^{-(2-1/\beta)}$$
$$\asymp (t+1)^{-(2-1/\beta)}.$$

Now define

$$g(t) := e(t) + \eta \int_0^t K(t-\tau)\frac{\sigma_{\text{eff}}^2(\tau)}{b(\tau)}\,\mathrm{d}\tau,$$

and define the linear Volterra operator

$$(\mathcal{T}f)(t) := \eta \int_0^t K(t-\tau)\frac{f(\tau)}{b(\tau)}\,\mathrm{d}\tau.$$

With this notation, the preceding relation admits the compact representation

$$f = g + \mathcal{T}f$$

up to universal multiplicative constants. Formally,

$$f = (I-\mathcal{T})^{-1}g = \sum_{m=0}^{\infty} \mathcal{T}^m g.$$

It remains to check that the higher-order terms generated by the multiplicative part of the gradient noise do not change the leading scaling. Since $K(t) \asymp (t+1)^{-(2-1/\beta)}$ is nonincreasing and integrable for $\beta > 1$,

$$(K * K)(t) = \int_0^t K(t-\tau)K(\tau)\,\mathrm{d}\tau \lesssim K(t).$$

Using $b(t) \geq B_{\min} \geq 1$, we obtain

$$\mathcal{T}^2 g(t) \lesssim \eta\,\mathcal{T}g(t).$$

Iterating this estimate and taking η small enough yields

$$f(t) \asymp g(t) + \mathcal{T}g(t).$$

Moreover, because $\sigma_{\text{good}}^2 \leq \sigma_{\text{eff}}^2(t) \leq \sigma_{\text{bad}}^2$, the term $\mathcal{T}g(t)$ is bounded by the same two leading quantities contained in $g(t)$. Indeed,

$$\mathcal{T}e(t) \lesssim \eta \int_0^t K(t-\tau)\frac{1}{b(\tau)}\,\mathrm{d}\tau$$
$$\lesssim \eta \int_0^t K(t-\tau)\frac{\sigma_{\text{eff}}^2(\tau)}{b(\tau)}\,\mathrm{d}\tau,$$

and

$$\mathcal{T}\left[\eta \int_0^{\cdot} K(\cdot - r)\frac{\sigma_{\text{eff}}^2(r)}{b(r)}\,\mathrm{d}r\right](t)$$
$$\lesssim \eta \int_0^t K(t-r)\frac{\eta\sigma_{\text{eff}}^2(r)}{b(r)}\,\mathrm{d}r.$$

Thus $\mathcal{T}g(t) \lesssim g(t)$, and consequently

$$f(t) \asymp e(t) + \eta \int_0^t K(t-\tau)\frac{\sigma_{\text{eff}}^2(\tau)}{b(\tau)}\,\mathrm{d}\tau.$$

Substituting $e(t) \asymp t^{-s}$ proves the theorem. □

C Proof of Theorem 4.1

Proof. The optimization problem is

$$\max_{0\leq p\leq 1} \int_0^T \frac{K(T-t)}{b(t)} p(t)\,\mathrm{d}t$$

subject to

$$\int_0^T p(t)b(t)\,\mathrm{d}t = \rho D.$$

Introduce the measure $\mathrm{d}\mu(t) := b(t)\,\mathrm{d}t$. Then the constraint becomes $\int_0^T p(t)\,\mathrm{d}\mu(t) = \rho D$, and the objective becomes

$$\int_0^T S_b(t)p(t)\,\mathrm{d}\mu(t).$$

This is a continuous knapsack problem. The optimizer fills the high-quality budget on the set where $S_b(t)$ is largest, giving the stated bang-bang structure. □

D Proof of Theorem 4.3

Throughout this section we write $\sigma_1^2 := \sigma_{\text{good}}^2$ and $\sigma_2^2 := \sigma_{\text{bad}}^2$ for brevity, and denote the objective of (3) by $\mathcal{E}[T, b, p]$.

For data budget $D > 0$, define the feasible set

$$\begin{aligned}\mathcal{B}_D := \big\{(T,b,p) :\ & T > 0,\ b \in L^1(0,T),\\ & p \in L^\infty(0,T),\\ & b(t) \geq B,\ 0 \leq p(t) \leq 1 \text{ a.e.},\\ & \textstyle\int_0^T b(t)\,\mathrm{d}t = D,\\ & \textstyle\int_0^T b(t)p(t)\,\mathrm{d}t = \rho D\big\}.\end{aligned}$$

We consider

$$\mathcal{E}^\star(D) := \min_{(T,b,p)\in\mathcal{B}_D} \mathcal{E}[T,b,p],$$

where

$$\mathcal{E}[T,b,p] := T^{-s} + \eta \int_0^T \frac{K(T-t)\,\sigma_{\text{eff}}^2(t)}{b(t)}\,\mathrm{d}t.$$

For the power kernel $K(u) = (u+1)^{-\gamma}$ with $\gamma := 2 - 1/\beta > 0$, the horizon is bounded by $T \leq D/B$ (from $b \geq B$) and bounded away from zero by the bias term T^{-s}, so the problem is well posed and the standard pointwise Lagrangian characterization for decomposable integral programs applies.

Lemma D.1 (Canonical structure of an optimal joint schedule)**.** *Assume*

$$B > 0, \quad D > 0, \quad 0 < \rho < 1, \quad \sigma_1 < \sigma_2.$$

Write $w(t) := K(T^\star - t)$. Then an optimal solution can be chosen to have the following canonical form.

1. ***Strict-switching case.*** *There exist $T_s \in [0, T^\star]$ and constants $C_0, C_1 > 0$ such that*

$$p^\star(t) = \begin{cases} 0, & 0 \le t < T_s, \\ 1, & T_s \le t \le T^\star, \end{cases}$$

and

$$b^\star(t) = \begin{cases} \max\{B,\, C_0\sqrt{w(t)}\}, & t < T_s, \\ \max\{B,\, C_1\sqrt{w(t)}\}, & t \ge T_s. \end{cases}$$

2. ***Terminal-tie case.*** *There exist $T_s \in [0, T^\star]$ and $C > 0$ such that $p^\star(t) = 0$ for $t < T_s$, while on $I_s := [T_s, T^\star]$ both labels are pointwise optimal and $p^\star(t) \in \{0, 1\}$ may be chosen measurably subject to*

$$\int_0^{T^\star} b^\star(t) p^\star(t)\, \mathrm{d}t = \rho D.$$

The batch schedule is

$$b^\star(t) = \begin{cases} \max\{B,\, C\sqrt{w(t)}\}, & t < T_s, \\ r^{p^\star(t)}\, C\sqrt{w(t)}, & t \in I_s, \end{cases}$$

where $r = \sigma_1^2/\sigma_2^2$, with $rC\sqrt{w(t)} \ge B$ on I_s.

Proof. Fix an optimal solution and write it as

$$(T^\star, b^\star, p^\star).$$

Since $(T^\star)^{-s}$ is a constant, it suffices to minimize the noise integral over (b, p) at this fixed horizon. Recall $w(t) = K(T^\star - t)$, which is nondecreasing in t.

Introduce Lagrange multipliers μ, ν for the data and quality constraints. Since the integrand is pointwise separable in t, the optimizer satisfies the following pointwise condition for a.e. t:

$$(b^\star(t), p^\star(t)) \in \operatorname*{arg\,min}_{\substack{b \ge B \\ 0 \le p \le 1}} \Big\{ \frac{w(t)}{b} \big(\sigma_2^2 - (\sigma_2^2 - \sigma_1^2)p\big) + \mu b + \nu b p \Big\}.$$

Let

$$\Delta := \sigma_2^2 - \sigma_1^2 > 0.$$

For $i = 0, 1$, define

$$a_0 := \mu, \qquad c_0 := \sigma_2^2,$$

and

$$a_1 := \mu + \nu, \qquad c_1 := \sigma_1^2.$$

For a fixed t, b, the two endpoint label costs are

$$L_0(t, b) := \frac{c_0 w(t)}{b} + a_0 b,$$

and

$$L_1(t, b) := \frac{c_1 w(t)}{b} + a_1 b.$$

The pointwise Lagrangian can be written as

$$(1 - p)L_0(t, b) + pL_1(t, b).$$

Define the reduced endpoint costs

$$\Phi_i(w) := \min_{b \ge B} \Big\{ \frac{c_i w}{b} + a_i b \Big\}, \qquad i = 0, 1.$$

Then pointwise optimality gives

$$p^\star(t) = 0 \quad \text{if} \quad \Phi_0(w(t)) < \Phi_1(w(t)),$$

and

$$p^\star(t) = 1 \quad \text{if} \quad \Phi_1(w(t)) < \Phi_0(w(t)).$$

Indeed, if $\Phi_0(w) < \Phi_1(w)$, then for every $b \ge B$ and $p > 0$,

$$(1 - p)L_0(t, b) + pL_1(t, b) > \Phi_0(w),$$

while $p = 0$ and a minimizer of L_0 achieve $\Phi_0(w)$. The case $\Phi_1(w) < \Phi_0(w)$ is symmetric. Thus nonendpoint choices of p can only occur on tie sets where

$$\Phi_0(w(t)) = \Phi_1(w(t)).$$

On such tie sets, possible fractional values of p may be removed without changing feasibility or the objective. Indeed, let

$$F := \{t : 0 < p^\star(t) < 1\}.$$

On F, the first-order condition in p gives

$$\Delta \frac{w(t)}{b^\star(t)} = \nu b^\star(t).$$

Since the measure $b^\star(t)\, \mathrm{d}t$ is nonatomic, there exists a measurable set $S \subseteq F$ such that

$$\int_S b^\star(t)\, \mathrm{d}t = \int_F b^\star(t) p^\star(t)\, \mathrm{d}t.$$

Define

$$\tilde{p}(t) = \begin{cases} 1, & t \in S, \\ 0, & t \in F \setminus S, \\ p^\star(t), & t \notin F. \end{cases}$$

Then

$$\int_0^{T^\star} b^\star(t)\tilde{p}(t)\,\mathrm{d}t = \int_0^{T^\star} b^\star(t)p^\star(t)\,\mathrm{d}t.$$

Moreover, the objective change on F is

$$\begin{aligned} &-\Delta \int_F \frac{w(t)}{b^\star(t)}(\tilde{p}(t) - p^\star(t))\,\mathrm{d}t \\ &= -\nu \int_F b^\star(t)(\tilde{p}(t) - p^\star(t))\,\mathrm{d}t = 0. \end{aligned}$$

Hence we may choose an optimal solution with

$$p^\star(t) \in \{0, 1\} \quad \text{a.e.}$$

We next show that

$$\nu > 0.$$

Suppose $\nu \le 0$. For any t with $w(t) > 0$ and any $b \ge B$,

$$L_1(t, b) - L_0(t, b) = -\frac{\Delta w(t)}{b} + \nu b < 0.$$

Hence $p^\star(t) = 1$ for a.e. $t \in [0, T^\star]$, contradicting

$$\int_0^{T^\star} b^\star(t)(1 - p^\star(t))\,\mathrm{d}t = (1 - \rho)D > 0.$$

Therefore

$$\nu > 0.$$

Consequently,

$$a_1 > a_0, \qquad c_1 < c_0.$$

Moreover, the endpoint reduced costs are finite and attained only if $a_0 > 0$ and $a_1 > 0$.

We now prove that the $p = 1$ region is terminal (i.e., corresponds to late training times) and identify the two possible canonical structures. A direct calculation gives

$$\Phi_i(w) = \begin{cases} a_i B + \dfrac{c_i w}{B}, & 0 < w \le w_i, \\ 2\sqrt{a_i c_i w}, & w \ge w_i, \end{cases}$$

where $w_i := a_i B^2 / c_i$. Since $a_1 > a_0$ and $c_1 < c_0$, we have $w_0 < w_1$.

Claim. The ratio $R(w) := \Phi_1(w)/\Phi_0(w)$ is nonincreasing on $(0, \infty)$, with

$$\lim_{w \downarrow 0} R(w) = \frac{a_1}{a_0} > 1, \quad \lim_{w \to \infty} R(w) = \sqrt{\frac{a_1 c_1}{a_0 c_0}}.$$

Proof of Claim. We verify the monotonicity on the three regions determined by $0 < w_0 < w_1$.

Region $0 < w \le w_0$ (both constrained):

$$R(w) = \frac{a_1 B + c_1 w/B}{a_0 B + c_0 w/B}.$$

Differentiating, the numerator of $R'(w)$ is proportional to $a_0 c_1 - a_1 c_0 < 0$. Hence R is strictly decreasing.

Region $w_0 \le w \le w_1$ (Φ_0 unconstrained, Φ_1 constrained):

$$\begin{aligned} R(w) &= \frac{a_1 B + c_1 w/B}{2\sqrt{a_0 c_0 w}} \\ &= \frac{a_1 B}{2\sqrt{a_0 c_0}}\, w^{-1/2} + \frac{c_1}{2B\sqrt{a_0 c_0}}\, w^{1/2}. \end{aligned}$$

Differentiating, the sign of $R'(w)$ is that of $c_1 w/B - a_1 B$, which is negative for $w < w_1$. Hence R is strictly decreasing.

Region $w \ge w_1$ (both unconstrained):

$$R(w) = \frac{2\sqrt{a_1 c_1 w}}{2\sqrt{a_0 c_0 w}} = \sqrt{\frac{a_1 c_1}{a_0 c_0}},$$

which is constant. Continuity at w_0 and w_1 is verified by direct substitution. □

Since R is nonincreasing with $R(0^+) > 1$, the set $\{w : R(w) \le 1\}$ is either empty or a terminal half-line. Because $w(t) = K(T^\star - t)$ is nondecreasing in t, the preimage $\{t : p^\star(t) = 1\}$ is a terminal interval in time (up to the threshold level set, on which the split may be chosen measurably). The structure depends on $\lim_{w\to\infty} R(w) = \sqrt{a_1 c_1/(a_0 c_0)}$.

Case 1: $a_1 c_1 > a_0 c_0$**.** Then $R(w) > 1$ for all $w > 0$, so pointwise optimality forces $p^\star(t) = 0$ a.e., contradicting $\int_0^{T^\star} b^\star(t)p^\star(t)\,\mathrm{d}t = \rho D > 0$. This case is excluded.

Case 2: $a_1 c_1 < a_0 c_0$ **(strict-switching).** Then $R(\infty) < 1$. From the proof of claim, the equation $R(w) = 1$ has a unique solution. Hence there exists $T_s \in [0, T^\star]$ such that

$$p^\star(t) = \begin{cases} 0, & 0 \le t < T_s, \\ 1, & T_s \le t \le T^\star. \end{cases}$$

On each region, the optimal batch size is obtained by minimizing $L_i(t, b) = c_i w(t)/b + a_i b$ over

$b \geq B$, giving $b^\star(t) = \max\{B,\, \sqrt{c_i/a_i}\sqrt{w(t)}\}$. Writing

$$C_0 := \sqrt{\tfrac{c_0}{a_0}} = \sqrt{\tfrac{\sigma_2^2}{\mu}},$$

$$C_1 := \sqrt{\tfrac{c_1}{a_1}} = \sqrt{\tfrac{\sigma_1^2}{\mu+\nu}},$$

we obtain

$$b^\star(t) = \begin{cases} \max\{B,\, C_0\sqrt{w(t)}\}, & 0 \leq t < T_s, \\ \max\{B,\, C_1\sqrt{w(t)}\}, & T_s \leq t \leq T^\star. \end{cases}$$

Case 3: $a_1c_1 = a_0c_0$ **(terminal-tie).** Then $R(w) \geq 1$ for all w, with $R(w) = 1$ exactly on $[w_1, \infty)$ (since R is strictly decreasing on $(0, w_1)$ and constant on $[w_1, \infty)$). The corresponding tie interval in time is $I_s := [T_s, T^\star]$. Before I_s, we have $p^\star(t) = 0$. On I_s, both labels are pointwise optimal, and $p^\star(t) \in \{0, 1\}$ may be chosen measurably subject to $\int_0^{T^\star} b^\star(t)p^\star(t)\,\mathrm{d}t = \rho D$.

From $a_1c_1 = a_0c_0$, i.e., $(\mu + \nu)\sigma_1^2 = \mu\sigma_2^2$, we obtain $C_1 = rC_0$ where $r = \sigma_1^2/\sigma_2^2 \in (0, 1)$. Writing $C := C_0$:

$$b^\star(t) = \begin{cases} \max\{B,\, C\sqrt{w(t)}\}, & t < T_s, \\ C\sqrt{w(t)}, & t \in I_s,\ p^\star = 0, \\ rC\sqrt{w(t)}, & t \in I_s,\ p^\star = 1. \end{cases}$$

Since I_s lies in the unconstrained region $w \geq w_1$, we have $rC\sqrt{w(t)} \geq B$ on I_s.

After excluding Case 1, only the strict-switching and terminal-tie canonical forms remain. This proves the lemma. □

Proof of Theorem 4.3. We prove the noise-limited regime ($s > 1-1/\beta$) and the signal-limited regime ($s < 1 - 1/\beta$) separately. Throughout, write $\delta := 1/(2\beta)$.

Step 1: the unconstrained problem. First ignore the lower-bound constraint $b(t) \geq B$. Fix $T > 0$. By Lemma D.1, an optimizer can be chosen with

$$p(t) \in \{0, 1\}$$

a.e. Let

$$S := \{t : p(t) = 1\}, \quad S^c := \{t : p(t) = 0\}.$$

For fixed S, Cauchy-Schwarz gives

$$\int_S \frac{K(T-t)}{b(t)}\,\mathrm{d}t \geq \frac{\left(\int_S \sqrt{K(T-t)}\,\mathrm{d}t\right)^2}{\int_S b(t)\,\mathrm{d}t},$$

and similarly on S^c. Equality holds iff $b(t) \propto \sqrt{K(T-t)}$ on each region.

Let

$$A_1 := \int_S \sqrt{K(T-t)}\,\mathrm{d}t,$$

$$A_0 := \int_{S^c} \sqrt{K(T-t)}\,\mathrm{d}t,$$

and $A(T) := A_0 + A_1 = \int_0^T \sqrt{K(T-t)}\,\mathrm{d}t$. The data constraints give

$$\int_S b(t)\,\mathrm{d}t = \rho D, \qquad \int_{S^c} b(t)\,\mathrm{d}t = (1-\rho)D.$$

Thus, for fixed T, the unconstrained variance term

$$\sigma_2^2 \int_{S^c} \frac{K(T-t)}{b(t)}\,\mathrm{d}t \;+\; \sigma_1^2 \int_S \frac{K(T-t)}{b(t)}\,\mathrm{d}t$$

is at least

$$\frac{\sigma_2^2 A_0^2}{(1-\rho)D} + \frac{\sigma_1^2 A_1^2}{\rho D}.$$

Minimizing over $A_0 + A_1 = A(T)$ gives

$$A_0^\star = \frac{(1-\rho)\sigma_1^2}{(1-\rho)\sigma_1^2 + \rho\sigma_2^2}\,A(T),$$

$$A_1^\star = \frac{\rho\sigma_2^2}{(1-\rho)\sigma_1^2 + \rho\sigma_2^2}\,A(T).$$

The minimized variance term is

$$\frac{\eta\,\kappa\,A(T)^2}{D}, \qquad \kappa = \frac{\sigma_1^2\sigma_2^2}{(1-\rho)\sigma_1^2 + \rho\sigma_2^2}.$$

Hence the unconstrained problem reduces to

$$\min_{T>0}\left\{T^{-s} + \frac{\eta\,\kappa\,A(T)^2}{D}\right\}.$$

The first-order condition is

$$-sT^{-s-1} + \frac{2\eta\,\kappa}{D}A(T)A'(T) = 0.$$

Since

$$A'(T) = (T+1)^{\delta-1}$$

and

$$A(T) = \frac{T^\delta}{\delta}(1 + o(1)),$$

we obtain

$$T_{\mathrm{unc}} = \left(\frac{sD}{4\beta\,\eta\,\kappa}\right)^{\frac{1}{s+1/\beta}}(1 + o(1)).$$

At this optimum,

$$\frac{\eta\,\kappa\,A(T_{\mathrm{unc}})^2}{D} = s\beta T_{\mathrm{unc}}^{-s}(1 + o(1)).$$

Therefore

$$\mathcal{E}_{\mathrm{unc}}^\star(D) = (1 + s\beta)T_{\mathrm{unc}}^{-s}(1 + o(1)).$$

Step 2: the noise-limited regime. We first record the explicit form of the unconstrained optimal batch profile. From the Cauchy-Schwarz step in Step 1, equality requires $b(t) \propto \sqrt{K(T-t)}$ on each of the two regions S and S^c. Hence the unconstrained optimizer at horizon T_{unc} takes the piecewise form

$$b^\star(t) = \begin{cases} C_0\sqrt{K(T_{\text{unc}}-t)}, & t \in S^c \ (p^\star = 0), \\ C_1\sqrt{K(T_{\text{unc}}-t)}, & t \in S \ (p^\star = 1), \end{cases}$$

where C_0, C_1 are the same constants as in Lemma D.1, evaluated at $T = T_{\text{unc}}$. They are determined by the data constraints $\int_{S^c} b^\star = (1-\rho)D$ and $\int_S b^\star = \rho D$:

$$C_0 = \frac{(1-\rho)D}{A_0^\star}, \qquad C_1 = \frac{\rho D}{A_1^\star}.$$

Substituting the optimal split $(A_0^\star, A_1^\star)$ found in Step 1 yields

$$C_0 = \frac{(1-\rho)\sigma_1^2 + \rho\sigma_2^2}{\sigma_1^2} \cdot \frac{D}{A(T_{\text{unc}})},$$
$$C_1 = \frac{(1-\rho)\sigma_1^2 + \rho\sigma_2^2}{\sigma_2^2} \cdot \frac{D}{A(T_{\text{unc}})}.$$

Since $\sigma_1 < \sigma_2$, we have $C_1 < C_0$; since K is decreasing, $\sqrt{K(T_{\text{unc}}-t)}$ is minimized at $t = 0$, and consequently so is $b^\star(t)$. Hence

$$\min_{t\in[0,T_{\text{unc}}]} b^\star(t) = C_1(T_{\text{unc}}+1)^{\delta-1}.$$

Using $A(T_{\text{unc}}) \asymp T_{\text{unc}}^\delta/\delta$ together with the explicit form of C_1, this gives

$$\min_t b^\star(t) \asymp \frac{D}{T_{\text{unc}}}.$$

If $s > 1 - 1/\beta$, then $s + 1/\beta > 1$ and so $T_{\text{unc}} = o(D)$ by Step 1. Therefore

$$\min_t b^\star(t) \to \infty \quad \text{as} \quad D \to \infty.$$

For any fixed B, the constraint $b(t) \geq B$ is therefore inactive for all sufficiently large D, and the unconstrained solution is also the constrained optimum.

The ratio $C_1/C_0 = \sigma_1^2/\sigma_2^2 = r$ places this regime in the *terminal-tie case* in Lemma D.1, with $T_s = 0$ and tie interval $I_s = [0, T^\star]$: the entire horizon lies in the unconstrained branch, and $S \subseteq [0, T^\star]$ may be chosen as any measurable set satisfying $\int_S b^\star(t)\,\mathrm{d}t = \rho D$. This proves the noise-limited regime.

Step 3: the signal-limited regime and the strict-switching class. Now assume

$$\beta > 1, \qquad 0 < s < 1 - \frac{1}{\beta}.$$

Recall $\gamma := 2 - 1/\beta$, so that $K(u) = (u+1)^{-\gamma}$. By Step 1, the unconstrained optimizer satisfies

$$T_{\text{unc}} \asymp \left(\frac{sD}{4\beta\,\eta\,\kappa}\right)^{\frac{1}{s+1/\beta}}.$$

Since $s + 1/\beta < 1$, we have

$$\frac{T_{\text{unc}}}{D} \to \infty,$$

i.e., T_{unc} grows faster than linearly in D. On the other hand, every feasible schedule satisfies

$$D = \int_0^T b(t)\,\mathrm{d}t \geq BT,$$

so

$$T \leq T_{\max} := \frac{D}{B}.$$

Thus the unconstrained optimizer is infeasible, and the lower-bound constraint is active.

We first solve the problem asymptotically within the strict-switching canonical family. This does not assume that this family is globally optimal; it only computes the best comparison value among strict-switching schedules.

A strict-switching schedule has, in chronological order, up to four possible phases:

phase	1	2	3	4
(p, b)	$(0, B)$	$(0, >B)$	$(1, B)$	$(1, >B)$
duration	T_1	T_2	T_3	T_4

That is, T_i denotes the length of phase i, so that $T = T_1 + T_2 + T_3 + T_4$. The $p = 0$ flat (resp. unconstrained) phase has length T_1 (resp. T_2); the $p = 1$ flat (resp. unconstrained) phase has length T_3 (resp. T_4). Any of the T_i may be zero.

Step 3a: optimization of the $T_2 = 0$ strict-switching family. We first solve the strict-switching subfamily with $T_2 = 0$, so that only phases 1, 3, 4 are present. The $p = 0$ data constraint forces

$$T_1 = \frac{(1-\rho)D}{B} = (1-\rho)T_{\max}.$$

Parametrize phase 4 by terminal age $u := T - t \in [0, T_4]$. On phase 4 take $p(t) = 1$ and

$$b(T-u) = B\left(\frac{u+1}{T_4+1}\right)^{\delta-1}, \quad 0 \leq u \leq T_4.$$

Since $\delta < 1$, this satisfies $b(T-u) \geq B$, and it matches the lower bound at $u = T_4$.

The amount of $p = 1$ data consumed in phase 4 is

$$D_4 = \int_0^{T_4} B\left(\frac{u+1}{T_4+1}\right)^{\delta-1} \mathrm{d}u = \frac{B}{\delta}\left[(T_4+1) - (T_4+1)^{1-\delta}\right].$$

The remaining $p = 1$ data is carried by phase 3, so

$$\rho D = BT_3 + D_4,$$

and hence

$$T_3 = \frac{\rho D}{B} - \frac{(T_4+1) - (T_4+1)^{1-\delta}}{\delta}.$$

For the optimizer found below, $T_4 = o(D)$, so $T_3 > 0$ for all sufficiently large D.

The total horizon is $T = T_1 + T_3 + T_4$, and therefore

$$T = \frac{D}{B} - \left(\frac{1}{\delta} - 1\right) T_4 + O(T_4^{1-\delta}).$$

Consequently, in the regime $1 \ll T_4 \ll D$,

$$T^{-s} = B^s D^{-s} + \alpha_D T_4 + O(D^{-s-1} T_4^{1-\delta}) + O(D^{-s-2} T_4^2),$$

where

$$\alpha_D := s\left(\frac{1}{\delta} - 1\right) B^{s+1} D^{-s-1}.$$

We next compute the leading variance correction. Phase 3 corresponds to terminal ages $u \in [T_4, T_4 + T_3]$. Its contribution is

$$V_3(T_4) = \frac{\eta\,\sigma_1^2}{B} \int_{T_4}^{T_4+T_3} (u+1)^{-\gamma}\,\mathrm{d}u.$$

Since $T_3 = O(D)$ and $T_4 = o(D)$,

$$V_3(T_4) = \frac{\eta\,\sigma_1^2}{B(\gamma-1)} T_4^{-(\gamma-1)}(1 + o(1)).$$

Phase 4 contributes

$$V_4(T_4) = \int_0^{T_4} \frac{\eta\,\sigma_1^2 (u+1)^{-\gamma}}{B\left(\frac{u+1}{T_4+1}\right)^{\delta-1}}\,\mathrm{d}u.$$

Using

$$\gamma = 2 - \frac{1}{\beta} = 2 - 2\delta,$$

we obtain

$$V_4(T_4) = \frac{\eta\,\sigma_1^2}{B}(T_4+1)^{\delta-1} \int_0^{T_4} (u+1)^{\delta-1}\,\mathrm{d}u = \frac{\eta\,\sigma_1^2}{B\delta} T_4^{-(\gamma-1)}(1 + o(1)).$$

Thus

$$V_3(T_4) + V_4(T_4) = \xi\, T_4^{-(\gamma-1)}(1 + o(1)),$$

where

$$\xi := \frac{\eta\,\sigma_1^2}{B}\left(\frac{1}{\gamma-1} + \frac{1}{\delta}\right).$$

Hence the leading correction to $B^s D^{-s}$ in the $T_2 = 0$ strict-switching family is

$$\alpha_D T_4 + \xi T_4^{-(\gamma-1)}.$$

Minimizing this expression gives

$$\alpha_D = (\gamma-1)\,\xi T_4^{-\gamma}.$$

Since

$$(\gamma-1)\left(\frac{1}{\gamma-1} + \frac{1}{\delta}\right) = \frac{1}{\delta} - 1,$$

we obtain the optimal phase-4 length within the $T_2 = 0$ strict-switching family:

$$T_4 = \left(\frac{\eta\,\sigma_1^2}{sB^{s+2}}\right)^{1/\gamma} D^{\frac{s+1}{\gamma}}.$$

Since $s < \gamma - 1$, $T_4 = o(D)$, confirming the assumption $T_4 \ll D$ used above. This family achieves

$$\mathcal{E}^\star_{T_2=0}(D) = B^s D^{-s} + O\left(D^{-\frac{(s+1)(\gamma-1)}{\gamma}}\right).$$

In particular,

$$\mathcal{E}^\star_{T_2=0}(D) = B^s D^{-s} + o(D^{-s}).$$

Step 3b: elimination of T_2 and conclusion for the strict-switching class. We now show that no asymptotically optimal strict-switching schedule can have a nontrivial $p = 0$ unconstrained phase.

Let (T, b, p) be any strict-switching schedule. Since $b(t) \geq B$, every feasible schedule satisfies $T \leq T_{\max}$. Moreover, the variance term is nonnegative, and hence

$$\mathcal{E}[T, b, p] \geq T^{-s} \geq T_{\max}^{-s} = B^s D^{-s}.$$

On the other hand, Step 3a constructs a $T_2 = 0$ strict-switching schedule with value

$$B^s D^{-s} + o(D^{-s}).$$

Therefore, any strict-switching schedule that can be asymptotically optimal must satisfy

$$T = T_{\max}(1 + o(1)).$$

Indeed, if $T \le (1-c)T_{\max}$ along a subsequence for some $c > 0$, then

$$T^{-s} \ge (1-c)^{-s} B^s D^{-s},$$

whose leading constant is strictly larger than the value achieved in Step 3a.

Consequently,

$$T_{\max} - T = o(T_{\max}).$$

Using

$$D = \int_0^T b(t)\,\mathrm{d}t = BT + \int_0^T (b(t) - B)\,\mathrm{d}t,$$

we get

$$\frac{1}{B}\int_0^T (b(t) - B)\,\mathrm{d}t = T_{\max} - T = o(T_{\max}).$$

Thus the total excess data above the lower bound is $o(D)$.

Splitting $b = B + (b - B)$ on $\{p = 1\}$ and using that the total excess above the lower bound is $o(D)$, the $p = 1$ data constraint gives

$$\begin{aligned}\rho D &= \int_{\{p=1\}} b(t)\,\mathrm{d}t \\ &= B\int_{\{p=1\}} \mathrm{d}t + \int_{\{p=1\}} (b - B)\,\mathrm{d}t \\ &= B\int_{\{p=1\}} \mathrm{d}t + o(D),\end{aligned}$$

so the total time occupied by the $p = 1$ phases satisfies

$$\int_{\{p=1\}} \mathrm{d}t \;\ge\; \rho T_{\max} - o(T_{\max}).$$

Hence the $p = 0$ unconstrained phase, if present, lies at terminal ages

$$u \ge \rho T_{\max} - o(T_{\max}) = O(D).$$

On this phase,

$$K(u) = (u+1)^{-\gamma} \le C D^{-\gamma}$$

for some constant $C > 0$ independent of D.

Suppose the $p = 0$ unconstrained phase has positive excess data. Define

$$\Delta_2 := \frac{1}{B}\int_{T_1}^{T_1+T_2} (b(t) - B)\,\mathrm{d}t.$$

Replace this phase by a $p = 0$, $b = B$ segment carrying the same amount of $p = 0$ data. This preserves both data constraints and increases the total horizon by Δ_2.

The modification takes place only at terminal ages $O(D)$. Therefore the variance can increase by at most

$$C D^{-\gamma} \Delta_2.$$

On the other hand, since the original horizon satisfies $T = T_{\max}(1 + o(1))$, the increase of the horizon decreases the bias term by at least

$$c D^{-s-1} \Delta_2$$

for some constant $c > 0$ independent of D, and for all sufficiently large D. In the signal-limited regime, $s < 1 - 1/\beta$, and hence

$$s + 1 < \gamma.$$

Therefore

$$D^{-s-1} \gg D^{-\gamma}.$$

The bias improvement strictly dominates the possible variance loss. Thus any positive $p = 0$ unconstrained phase can be removed to improve the objective for all sufficiently large D. Hence an asymptotically optimal strict-switching schedule must satisfy

$$T_2 = 0.$$

Therefore the $T_2 = 0$ family optimized in Step 3a is asymptotically exhaustive within the strict-switching class. The corresponding phase durations and objective value are given by Step 3a; their explicit forms are recorded in Step 5 below.

Step 4: exclusion of terminal-tie schedules. It remains to show that terminal-tie schedules cannot improve on the strict-switching value.

Consider any feasible terminal-tie schedule. All $p = 1$ data lies in a terminal tie interval. Let L be the length of this interval and write $u = T - t \in [0, L]$. Whenever $p = 1$ in this interval,

$$b(T - u) = rC(u+1)^{\delta - 1}.$$

Since the tie interval is contained in the region where the $p = 1$ unconstrained branch is active,

$$rC(L+1)^{\delta-1} \geq B.$$

Therefore, on the $p = 1$ part of the tie interval,

$$\frac{b(T-u)}{B} \geq \left(\frac{u+1}{L+1}\right)^{\delta-1}.$$

Let

$$S := \{u \in [0, L] : p(T-u) = 1\}, \quad m := |S|.$$

The $p = 1$ data constraint gives

$$\rho T_{\max} = \int_S \frac{b(T-u)}{B}\,\mathrm{d}u.$$

Hence

$$\rho T_{\max} \geq \int_S \left(\frac{u+1}{L+1}\right)^{\delta-1} \mathrm{d}u.$$

The integrand $f(u) := \left(\frac{u+1}{L+1}\right)^{\delta-1}$ is decreasing in u. By the rearrangement inequality, for any measurable $S \subseteq [0, L]$ with $|S| = m$,

$$\begin{aligned}\int_S f(u)\,\mathrm{d}u &\geq \int_{L-m}^{L} f(u)\,\mathrm{d}u \\ &= \frac{L+1}{\delta}\left[1 - \left(1 - \frac{m}{L+1}\right)^{\delta}\right].\end{aligned}$$

(Since f is decreasing, replacing S by the rightmost subset of equal measure can only reduce the integral.) Combining with the previous inequality gives

$$\rho T_{\max} \geq \frac{L+1}{\delta}\left[1 - \left(1 - \frac{m}{L+1}\right)^{\delta}\right].$$

Solving for m,

$$m \leq (L+1)\left[1 - \left(1 - \frac{\delta\rho T_{\max}}{L+1}\right)^{1/\delta}\right].$$

We now bound this uniformly over $L \leq T \leq T_{\max}$. If

$$L + 1 \leq \delta\rho T_{\max},$$

then $m \leq L + 1 \leq \delta\rho T_{\max}$, which is already strictly smaller than $\rho T_{\max}$ by a linear amount. Otherwise the function

$$x \mapsto x\left[1 - \left(1 - \frac{\delta\rho T_{\max}}{x}\right)^{1/\delta}\right]$$

is increasing for $x > \delta\rho T_{\max}$. Indeed, with $z = \delta\rho T_{\max}/x$, it equals

$$\delta\rho T_{\max}\,\frac{1-(1-z)^{1/\delta}}{z},$$

and the last ratio is decreasing in z, because

$$z \mapsto 1 - (1-z)^{1/\delta}$$

is concave and vanishes at 0. Hence, since $L+1 \leq T_{\max} + 1$,

$$m \leq T_{\max}\Big[1 - (1-\delta\rho)^{1/\delta}\Big] + o(T_{\max}).$$

Define

$$c_{\rho,\delta} := \rho - \Big[1 - (1-\delta\rho)^{1/\delta}\Big].$$

Because $0 < \rho < 1$ and $0 < \delta < 1$,

$$(1-\delta\rho)^{1/\delta} > 1 - \rho.$$

Thus

$$c_{\rho,\delta} > 0,$$

and

$$m \leq \rho T_{\max} - c_{\rho,\delta} T_{\max} + o(T_{\max}).$$

The time spent on $p = 0$ is at most

$$(1-\rho)T_{\max},$$

because $b \geq B$. Hence any terminal-tie schedule satisfies

$$T \leq (1-\rho)T_{\max} + \big(\rho T_{\max} - c_{\rho,\delta} T_{\max} + o(T_{\max})\big),$$

and therefore

$$T \leq (1 - c_{\rho,\delta})T_{\max} + o(T_{\max}).$$

Its bias term is therefore bounded below by

$$T^{-s} \geq (1 - c_{\rho,\delta})^{-s} B^s D^{-s}(1 + o(1)).$$

Since $c_{\rho,\delta} > 0$, the leading prefactor of $B^s D^{-s}$ is strictly larger than 1. Therefore every terminal-tie schedule has objective strictly larger, at the leading constant level, than

$$B^s D^{-s} + o(D^{-s}),$$

which is achieved by the strict-switching family. Thus terminal-tie schedules are not asymptotically optimal.

Step 5: conclusion for the signal-limited regime. By Lemma D.1, every optimal schedule may be chosen in either the strict-switching case or the terminal-tie case. Step 4 excludes the terminal-tie case. Hence the global asymptotic optimum is given by the strict-switching optimum computed in Step 3.

Therefore

$$T_1^\star = \frac{(1-\rho)D}{B},$$

$$T_4^\star = \left(\frac{\eta\sigma_1^2}{sB^{s+2}}\right)^{1/\gamma} D^{\frac{s+1}{\gamma}}(1+o(1)),$$

and

$$T_3^\star = \frac{\rho D}{B} - \frac{(T_4^\star+1)-(T_4^\star+1)^{1-\delta}}{\delta} + o(T_4^\star).$$

Moreover,

$$T^\star = \frac{D}{B} - \left(\frac{1}{\delta}-1\right)T_4^\star + O\big((T_4^\star)^{1-\delta}\big),$$

or equivalently,

$$T^\star = \frac{D}{B} - (2\beta-1)T_4^\star + O\left((T_4^\star)^{1-\frac{1}{2\beta}}\right).$$

Finally, every feasible schedule satisfies

$$\mathcal{E}[T,b,p] \geq T^{-s} \geq B^s D^{-s},$$

because $T \leq D/B$. Together with the strict-switching upper bound, this gives

$$\mathcal{E}^\star(D) = B^s D^{-s} + O\left(D^{-\frac{(s+1)(\gamma-1)}{\gamma}}\right),$$

which, expanded in β, reads

$$\frac{(s+1)(\gamma-1)}{\gamma} = \frac{(s+1)(1-1/\beta)}{2-1/\beta}.$$

This is the claimed signal-limited regime expansion. The proof is complete. □

Remark D.2 (Critical regime)**.** *The proof deliberately excludes the critical case*

$$s = 1 - \frac{1}{\beta}.$$

At criticality, the subcritical balance gives $T_4 \asymp c\,D$ for some constant $c > 0$, so the terminal boundary layer is no longer sublinear. The expansion used to derive the signal-limited schedule therefore no longer applies.

The argument forcing the $p = 0$ unconstrained phase to vanish uses

$$s + 1 < 2 - \frac{1}{\beta}.$$

At criticality this becomes equality, so the bias loss from shortening the training horizon and the variance gain from extra batching are of the same order. Similarly, the argument excluding the terminal-tie form is no longer decisive, because different canonical forms may differ only at the level of leading constants.

Nevertheless, Lemma D.1 still applies. Thus, if an optimal solution exists, it may be chosen in one of the canonical forms described there. Determining which canonical form is selected, and which phase proportions vanish, requires solving the corresponding finite-dimensional constant-level problem.

E Experiments Details

E.1 Experimental details for Theorem 3.1 verification

We verify Theorem 3.1 by running one-pass SGD in the diagonal feature space, where the assumptions hold exactly. We test both the noise-limited regime and the signal-limited regime with four schedules satisfying the same total high-quality data budget $\sum_{k=1}^K p_k B_k = \rho \sum_{k=1}^K B_k$, where $\rho \in (0,1)$ is the fraction of high-quality data:

- **Early / Late / Middle**: bang-bang quality placement ($p = 1$ on the first, last, or central ρK steps) with constant batch.
- **Sinusoidal**: jointly varying quality $p(t) \propto \rho + 0.25\sin(8\pi t/T)$ and oscillating-upward batch $b(t) \propto 1 + 3t/T + 0.8\sin(12\pi t/T)$, both rescaled to satisfy the budget constraints.

Figure 5 shows the results. In each panel, the measured excess risk (colored, averaged over 10 seeds) closely tracks the scaled quality-aware FSL prediction (black) throughout training. Inset plots display the schedules $p(t)$ and $b(t)$. The theory accurately tracks all schedule-dependent differences—including the complex Sinusoidal case where both controls vary simultaneously—confirming that the forgetting kernel $K(t-\tau)$ correctly weights $\sigma_{\text{eff}}^2(\tau)/b(\tau)$ at each τ.

Parameter settings. Throughout, we fix dimension $N = 500$, capacity exponent $\beta = 2$, learning rate $\eta = 0.01$, and high-quality data fraction $\rho = 0.3$, with critical exponent $s_{\text{crit}} = 1 - 1/\beta = 0.5$.

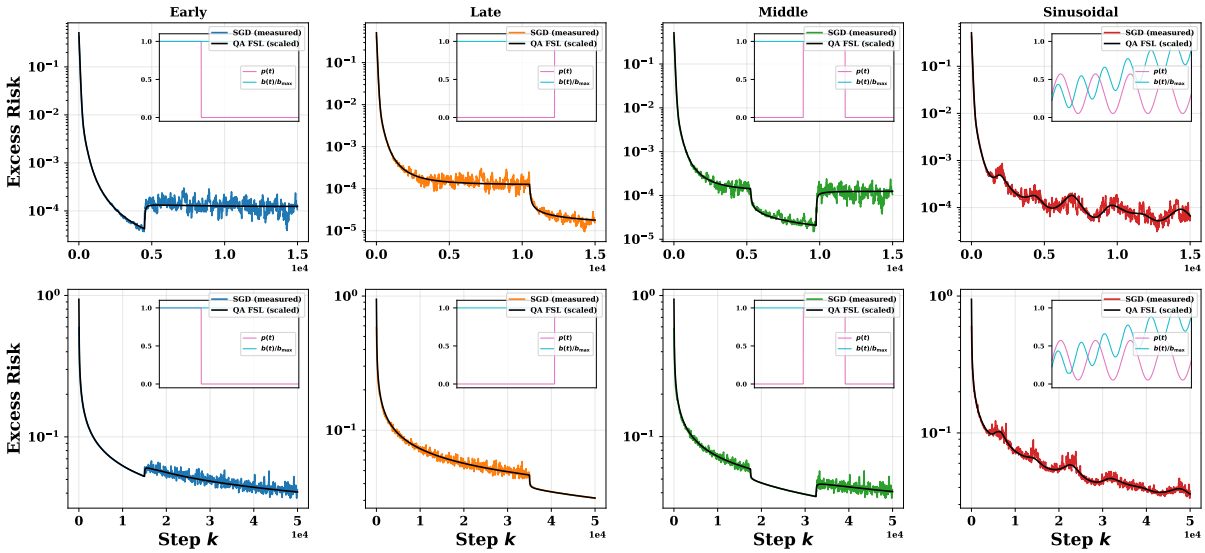

Figure 5: Verification of Theorem 3.1. Top: noise-limited ($s = 2$). Bottom: signal-limited ($s = 0.4$). Colored curves: measured SGD risk (averaged over 10 seeds). Black curves: quality-aware FSL theory (scaled).

Noise-limited setting ($s = 2$): total iterations $K = 15{,}000$ ($T = 150$), reference batch size $B = 32$, noise variances $\sigma^2_{\text{good}} = 0.1$, $\sigma^2_{\text{bad}} = 1.0$.

Signal-limited setting ($s = 0.4$): total iterations $K = 50{,}000$ ($T = 500$), reference batch size $B = 4$, noise variances $\sigma^2_{\text{good}} = 0.1$, $\sigma^2_{\text{bad}} = 10.0$. In the signal-limited regime, the bias term t^{-s} decays slowly and dominates unless the noise strength is sufficiently large. We therefore use a smaller batch size and a larger bad-noise variance to ensure that the noise integral is comparable to the bias T^{-s}, so that the effect of different quality schedules is visible.

Schedule construction. The Early, Late, and Middle schedules use a constant batch size $b(t) = B$ and bang-bang quality schedules that place $p(t) = 1$ on the first, last, or central ρK steps respectively.

The Sinusoidal schedule uses a joint non-constant design:

$$b_{\text{raw}}(t) = 1 + 3\frac{t}{T} + 0.8\sin\Big(\frac{2\pi \cdot 6\,t}{T}\Big),$$
$$p_{\text{raw}}(t) = \rho + 0.25\sin\Big(\frac{2\pi \cdot 4\,t}{T}\Big).$$

The batch schedule is rescaled so that $\sum_{k=1}^{K} B_k = BK$ (matching the total data budget of the constant-batch schedules), and the quality schedule is rescaled so that $\sum_{k=1}^{K} p_k B_k = \rho \sum_{k=1}^{K} B_k$, with values clipped to $[0, 1]$. This produces an oscillating-upward batch curve (6 cycles) and a sinusoidal quality curve (4 cycles) that jointly satisfy the budget constraints.

Theoretical prediction curve. For the theoretical prediction, we compute the kernel $K(t) = \sum_{j=1}^{N} \lambda_j^2 e^{-2\lambda_j t}$ and the bias term $e(t) = \sum_{j=1}^{N} \lambda_j |\theta_j^\star|^2 e^{-2\lambda_j t}$ exactly in the finite-dimensional setting (rather than using the asymptotic power-law approximation). The noise integral $\int_0^t K(t-\tau)\sigma^2_{\text{eff}}(\tau)/b(\tau)\,\mathrm{d}\tau$ is discretized with step size η. Since $b(\tau)$ and $p(\tau)$ now vary across schedules, the integrand $\sigma^2_{\text{eff}}(\tau)/b(\tau)$ is evaluated pointwise at each discrete time step.

Fitting procedure. Since the theorem establishes an asymptotic equivalence ($\asymp$) rather than an exact equality, we fit a single global multiplicative constant (via the median of measured/predicted ratios) to align the theory curve with the measurement. All experimental curves are averaged over 10 independent random seeds.

E.2 Experimental details for Corollary 4.2 verification

Constant-batch experiment (Corollary 4.2(i)). We use the same settings as in Appendix E.1: noise-limited ($s = 2$, $B = 32$, $\sigma^2_{\text{bad}} = 1$, $K = 15{,}000$) and signal-limited ($s = 0.4$, $B = 4$, $\sigma^2_{\text{bad}} = 10$, $K = 50{,}000$), with $N = 500$, $\beta = 2$, $\eta = 0.01$, $\sigma^2_{\text{good}} = 0.1$, and $\rho = 0.3$ throughout. All results are averaged over 10 seeds.

When $b(t)$ is constant, the score reduces to $S_b(t) \propto K(T - t)$, which is monotonically increasing in t: noise injected late is remembered

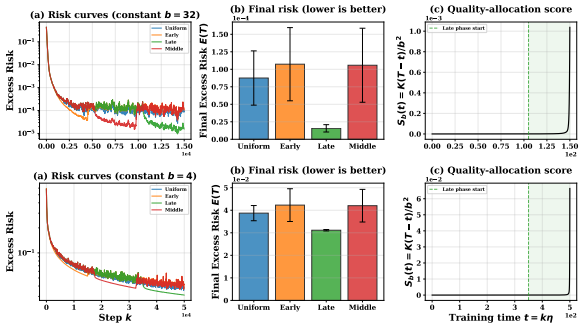

Figure 6: Corollary 4.2(i): constant batch. Top: noise-limited ($s = 2$); bottom: signal-limited ($s = 0.4$). (a) Risk curves: Late achieves the lowest final risk. (b) Final excess risk bar chart. (c) Score $S_b(t) \propto K(T-t)$ is monotonically increasing; shaded region marks where Late places high-quality data.

most strongly, so the optimal quality schedule concentrates high-quality data at the end of training. Figure 6 confirms the prediction in both the noise-limited and signal-limited regimes. The Late schedule achieves the lowest final risk, with a consistent ranking Late $<$ Uniform $<$ Middle $<$ Early. Panel (c) shows that $S_b(t)$ is monotonically increasing, and the Late schedule places its high-quality data precisely in the region where $S_b(t)$ is largest.

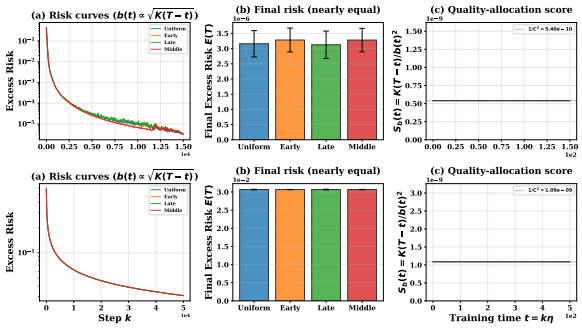

Figure 7: Corollary 4.2(ii): batch $b(t) = C\sqrt{K(T-t)}$. Top: noise-limited ($s = 2$); bottom: signal-limited ($s = 0.4$). (a) Risk curves for the four schedules are nearly identical. (b) Final excess risks are equal within standard error. (c) Score $S_b(t) \approx 1/C^2$ is approximately constant, confirming degeneracy.

Optimal-batch experiment (Corollary 4.2(ii)). We use the same eigenstructure, learning rate, and quality parameters as above, with total data budget $D = 200{,}000$. For the noise-limited setting ($s = 2$, $\sigma^2_{\text{bad}} = 1$, $K = 15{,}000$) and the signal-limited setting ($s = 0.4$, $\sigma^2_{\text{bad}} = 10$, $K = 50{,}000$), we replace the constant batch with the schedule $B_k = \max\{C\sqrt{K(T-k\eta)}, 1\}$, where $C = D/\int_0^T \sqrt{K(T-t)}\,\mathrm{d}t$.

Because the batch size now varies across time, a fixed number of high-quality *steps* would no longer guarantee equal high-quality *data* budgets across schedules. We therefore construct budget-aware quality schedules: for the Early schedule, we set $p_k = 1$ on the first n_{early} steps such that $\sum_{k=1}^{n_{\text{early}}} B_k\eta = \rho D$; for the Late schedule, we set $p_k = 1$ on the last n_{late} steps satisfying the same budget; and similarly for Middle. This ensures that all schedules consume exactly the same total number of high-quality samples, so any performance difference can be attributed solely to the *timing* of quality placement.

When $b(t) = C\sqrt{K(T-t)}$ (the rate-optimal schedule of Wang et al. (2026)), the score becomes $S_b(t) = 1/C^2$, a constant independent of t, so quality placement is degenerate. Figure 7 confirms this: unlike the constant-batch case, the four quality schedules produce nearly indistinguishable risk curves in both the noise-limited and signal-limited settings, with final excess risks equal within standard error. Panel (c) shows that $S_b(t)$ is approximately flat, confirming degeneracy.

E.3 Experimental details for Theorem 4.3 verification

Parameter settings. We fix dimension $N = 500$, high-quality fraction $\rho = 0.3$, and $\sigma^2_{\text{good}}/\sigma^2_{\text{bad}} < 1$ throughout. All results are averaged over 10 independent random seeds.

Noise-limited regime ($s = 2$, $\beta = 2$): learning rate $\eta = 0.01$, noise variances $\sigma^2_{\text{good}} = 0.1$, $\sigma^2_{\text{bad}} = 1.0$ (variance ratio $r = 0.1$), total data budget $D = 200{,}000$.

Signal-limited regime ($s = 0.5$, $\beta = 3$): learning rate $\eta = 0.05$, noise variances $\sigma^2_{\text{good}} = 0.25$, $\sigma^2_{\text{bad}} = 1.0$ (variance ratio $r = 0.25$), total data budget $D = 10{,}000$, minimum batch size $B_{\min} = 4$.

Strategy construction. All four strategies share the same total data budget D and high-quality data budget ρD.

1. **Constant b + uniform p:** $b(t) = D/T$, $p(t) = \rho$. In the noise-limited regime, the training horizon is set to the joint optimal $T^\star$ from Strategy 4. In the signal-limited regime, the constant batch is $b = B_{\min}$, which forces $T = D/B_{\min}$ (the maximum feasible horizon).
2. **Constant b + late p:** same batch size and horizon as Strategy 1; quality is bang-bang

with $p = 1$ on the last $\rho T/\eta$ steps and $p = 0$ otherwise, satisfying the high-quality budget constraint.

3. **Uniform p + optimal b**: $p(t) = \rho$ throughout; batch size $b(t) = \max(C\sqrt{K(T-t)}, B_{\min})$ with the constant C determined by bisection so that $\int_0^T b(t)\,\mathrm{d}t = D$. The training horizon $T^\star$ is independently optimized by minimizing the FSL objective $T^{-s} + \eta \int_0^T K(T-t)\sigma_{\text{eff}}^2/b(t)\,\mathrm{d}t$ over T using bounded scalar minimization.
4. **Joint optimal**: implements Theorem 4.3 directly. In the noise-limited regime (Part I), the training horizon $T^\star$ is optimized over T; the batch schedule is $b(t) = C\sqrt{K(T^\star - t)}$ on low-quality steps and $b(t) = rC\sqrt{K(T^\star - t)}$ on high-quality steps, with quality placement chosen as late (one valid choice among the degenerate family). In the signal-limited regime (Part II), the three-phase structure is constructed: Phase 1 ($p = 0$, $b = B_{\min}$) exhausts all low-quality data over $T_1^\star = (1-\rho)D/B_{\min}$; Phase 3 ($p = 1$, ramp-up) uses $b(t) = B_{\min}\big((T^\star - t + 1)/(T_4^\star + 1)\big)^{1/(2\beta)-1}$; Phase 2 ($p = 1$, $b = B_{\min}$) occupies the remainder $T_3^\star = T^\star - T_1^\star - T_4^\star$. The ramp-up duration $T_4^\star$ is optimized numerically by minimizing the quality-aware FSL objective over T_4, which implicitly determines the total horizon $T^\star = T_1^\star + T_3^\star + T_4^\star$.

F LLM Experiment Details

F.1 Model architecture and training infrastructure

We use a Mixture-of-Experts (MoE) model with 15B total parameters and 2.4B active parameters, following the shortcut-connected MoE architecture of Cai et al. (2025). The model has 12 transformer layers with a hidden dimension of 2304, 24 attention heads with 8-way grouped-query attention, and 112 experts per MoE layer with top-8 routing. The sequence length is 8192. All experiments are initialized from the same checkpoint pretrained on 1.2T tokens of general web data.

Training uses BF16 mixed precision with the AdamW optimizer (Loshchilov and Hutter, 2019) (β_1=0.9, β_2=0.95, ϵ=10^{-12}, weight decay 0.1). We use the standard Megatron-style hybrid parallelism (tensor parallelism $\times$ pipeline parallelism). The peak learning rate is η=6.51×10^{-4} with a minimum of 4.2×10^{-5}, configured per-schedule (see below). The total token budget for all midtraining experiments is 108B tokens.

F.2 Data composition

Pretraining corpus. The pretraining corpus consists of a large-scale proprietary dataset serving as general pretraining data. It covers a broad and diverse distribution of topics, domains, and languages to establish the model's foundational capabilities. Similar to standard web corpora, this general data is diverse but inherently noisy with respect to specific downstream task distributions, corresponding to high σ_{bad}^2 in our theoretical framework. All midtraining experiments fork from a single checkpoint pretrained on this corpus at 1.2T tokens.

Midtraining corpus. The midtraining corpus is specifically curated for reasoning and coding enhancement, designed to align closely with complex downstream evaluation benchmarks. It primarily consists of:

- **Reasoning enhancement data**: high-quality question-answering pairs, chain-of-thought annotations, and step-by-step mathematical proof trajectories to bolster logical deduction and problem-solving capabilities.
- **Coding enhancement data**: carefully filtered proprietary code datasets and high-quality repositories with documentation, covering algorithmic problem-solving across multiple programming languages.

This focused corpus exhibits significantly lower noise variance σ_{good}^2 with respect to downstream reasoning and coding benchmarks, as it is tightly aligned in both format and distribution.

F.3 Batch-size schedule configurations

All schedules below use the same 108B token budget, sequence length 8192, peak learning rate η=6.51×10^{-4}, and minimum learning rate 4.2×10^{-5}. Steps are reported in units of 1k.

Batch-size schedule comparison (the first two sets of Table 1). Three schedules are compared under identical token budgets:

- **Constant 4k**: global batch size held at 4096 throughout the 108B midtraining span ($\approx$ 3.2k steps).
- **Ramp-up 4k$\rightarrow$8k**: batch size steps through $\{4096, 5120, 6144, 7168\}$ over the first 86B tokens (Megatron's equal-sample ramp with increment 1024), then stays at 8192 for the remaining 22B ($\approx$ 2.3k steps).

- **Drop 1k→8k**: batch size drops to 1024 at the start of midtraining, steps through $\{1024, 2048, \ldots, 7168\}$ over 94B tokens, then stays at 8192 for the remaining 14B ($\approx$ 4.3k steps).

All three schedules consume the same total tokens. Drop spends more gradient updates in the early phase (when the model first sees the midtraining distribution shift), consistent with our theoretical prediction that smaller batch sizes at the quality transition preserve more gradient signal.

Stable-phase ratio experiment (Figure 4). To finely sweep the duration of the low-batch stable phase, we instantiate the DSR recipe with batch-size bounds $[512, 4096]$ and vary the stable-phase ratio:

- **Drop**: at the start of midtraining, the batch size drops from the pretraining value of 4096 to 512.
- **Stable**: batch size held at 512 with learning rate fixed at η.
- **Rampup**: batch size steps through $\{512, 1024, \ldots, 3584\}$ in equal-sample increments of 512 with learning rate fixed at η, then held at 4096 with learning rate linearly decayed from η to 4.2×10^{-5} (equivalent to continuing the batch-size ramp-up by Smith et al. (2018)).

We fix the ramp-up and decay phases to receive equal token shares of the midtraining budget after the stable phase, and sweep the stable-phase ratio from 0% to 62.14% (corresponding to stable-phase entry points of $\{0, 3200, 6400, 9600, 12800, 16000\}$ steps in the parent run).

Schedule comparison (the last set of Table 1). Three schedules are compared:

- **Cosine-decay**: Constant batch size 4096, learning rate follows a cosine decay schedule from 6.51×10^{-4} to 4.2×10^{-5}.
- **WSD** (Hu et al., 2024): Constant batch size 4096, learning rate held at 6.51×10^{-4} for the first 72B tokens, then linearly decayed to 4.2×10^{-5} over the final 36B tokens.
- **Drop-Stable-Rampup**: Our proposed schedule with batch size [512, 4096], 24.9% stable ratio, followed by linear ramp-up and learning rate decay.

All three schedules consume the same 108B token budget.

Training infrastructure and MFU accounting. All experiments are conducted on 384 accelerators using the Megatron framework (Shoeybi et al., 2019), configured with pipeline parallelism $pp = 6$ and tensor parallelism $tp = 1$. Relative to the constant-batch baseline at batch size 4096, batch size 512 achieves approximately 0.78× MFU, reflecting lower hardware saturation at small batches. Accounting for the measured MFU at each batch-size bucket and the corresponding token allocation in the schedule, **Drop-Stable-Rampup** incurs a 1.1× wall-clock/device-time overhead compared to constant-batch baselines such as Cosine-decay and WSD, which operate at batch size 4096 throughout. Since all schedules consume the same 108B token budget, this overhead reflects reduced hardware utilization rather than additional algorithmic FLOPs.

F.4 Evaluation benchmarks

We evaluate on 14 benchmarks spanning four categories:

- **Knowledge**: MMLU (Hendrycks et al., 2021a), MMLU-Pro (Wang et al., 2024), CMMLU (Li et al., 2024), C-Eval (Huang et al., 2023).
- **Reasoning**: ARC-Challenge (Clark et al., 2018), BIG-Bench Hard (BBH) (Suzgun et al., 2023).
- **Math**: GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021b).
- **Code**: HumanEval+ (HE+) (Liu et al., 2023), BigCodeBench (BCB) (Zhuo et al., 2025), LiveCodeBench (LCB) (Jain et al., 2025), CRUX-I, CRUX-O (Gu et al., 2024), MultiPL-E (Cassano et al., 2023).

All benchmarks are evaluated in zero-shot or few-shot settings following standard protocols. For MMLU, CMMLU, and C-Eval, we report 5-shot accuracy. For GSM8K and MATH, we use 8-shot chain-of-thought prompting. For code benchmarks, we use pass@1 with greedy decoding.

G Heterogeneous training dynamics across benchmarks

Our theory predicts that the optimal schedule depends on whether training is signal-limited or noise-limited, determined by the exponents s and β in the linear regression model. In practice, we do not have direct access to these parameters for LLM training, nor do we know a priori whether different

tasks exhibit uniform or divergent dynamics. In this section, we analyze the accuracy trajectories of 14 benchmarks and report three findings.

First, different benchmarks exhibit clearly divergent dynamics during the stable phase. We define the *early-phase gap* as the average accuracy difference between WSD and Drop-Stable-Rampup over all checkpoints within the stable phase (Figure 9). Since the stable phase uses a much smaller batch size (512 vs. 4096), all benchmarks suffer from increased gradient noise and DSR generally trails WSD during this phase. However, the magnitude of this deficit varies substantially across benchmarks: some show a large early-phase gap (DSR falls far behind WSD), while others show a negligible gap (DSR remains close to WSD despite the small batch size).

Second, consistent with our theoretical prediction, benchmarks with smaller early-phase gaps achieve larger final gains from Drop-Stable-Rampup. A small early gap indicates that the increased noise from reducing the batch size is largely offset by the signal accumulated through additional gradient steps—this signal remains latent during the stable phase but manifests after the ramp-up. Conversely, a large early gap indicates that the noise cost of small batch size dominates, and the ramp-up phase can only partially recover the deficit.

Third, from the perspective of task content, benchmarks with small early-phase gaps correspond to mathematical reasoning (GSM8K, MATH) and algorithmic code generation (HumanEval+, MultiPL-E)—capabilities that require compositional, multi-step inference. A notable exception is BigCodeBench: despite being a code benchmark, it exhibits a large early-phase gap. BigCodeBench emphasizes diverse library/function calls and complex instructions; in our trajectories, its behavior suggests that the tested capabilities may be more sensitive to API/library knowledge than to the compositional signal accumulation observed in HumanEval+ and MultiPL-E.

Benchmarks with small early-phase gaps. These benchmarks show that DSR remains close to WSD during the stable phase despite using a much smaller batch size (Figure 9). The noise penalty of small batch size is largely absorbed, and DSR overtakes WSD during the ramp-up/decay stage, ultimately achieving the largest final gains. This is consistent with the interpretation that extra gradient steps at low batch size accumulate latent capacity that only manifests once noise is suppressed during ramp-up.

- **GSM8K** (+4.23): Tests multi-step arithmetic reasoning requiring chaining 2–8 sequential operations. The training curve shows near-zero early gap, with a sharp separation emerging only during the ramp-up/decay phase.
- **MATH** (+2.80): Probes competition-level mathematical problem-solving across prealgebra, algebra, number theory, counting and probability, geometry, intermediate algebra, and precalculus. The curve shows minimal early gap, with gains accumulating gradually during ramp-up.
- **HumanEval+** (+2.44): Evaluates Python code generation with rigorous test cases. Virtually no early-phase gap; the performance separation emerges only in the later ramp-up/decay stage.
- **MultiPL-E** (+2.66): Assesses code generation across multiple programming languages. Small early gap combined with large final gain.

Benchmarks with large early-phase gaps. These benchmarks show that DSR falls substantially behind WSD during the stable phase. The ramp-up phase partially closes this deficit, but the final gains are smaller than the small-gap group, and in one case DSR underperforms a baseline.

- **CMMLU** (+0.98): The Chinese counterpart of MMLU, it shows the most pronounced early-phase gap among all benchmarks—DSR trails WSD by a large margin throughout the stable phase, and only partially recovers during ramp-up.
- **BigCodeBench (BCB)** (+2.81 over WSD, but -1.40 vs. Cosine-decay): Despite being a code benchmark, BCB exhibits a very large early-phase gap. BigCodeBench emphasizes practical code generation with diverse library/function calls and complex instructions. In our trajectories, its large early gap suggests that the tested capabilities may be more sensitive to API/library knowledge and function-call conventions than to the delayed signal accumulation observed in algorithmic code-generation benchmarks such as HumanEval+ and MultiPL-E. Notably, BCB is the only benchmark where Drop-Stable-Rampup underperforms Cosine-decay, suggesting that for

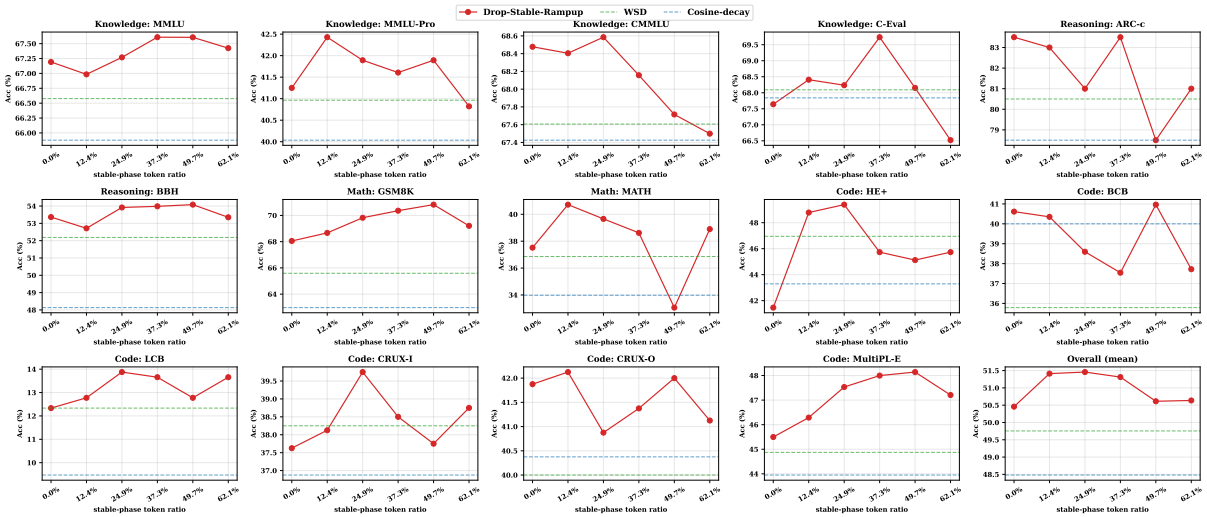

Figure 8: Final checkpoint evaluation results across different schedule configurations, demonstrating the robustness of the Drop-Stable-Rampup strategy.

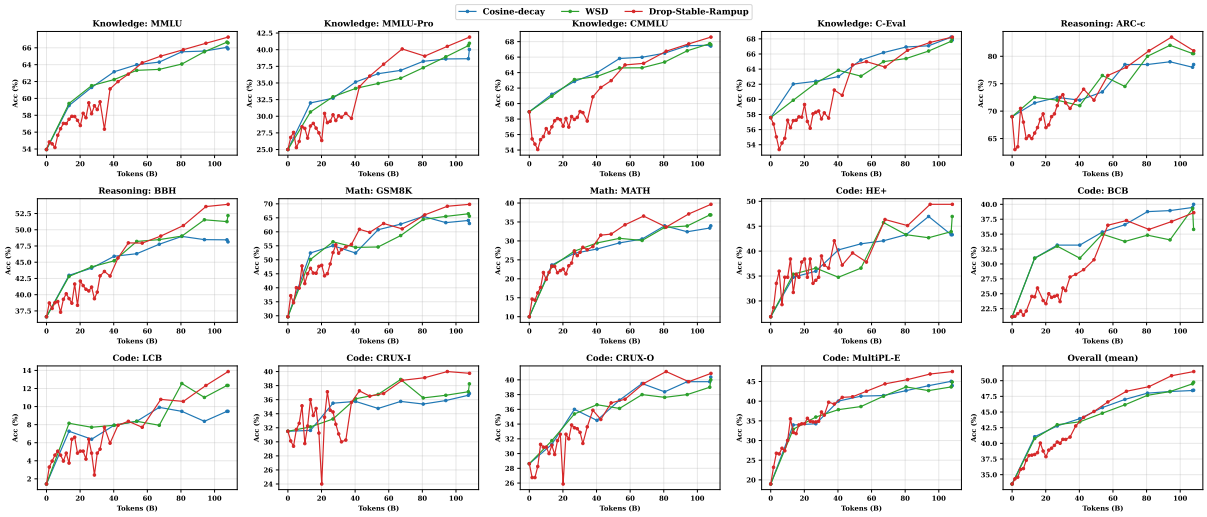

Figure 9: Per-benchmark accuracy curves for Cosine-decay, WSD, and Drop-Stable-Rampup across all 14 evaluation benchmarks.

tasks with this trajectory pattern, the extended low-batch-size phase is unnecessary and a smoother schedule may be preferable.

Benchmarks with moderate early-phase gaps. Several benchmarks fall between the two groups, exhibiting a moderate deficit during the stable phase that is largely recovered during ramp-up.

- **MMLU** (+0.69): Measures broad factual knowledge across 57 subjects. A moderate early-phase gap, with the final gain being modest but positive.
- **MMLU-Pro** (+0.93): An extended version of MMLU with harder distractors and 10-way multiple choice. Similar moderate early gap and final gain.
- **BBH** (+1.73): Tests diverse compositional reasoning (logical deduction, causal reasoning, disambiguation). The moderate gap likely reflects a mixture of subtask types with different noise sensitivities.

Summary. The key empirical finding is a negative correlation between the early-phase gap (how much DSR trails WSD during the stable phase) and the final gain from Drop-Stable-Rampup. Benchmarks where the small batch size causes little performance deficit during the stable phase (GSM8K, MATH, HumanEval+, MultiPL-E) ultimately achieve the largest final improvements (+2.44 to +4.23). These correspond to compositional reasoning and algorithmic code generation.

Benchmarks where the small batch size causes a large deficit (CMMLU, BCB) achieve smaller or negative final gains—the noise cost of the extended stable phase is not fully compensated by signal accumulation. This pattern is consistent with our theoretical prediction: tasks where signal accumulation is the bottleneck tolerate the noise of small batch size and benefit from extra gradient steps; tasks where noise dominates are harmed by the extended low-batch-size phase.

Implications for future work. The divergence in dynamics across tasks suggests a natural direction: *task-aware data scheduling*. If the early-phase gap can be estimated from a small pilot run, one could tailor the data mixture to the schedule phase—prioritizing compositional data (math, code, chain-of-thought) during the stable phase to maximize signal accumulation, and shifting toward distributional content (world knowledge, encyclopedic text) during ramp-up. We leave the formal optimization of such per-domain scheduling as future work.