ANYTIME PRETRAINING: HORIZON-FREE LEARNING-RATE SCHEDULES WITH WEIGHT AVERAGING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models are increasingly trained in continual or open-ended settings, where the total training horizon is not known in advance. Despite this, most existing pretraining recipes are not anytime: they rely on horizon-dependent learning rate schedules and extensive tuning under a fixed compute budget. In this work, we provide a theoretical analysis demonstrating the existence of anytime learning schedules for overparameterized linear regression, and we highlight the central role of weight averaging—also known as model merging—in achieving the minimax convergence rates of stochastic gradient descent. We show that these anytime schedules polynomially decay with time, with the decay rate determined by the source and capacity conditions of the problem. Empirically, we evaluate 150M and 300M parameter language models trained for up to $32\times$ Chinchilla scale, comparing constant and $1/\sqrt{t}$ schedules with weight averaging against a well-tuned cosine schedule. Across the full training range, the anytime schedules achieve comparable final loss to cosine decay. Taken together, our results suggest that weight averaging combined with simple, horizon-free step sizes offers a practical and effective anytime alternative to cosine learning rate schedules for large language model pretraining.

1 INTRODUCTION

Since its introduction in the seminal paper by Loshchilov & Hutter (2016), cosine decay has become the standard learning rate scheduler for large language model (LLM) pretraining, being used by practitioners for training frontier models across a wide range of scales and architectures (Dubey et al., 2024; Olmo et al., 2025; Liu et al., 2024). Despite its widespread adoption, cosine decay is a horizon dependent scheduler, since it requires knowing the length of the training run ahead of time, making it ill suited for continual learning setups in which data is continuously received and mixed into the pretraining run.

Recent work (Hu et al., 2024; Wen et al., 2024) has proposed the warmup-stable-decay (WSD) schedule as a nearly-anytime alternative to cosine decay, achieving competitive performance (Team et al., 2025). WSD consists of 3 phases, namely a linear learning rate warm-up phase up to a maximum learning rate, then keeping the learning rate constant until 90% of the training length, followed by decaying the learning rate for the rest of the 10%. WSD is not a strictly anytime schedule, since it requires additional overhead in the form of storing checkpoints of the training run at sparse intervals, from which one can either extend the constant learning rate phase in case of new data, or decay in order to get the final performance. In the case of continued training after decaying, the FLOPs spent on the decay part of the scheduler need to be charged again, introducing additional cost. From a theory point of view, polynomial learning rate schedules are common in the stochastic optimization literature (Shamir & Zhang, 2013; Dieuleveut & Bach, 2014; Wu et al., 2022a), and carefully designed horizon-aware decay sequences exist for least squares (Ge et al., 2019; Jain et al., 2019). However, unless weight averaging is used, these schedules generally do not achieve optimal rates for SGD in linear regression (Ge et al., 2019; Wu et al., 2022a). In this work we investigate theoretically in infinite dimensional Gaussian linear regression, and empirically by training LLMs for up to $32\times$ Chinchilla tokens, how $1/t^\gamma$ with iterate averaging, constant learning rate with iterate averaging and WSD compare to cosine decay in long training runs, and which of these schedules

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

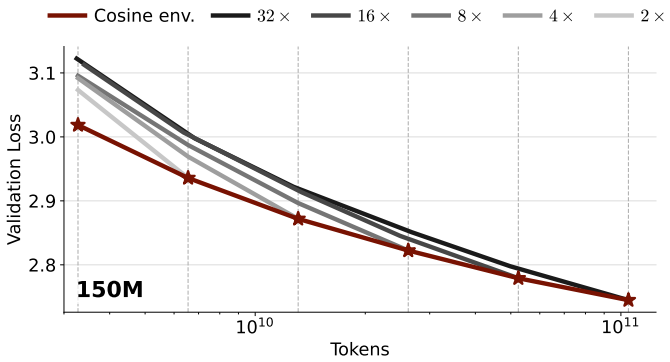


Figure 1: 150M model: **Cosine schedules do not transfer across horizons.** The *cosine envelope* (red) is formed by independently tuning a horizon-aware cosine schedule for each terminal compute budget (1×–32× Chinchilla) and taking the best validation loss at that horizon. Gray curves show the same cosine schedule evaluated at intermediate checkpoints when tuned for a single fixed terminal budget. The gap illustrates why cosine decay is not anytime: tuning for a long horizon can be far from optimal at shorter budgets. An analogous plot for the 300M model appears in Figure 10b (Appendix C).

provides a viable anytime alternative to cosine annealing. Concretely, we require two properties from an anytime scheduler: (i) the scheduler should not depend on token horizon, and (ii) for any intermediate duration T, it should be competitive with a well-tuned cosine schedule run for T steps, i.e. it should track the *cosine envelope* that these tuned cosine schedules define across checkpoints in a long run. Prior work has compared schedules across multiple training durations: Hägele et al. (2024) tune cosine baselines separately per duration and compare cooldown runs against them, and Hu et al. (2024) report similar multi-horizon comparisons. In our evaluation, we ask whether a *single fixed horizon-free trajectory* can track the envelope of per-horizon-tuned cosine schedules at almost every intermediate checkpoint of a run up to 32× the Chinchilla-optimal budget.

To address this question, we turn to the simplest theoretical proxy for pretraining, the quadratic model. Concretely, the bias-variance decomposition of the risk in quadratics (Jain et al., 2018; Zou et al., 2023) suggests that weight averaging combined with polynomial learning rate decay can be horizon-free under certain spectral assumptions, while still achieving optimal rates. The same analysis predicts that when the variance contribution is small, e.g. at large batch size, maintaining a larger learning rate decreases the bias term faster, and decay becomes unnecessary. A central goal of this paper is to test whether these prescriptions survive contact with LLM pretraining at scale.

Contributions. In Section 2, we begin by deriving an anytime schedule in linear regression on quadratics, and specializing this theorem to power law covariance. Zhang et al. (2024b) (Corollary 2) have shown that for linear regression on power law data with capacity exponent a and source exponent b, a constant learning rate with tail averaging is anytime for $b \leq a$ (and in particular it is optimal for $b = a$). In this work, we extend this regime by showing that a polynomial learning rate scheduler of the form $1/t^\gamma$ for γ depending on a and b is anytime for $a < b \leq 2a$, and achieves the minimax optimal exponent for power law regression (Zhang et al., 2024a), up to logarithmic factors and constants. Zhang et al. (2024b) also provide a tight risk analysis in this regime for a constant learning rate with averaging. However, note that the constant rate is horizon dependent, as it scales as $N^{a/b-1}$. Our results are compared to existing literature in Table 2.

Following our theoretical analysis, we stress test the prescriptions of the quadratic model in Section 3. We establish a baseline *cosine envelope* by training 150M up to 32× Chinchilla and 300M LLMs up to 16× Chinchilla tokens using cosine decay tuned for each power-of-2 horizon in this interval. We then test several candidates for anytime schedules: a constant learning rate with iterate EMA and a polynomially decaying $1/\sqrt{t}$ with iterate EMA, as well as the nearly-anytime scheduler WSD, and seeing if they can closely track the cosine envelope. Figure 1 shows why this is a hard baseline: a cosine tuned for a long horizon is substantially suboptimal at intermediate regimes. For the anytime candidates, we provide 2 evaluations: in Figure 2 we show that amongst all sweep configurations,

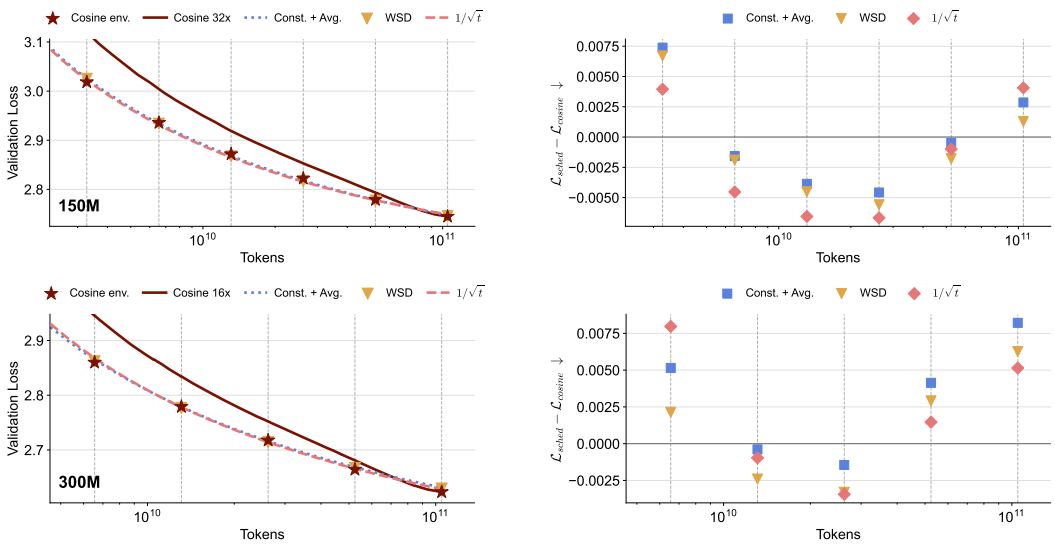


Figure 2: Anytime schedules vs. cosine decay; 150M (top) and 300M (bottom). **Left:** Validation loss vs. training compute at $1\times$–$32\times$ Chinchilla ($1\times$–$16\times$ for 300M). Cosine baselines are *separate* runs, each tuned for its own duration; red stars mark the resulting per-duration optimal cosine envelope, and the red curve is the single cosine run tuned for the full horizon ($32\times$ and $16\times$, respectively). Constant LR with averaging and the $1/\sqrt{t}$ schedule are each a *single* run trained to the full horizon and evaluated at intermediate checkpoints; the $1/\sqrt{t}$ schedule uses the multiplicative factor $\sqrt{\alpha/(t+\alpha)}$ with tuned α. WSD branches from checkpoints of the constant-LR run at 90% of each target budget and applies a linear decay over the final 10% (one decay branch per budget, not a single trajectory). **Right:** Loss difference relative to cosine at each compute multiple (negative = better than cosine). Hyperparameters of the single run are chosen to closely track the cosine envelope and are reported in Table 1, with prospective transfer results shown in Figure 4.

there exists a *single run* that can closely track the cosine envelope. Next, in Figure 4 we show that one can choose these runs prescriptively: following a burn-in, the optimal configuration at $4\times$ Chinchilla across both models closely tracks the cosine envelope until the last evaluated point, while only incurring a gap at the $1\times$ horizon. A detailed discussion of related work appears in Appendix A.

2 Theoretical Analysis

In this section, we provide an upper bound on the risk of SGD with iterate averaging, and show that the $1/t^{\gamma}$ schedule achieves the minimax rate exponent for power law Gaussian data established by Zhang et al. (2024a), up to logarithmic and constant factors. To complement these asymptotic results, Figure 3 evaluates the infinite dimensional (approximated by $d \gg N$) recursion for Gaussian linear regression with constant noise variance. The lower noise experiment in Figure 9 shows that decreasing σ^2 delays the variance-dominated crossover while leaving the schedules horizon-free.

Setup and notation. Throughout the theory section of the manuscript, we study linear regression on Gaussian data with batch size 1 and independent additive noise in the samples. We use the notation $f \lesssim g$ to say there exists some positive constant c such that for any x in the domain we have $f(x) \leq cg(x)$. We also denote $f \eqsim g$ if $f(x) \lesssim g(x) \lesssim f(x)$ for all x. Note that we will also absorb $\log$ factors in the $\lesssim$ notation, stating where we do so. We also denote the induced norm $\|\mathbf{w}\|_{\mathbf{A}}^2 = \mathbf{w}^\top \mathbf{A}\mathbf{w}$. For a diagonal matrix we define its restriction to entries between i and j where $1 \leq i \leq j \leq \infty$ as $\mathbf{\Lambda}_{i:j} = \mathrm{diag}(\lambda_i, \ldots, \lambda_j)$. We denote the i.i.d. input-output stream as $\{(\mathbf{x}_t, y_t)\}_{t \geq 1}$ where each $\mathbf{x}_t \in \mathbb{R}^d$ and $y_t \in \mathbb{R}$ satisfy:

$$\mathbf{x}_t \sim \mathcal{N}(0, \mathbf{H}) \qquad\qquad y_t | \mathbf{x}_t \sim \mathcal{N}(\mathbf{x}_t^\top \mathbf{w}^\star, \sigma^2)$$

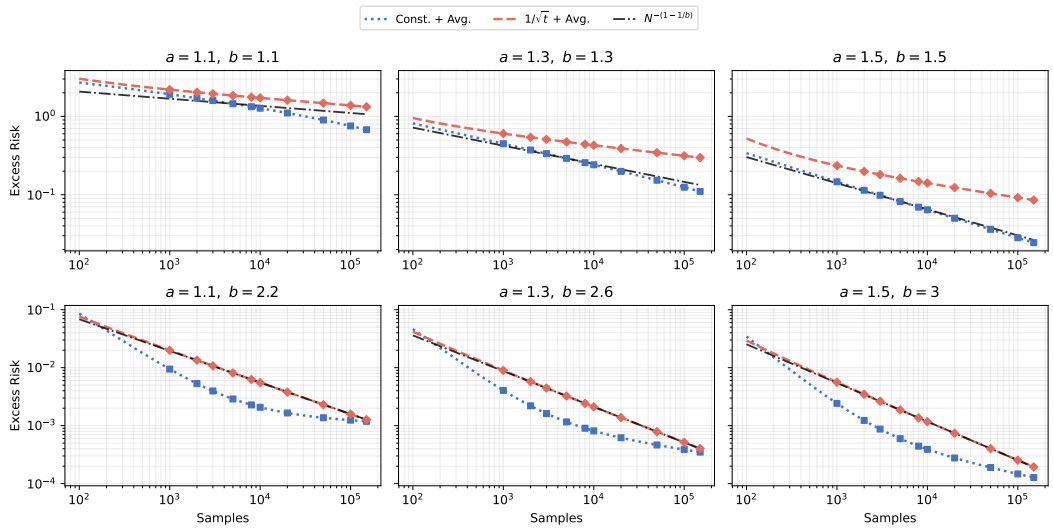


Figure 3: Synthetic excess risk for SGD on Gaussian linear regression with diagonal Hessian $\lambda_i = i^{-a}$, target coordinates $(w_i^\star)^2 = i^{-(b-a)}$, dimension $d = 500{,}000$, $N = 150{,}000$, and $\sigma^2 = 0.1$. Constant and $1/\sqrt{t}$ learning rates both use iterate averaging. For each panel and scheduler, we sweep fixed multipliers $c \in \{10^{-4}, 2{\times}10^{-4}, 5{\times}10^{-4}, 7{\times}10^{-4}, 10^{-3}, 2{\times}10^{-3}, 5{\times}10^{-3}, 10^{-2}, 2{\times}10^{-2}, 3{\times}10^{-2}, 5{\times}10^{-2}, 7.5{\times}10^{-2}, 0.1, 0.2, 0.3, 0.5, 0.8, 1, 2, 3, 5, 10\}$, where the actual step size is $c/\mathrm{Tr}(\mathbf{H})$ for constant LR and $c/(\mathrm{Tr}(\mathbf{H})\sqrt{t})$ for inverse square root. We select the learning rate using the best loss at $N = 5{,}000$ and evaluate that same trajectory at every later checkpoint. The dashed trace shows the predicted rate $N^{-(1-1/b)}$. We show the synthetic results for $\sigma^2 = 0.01$ in Figure 9.

where $\mathbf{w}^\star$ is the minimizer and σ^2 is the variance of the additive noise. We take the convention that $\lambda_{\max} = \lambda_1$ and eigenvalues are sorted in nonincreasing order. We define the risk under the mean squared error loss to be:

$$\mathcal{R}(\mathbf{w}) = \frac{1}{2}\mathbb{E}_{\mathbf{x},y}(\mathbf{x}^\top \mathbf{w} - y)^2$$

We denote the learning rate at time $t \geq 1$ by η_t and in general $\eta_t := \eta \cdot f(t, N)$ for some function f of the current time t and optionally the end time N, and for a constant base learning rate $\eta \in (0, 1)$ that does not depend on N. We take $\mathbf{w}_0 = 0$ as the initialization without loss of generality, and count updates from $t = 1$, so update t uses $(\mathbf{x}_t, y_t)$ and produces $\mathbf{w}_t$. We also denote $\bar{\mathbf{w}}_{s:s+T} := \frac{1}{T}\sum_{i=s}^{s+T-1} \mathbf{w}_i$ to be the average of the last T iterates starting from step $s \geq 1$. The main proof technique used to establish the rates is based on the bias-variance decomposition of the risk (Jain et al., 2018; 2017; Zou et al., 2023; Wu et al., 2022a;b; Meterez et al., 2025). All the proofs are deferred to Appendix B.

2.1 Main Results

We now state the main result regarding the excess risk rate of SGD with $\eta_t = 1/t^\gamma$. We state Theorem 1 using $s = \Theta(N)$ in order to simplify notation, but we provide the general bound in the proof.

Theorem 1. *Let $\mathbf{\Lambda} = diag(\lambda_i)_{i\geq 1}$ be the eigenvalues of $\mathbf{H}$ and learning rate $\eta_t = \eta/t^\gamma$ for $t \geq 1$, $\gamma \in (0, 1/2]$, and some constant $0 < \eta \lesssim 1/Tr(\mathbf{H})$ independent of N, and $\|\mathbf{w}^\star\|_{\mathbf{H}}^2 \lesssim \sigma^2$. We have for $s = \Theta(N)$:*

$$\mathbb{E}\mathcal{R}(\bar{\mathbf{w}}_{s:s+N}) - \frac{\sigma^2}{2} \lesssim \frac{1}{N}\|\mathbf{w}^\star\|_{\mathbf{\Lambda}_{1:k^\star}}^2 + \|\mathbf{w}^\star\|_{\mathbf{\Lambda}_{k^\star:\infty}}^2 + \frac{k^\star\sigma^2}{N} + \sigma^2 \sum_{k>k^\star} \left(\eta^2\lambda_k^2\, N^{1-2\gamma} + \eta\lambda_k\, N^{-\gamma}\right)$$

for $k^\star := \max\left\{k : \lambda_k \geq \frac{\log N}{\eta N^{1-\gamma}}\right\}$, *where* $\lesssim$ *absorbs absolute constants and the* $\log N$ *term for* $\gamma = 1/2$.

Following Jain et al. (2017), we have one contribution to the rate coming from the mean SGD process fitting the data and approaching the minimizer $\mathbf{w}^\star$ termed *bias*, and another term coming from the additive and sampling noise, known as *variance*. In finite dimension, taking $k^\star = d$ recovers the asymptotic $d\sigma^2/N$ variance rate (Polyak & Juditsky, 1992; Bach & Moulines, 2013; Ge et al., 2019).

In Corollary 1, we specialize the covariance spectrum to a power law setting, using source and capacity exponents (Caponnetto & De Vito, 2007; Lin et al., 2024; Bordelon et al., 2024; Paquette et al., 2024), and compute the optimal choice of gamma as a function of these exponents.

Corollary 1. *Suppose the following capacity and source conditions*

$$\lambda_k \eqsim k^{-a}, \qquad \lambda_k (w_k^\star)^2 \eqsim k^{-b}, \qquad 1 < a < b \leq 2a$$

and assume $\|\mathbf{w}^\star\|_{\mathbf{H}}^2 \lesssim \sigma^2$. *Then the rate-optimal decay exponent is*

$$\gamma^\star = 1 - \frac{a}{b} \in \left(0, \frac{1}{2}\right].$$

In particular, for $\eta_t = \eta/t^{\gamma^\star}$ *for* $t \geq 1$ *and* $s = \Theta(N)$, *Theorem 1 gives*

$$\mathbb{E}\mathcal{R}(\bar{\mathbf{w}}_{s:s+N}) - \frac{\sigma^2}{2} \lesssim N^{-\left(1-\frac{1}{b}\right)},$$

where $\lesssim$ *hides constants and the* $\log N$ *factors.*

From Corollary 1 we see that the optimal $\gamma^\star$ depends on the spectral properties of the data. Within the regime $1 < a < b \leq 2a$, larger values of b correspond to signal that decays more rapidly across the eigendirections of $\mathbf{H}$, and therefore to faster learning-rate decay: $\gamma^\star = 1 - a/b$ increases from 0 to $1/2$. Conversely, as b approaches a, $\gamma^\star$ approaches 0, for which prior work shows that constant-stepsize SGD with tail averaging can attain the corresponding optimal rates (Zhang et al., 2024b; Zou et al., 2023). Intuitively, when $b < a$, more target energy lies in tail directions, so maintaining a larger learning rate improves contraction of the bias.

Constant learning rate. Jain et al. (2018) (Theorem 1) have shown that in the classical regime, a constant learning rate with tail averaging suffices to achieve the minimax rate in the variance of $\sigma^2 d/N$. In the high dimensional regime, Zou et al. (2023); Zhang et al. (2024b) have shown that a constant learning rate with tail averaging can achieve the minimax SGD rate under source and capacity constraints. For $b \leq a$, Zhang et al. (2024b) (Corollary 2) show that a constant learning rate with averaging is a true anytime schedule, whereas for $b > a$ the learning rate has to scale with the horizon N. We have tried a constant learning rate with weight averaging in Figure 2, showing that this schedule performs competitively with cosine and $1/t^\gamma$.

WSD. WSD learning rate schedules have shown great promise empirically, being adopted by frontier labs (Team et al., 2025). Recall that the WSD schedule, as introduced by Hu et al. (2024), consists of a warmup stage, a constant learning rate stage and a decay over the last 10% of the training run. Related work by Bordelon & Mori (2026) (Result 2) and Li et al. (2026a) derive optimal last-iterate horizon dependent learning rate schedules in a power law random feature model, which recovers our linear regression setup for identity features. Informally, the WSD schedule is characterized by how long the constant learning rate region should lsat for as a function of the end time. Bordelon & Mori (2026) (Result 2) recover 2 separate schedules: for $b > a$, it is optimal to decay for the whole training run, whereas for $b < a$ they recover a schedule resembling WSD where the constant learning rate fraction scales as a function of the end time and the source and capacity exponents. The main focus of our work is complementary: we study whether horizon-free learning-rate trajectories, combined with iterate averaging, can achieve optimal rates without knowing the stopping time.

3 Empirical Findings

In this section, we describe how we compare our learning-rate schedules to cosine decay. We focus on anytime schedulers, which are intended to operate without prior knowledge of the training horizon.

Model	Schedule	Evaluated through	η	β_2	α
150M	Constant	32×	0.003	0.99	–
150M	$1/\sqrt{t}$	32×	0.003	0.98	51,200
300M	Constant	16×	0.003	0.95	–
300M	$1/\sqrt{t}$	16×	0.003	0.99	51,200

Table 1: Selected hyperparameters for the fixed constant-LR and $1/\sqrt{t}$ trajectories in Figure 2. Each row is one trajectory evaluated at every displayed checkpoint through the stated horizon. All runs use a 5,000-step warmup, $\beta_1 = 0.9$, $\epsilon = 10^{-8}$, zero weight decay, and batch size 256 for 150M or 512 for 300M.

For each anytime schedule we report results from *a single training run* using a hyperparameter setting chosen to track cosine performance across the entire training trajectory, rather than maximizing final-step performance. As a point of comparison, Figure 4 shows the transfer across subsequent horizons when optimally tuning for an earlier one, and Appendix C includes additional loss curves obtained by selecting the best hyperparameter setting separately for each training budget.

Architecture and Dataset. We pretrain 150M and 300M models based on the OLMo codebase (Groeneveld et al., 2024; OLMo et al., 2024; Olmo et al., 2025). The architectural details are reported as depth, number of heads and width: 150M (12, 16, 1024), 300M (24, 16, 1024). The models are trained using AdamW with no weight decay, fixing $\epsilon = 10^{-8}$ and momentum $\beta_1 = 0.9$. The 150M cosine runs sweep over $\eta \in \{0.0003, 0.001, 0.003, 0.01\}$ at the 1×–8× horizons and $\eta \in \{0.0003, 0.001, 0.003\}$ at the 16×–32× horizons, with $\beta_2 \in \{0.95, 0.98, 0.99\}$.

Note that our theoretical guarantees determine the asymptotic decay exponent only up to logarithmic and constant factors, however in practice a finite time offset is necessary. We parameterize the schedule as

$$\eta_t = \eta\sqrt{\frac{\alpha}{t+\alpha}},$$

where $\alpha > 0$ is a tuned offset controlling the transition from an approximately constant learning rate at $t \ll \alpha$ to $1/\sqrt{t}$ decay at $t \gg \alpha$. Once η and α are fixed, the schedule depends only on the current step t, rather than on the terminal training horizon. In Figure 2, α is selected as part of a single configuration that is then evaluated at every intermediate horizon; it is not retuned separately at each endpoint. Figure 7 examines sensitivity to this choice by fixing the remaining hyperparameters to those selected at 4× Chinchilla and comparing all executed values of α across the evaluated horizons. We refer to the resulting shifted schedule as $1/\sqrt{t}$ throughout for the remainder of this section.

For 150M, the constant learning-rate runs sweep over $\eta \in \{0.0003, 0.001, 0.003\}$ and $\beta_2 \in \{0.95, 0.98, 0.99\}$, while the inverse-square-root runs sweep over the same learning rates, $\beta_2 \in \{0.98, 0.99\}$, and offsets $\alpha \in \{12{,}800, 25{,}600, 51{,}200\}$. For 300M, all schedulers sweep over $\eta \in \{0.0003, 0.001, 0.003\}$ and $\beta_2 \in \{0.95, 0.99\}$, with the inverse-square-root runs additionally sweeping over offsets $\alpha \in \{12{,}800, 25{,}600, 51{,}200\}$. All the WSD runs are continued from the checkpoints saved during the constant learning rate sweep, and we plot the best (without averaging) at each point. We train without z-loss and with a sequence length $L = 1024$. We train our models on the C4 dataset (Raffel et al., 2020), and we use the T5 tokenizer, and we do not repeat over the data, with all our runs being fully online.

Training and evaluation details. Following the Chinchilla (Hoffmann et al., 2022) calculation of 20 tokens-per-parameter (TPP), we take as 1× Chinchilla for the 150M model to be 3.3B tokens and 6.6B tokens for the 300M model. Unless specified otherwise, we train all our models close to the critical batch size for 1× Chinchilla, based on Zhang et al. (2024b), which is 256 for 150M and following the $\sqrt{N}$ (for N denoting the total data size scaling), we approximate at 512 the critical batch size for the 300M model. We fix the warm-up duration to be 5000 steps (corresponding to approximately 40% of the respective 1× Chinchilla number of tokens for each of the models), for all training runs (the only exception being the $B = 4096$ runs, where we do 500 warmup steps). For the 150M model, we train a cosine baseline at each power-of-2 multiple of Chinchilla from 1× to 32×. For the $1/\sqrt{t}$ and constant with averaging runs, we train directly for the 32× and compare

with cosine at the intermediate points. Moreover, we save checkpoints at 90% of each Chinchilla multiple for the constant learning rate run, and do a linear decay from each of these points in order to implement the WSD schedule. We follow a similar procedure for the 300M models, but stop at $16\times$. We keep the batch size fixed to the $1\times$ Chinchilla critical batch size in order to mimic an anytime training regime in which one would start training a model for $1\times$ Chinchilla, and more data would be streamed in the training process down the line. We leave as future work how to combine anytime schedules with a batch ramp-up scheme.

The selected hyperparameters for the fixed constant-LR and $1/\sqrt{t}$ trajectories in Figure 2 are reported in Table 1. These configurations were chosen in order to minimize the average gap to the cosine envelope across all points, showing the existence of a *single run* for the 2 schedulers that closely tracks the cosine envelope at all points. For a prospective comparison, we show in Figure 4 that one can choose the $4\times$ optimal hyperparameters and still stay close to the cosine envelope, while incurring a penalty at the $1\times$ multiplier.

Averaging. We maintain EMA parameters

$$\bar{\mathbf{w}}_{t+1} = \tau_t \bar{\mathbf{w}}_t + (1-\tau_t)\theta_t \qquad \tau_t = 2^{-f/t}$$

At time t, the half life h satisfies

$$(\tau_t)^h = \frac{1}{2} \implies h = \frac{t}{f}$$

For all runs, we maintain multiple EMAs for $f \in \{6.25, 12.5, 25.0, 50.0, 100.0\}$ and treat $f = 0.0$ separately by fixing it to mean the last iterate. Thus, for $f = 25$ for example, the half life of the EMA is roughly $t/25 \approx 4\%$ of t. All coefficients depend only on the current time and a fixed value of f, not on the terminal training horizon. The averaging procedure is therefore online and horizon-free. Our implementation maintains 5 model copies (corresponding to each value of f) in addition to the trainable model. Unless mentioned otherwise, the constant learning rate and $1/\sqrt{t}$ curves in the main figures use $f = 25$ throughout, whereas cosine and WSD use the last iterate, $f = 0$. We do not select f separately at each training horizon. We decided to fix $f = 25$ across all experiments that use an EMA as this always corresponded to the best loss out of the EMA factors in our experiments.

Choice of γ. We experimentally compare $\gamma = 0$ (constant learning rate) and $\gamma = 1/2$, meaning $1/\sqrt{t}$, for which we provide the following explanation. Mlodozeniec et al. (2025); Bjorck et al. (2024) have shown that the optimal learning rate in practice approximately scales proportionally to $1/\sqrt{N}$. In the quadratic regime, the bias along eigendirection i contracts at a rate controlled by the cumulative step size, roughly as $\exp\big(-\lambda_i \sum_{s\le t} \eta_s\big)$. Consequently, the mean process continues to make substantial progress up to time N in all directions satisfying

$$\lambda_i \sum_{s=1}^{N} \eta_s \gtrsim 1, \qquad \implies \qquad \lambda_i \gtrsim \frac{1}{\sum_{s=1}^{N} \eta_s}.$$

For $\eta_s \eqsim 1/\sqrt{N}$, we have $\sum_{s=1}^{N} \eta_s \eqsim \sqrt{N}$. In order to ensure that an anytime scheme has the same scaling at any time t, one reasonable option is to choose $\eta_s = 1/\sqrt{s}$, since $\sum_{s=1}^{t} s^{-1/2} \eqsim \sqrt{t}$, thus motivating the choice of $\gamma = 1/2$. We believe that other schedules around this value might also work (Bjorck et al., 2024; Shuai et al., 2024), but for all our experiments, we chose to keep $1/2$. We formalize this bias–variance decomposition and the resulting recursion in Appendix B, equation 7.

3.1 Anytime Performance and Early-to-Late Hyperparameter Transfer

Fixed run comparison. Figure 2 shows that constant learning with averaging rate and $1/\sqrt{t}$ with averaging, provide competitive horizon-free alternatives to cosine decay. For each scheduler, we fix a single hyperparameter configuration and evaluate the same trajectory at every intermediate horizon. We also compare against WSD, based on Team et al. (2025); Hu et al. (2024): a warmup phase, followed by a constant learning rate until 90% of each target horizon and a linear decay over the remaining 10%, ending at 10% of the peak learning rate. In practice, we save checkpoints at 90% of each intermediate horizon and initiate a separate decay from each checkpoint. Unlike the constant and $1/\sqrt{t}$ trajectories, WSD is not horizon-free because the start of its decay depends on the selected stopping horizon.

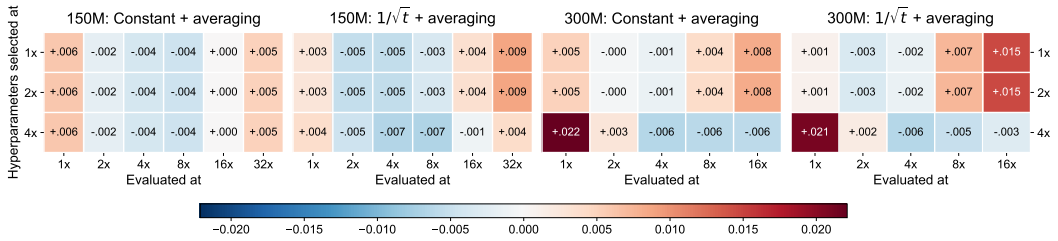


Figure 4: Prescriptive transfer for constant learning rate and $1/\sqrt{t}$. The rows select the configuration with the lowest validation loss at $1\times$, $2\times$, or $4\times$ Chinchilla and show the performance of that configuration at later points. Values are the 4%-EMA validation loss relative to the cosine envelope, and negative values indicate improvement over cosine. The hyperparameters are shown in Table 3 and Table 4.

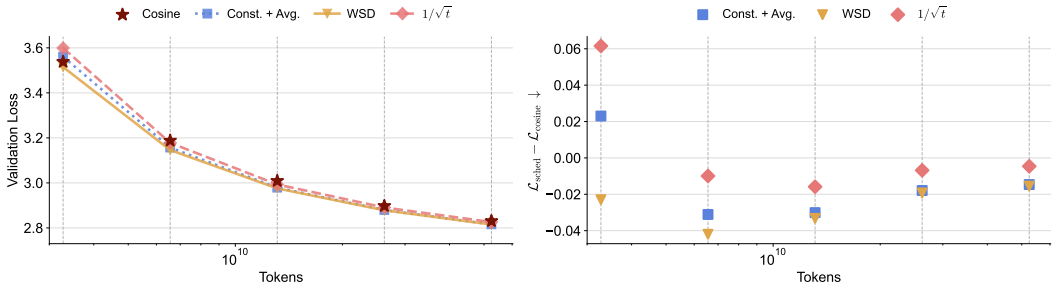


Figure 5: For a 150M-parameter model trained with batch size 4096 and a 500-step warmup, we compare cosine decay to constant learning rate with averaging, a $\sqrt{\alpha/(t+\alpha)}$ schedule with averaging, and WSD across end times ranging from $1\times$ to $16\times$ Chinchilla compute. **Left:** Validation loss versus training compute for each schedule. **Right:** Loss improvement over cosine at each compute multiple, where a negative value is an improvement over cosine. A per-duration optimal hyperparameter version of this plot is in Figure 8 (Appendix C).

Prospective hyperparameter transfer. The fixed trajectories in Figure 2 establish that a single configuration can remain competitive across the evaluated horizons, but practical use also requires a way to select that configuration. At a checkpoint T_0, we select the configuration with the lowest validation loss among the executed sweep. We then continue with the same configuration and evaluate it at every subsequent horizon, without using later-horizon losses for selection. We measure transfer relative to the independently tuned cosine envelope in Figure 4, with negative values indicating improvement over cosine. Across the evaluated settings, $4\times$ Chinchilla is the earliest common calibration horizon that transfers consistently for both schedules and model sizes, with $1\times$ also being close to it. For the 150M model, configurations selected at $4\times$ remain close to the cosine envelope through $32\times$: the maximum gaps are 0.0051 for constant learning rate and 0.0041 for $1/\sqrt{t}$. For the 300M model, both configurations selected at $4\times$ outperform the cosine envelope at every evaluated horizon from $4\times$ through $16\times$. This yields a practical anytime protocol: tune at an intermediate calibration checkpoint and continue training.

3.2 Large Batch Setting

Based on the quadratic analysis, learning rate decay has the role of reducing variance. However, there is an inherent trade-off: a smaller learning rate has a smaller variance, but also a worse contraction in the bias term. In this section we test whether this theoretical prediction from the quadratic model is also relevant in LLM pretraining: if the variance is already very small, is decay still necessary to get a good final performance? Concretely, in the quadratic regime, SGD can be written as GD plus a batch-noise term:

$$\theta_{t+1} - \theta^\star \approx (I - \eta H)(\theta_t - \theta^\star) + \eta \xi_t,$$

where $\mathbb{E}[\xi_t] = 0$ and $\mathrm{Cov}(\xi_t) \propto 1/B$. Thus, moving to very large batches suppresses gradient noise, making learning rate decay less necessary for controlling the variance term, and a larger learning rate more advatangeous for faster bias decay. In this section, we repeat the experiment from Figure 2 in a very large-batch regime (batch size 4096), well beyond the critical batch size (CBS). While this regime is not relevant for practice, the large batch size reduces the variance in the risk linearly, stress testing the predictivity of the quadratic model for LLM pretraining experiments. Figure 5 is consistent with this prediction, since for $B = 4096$, a constant learning rate with averaging is generally better than cosine for all horizons beyond $1\times$ Chinchilla. The $1/\sqrt{t}$ schedule also remains competitive and improves over cosine, but underperforms constant, which is again consistent with unnecessary decay in this regime. Lastly, the difference between constant with averaging and WSD decreases with N. We believe this is due to a higher order effect caused by the deterministic edge of stability as studied in Damian et al. (2023). However, we leave the precise study of this phenomenon to future work.

4 Discussion and Limitations

4.1 Limitations

Theoretical Limitations. From a theoretical point of view, our upper bound matches the minimax lower bound exponent established by Zhang et al. (2024a) up to constants and logarithmic factors. An important future work would be to remove the $\log N$ factor while still having an anytime scheduler.

Hyperparameter selection. We evaluate the practical anytime setting by selecting the optimal hyperparameters at $4\times$ Chinchilla and continuing the same runs unchanged through $32\times$ for the 150M model and $16\times$ for the 300M model. The selected configurations remain competitive at subsequent horizons, showing that hyperparameters chosen early in training can transfer to substantially longer runs without advance knowledge of their final duration. However, the $4\times$ is a heuristic choice, and it is not clear how to choose the point for which to tune the hyperparameters for in order to obtain transfer down the line. We believe one approach to this would be to build scaling laws across the schedulers in order to understand the optimal tuning point, and we leave this important extension to future work.

Scope. Our empirical conclusions are limited to 150M- and 300M-parameter models trained with AdamW, no weight decay, and the batch-size regimes considered here. In particular, weight decay, alternative optimizers, and substantially larger model scales may change the relative behavior of the schedules.

4.2 Conclusion

In this work, we study the feasibility of horizon-free learning-rate schedules for LLM pretraining, specifically a constant learning rate and $1/\sqrt{t}$ combined with EMA. Across well-tuned cosine-decay baselines for 150M and 300M parameter models, spanning $1\times$ to $32\times$ (and $1\times$ to $16\times$, respectively) Chinchilla regimes, we find that these anytime schedules are close to cosine (within small gaps) at intermediate checkpoints without requiring knowledge of the total training horizon. This suggests a promising direction for continual and open-ended training. On the theory side, for linear regression with power-law spectra, we derive convergence rates as a function of source and capacity. We show that under spectral constraints, a schedule of the form $1/t^{\gamma}$ attains optimal rates up to logarithmic and constant factors, with $1/\sqrt{t}$ as a notable special case. We also compare against WSD empirically: while WSD is not strictly horizon-independent because it relies on additional checkpoints near 90% of the run, it achieves performance comparable to cosine and the proposed anytime schedules. Overall, our results indicate that anytime schedules can be competitive with carefully tuned cosine baselines, emphasize the practical importance of weight averaging in pretraining, and motivate further study of horizon-free optimization.

5 AI Disclosure

In this work, we used generative AI tools for generating figures, proof checking the mathematical results, and proof reading the final manuscript. We have reviewed all AI-assisted work manually by deriving the mathematical formulations and the code, as well as text and plots. We take responsibility

for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Niccolò Ajroldi, Antonio Orvieto, and Jonas Geiping. When, where and why to average weights? *arXiv preprint arXiv:2502.06761*, 2025.

Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. *Advances in neural information processing systems*, 32, 2019.

Devansh Arpit, Huan Wang, Yingbo Zhou, and Caiming Xiong. Ensemble of averages: Improving model selection and boosting performance in domain generalization. *Advances in Neural Information Processing Systems*, 35:8265–8277, 2022.

Alexander Atanasov, Jacob A Zavatone-Veth, and Cengiz Pehlevan. Scaling and renormalization in high-dimensional regression. *arXiv preprint arXiv:2405.00592*, 2024.

Alexander Atanasov, Blake Bordelon, Jacob A Zavatone-Veth, Courtney Paquette, and Cengiz Pehlevan. Two-point deterministic equivalence for stochastic gradient dynamics in linear models. *arXiv preprint arXiv:2502.05074*, 2025.

Francis Bach and Eric Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate o (1/n). *Advances in neural information processing systems*, 26, 2013.

Annalisa Belloni, Lorenzo Noci, and Antonio Orvieto. Universal dynamics of warmup stable decay: understanding wsd beyond transformers. *arXiv preprint arXiv:2601.09000*, 2026.

Shane Bergsma, Nolan Dey, Gurpreet Gosal, Gavia Gray, Daria Soboleva, and Joel Hestness. Straight to zero: Why linearly decaying the learning rate to zero works best for llms. *arXiv preprint arXiv:2502.15938*, 2025.

Johan Bjorck, Alon Benhaim, Vishrav Chaudhary, Furu Wei, and Xia Song. Scaling optimal lr across token horizons. *arXiv preprint arXiv:2409.19913*, 2024.

Blake Bordelon and Francesco Mori. Theory of optimal learning rate schedules and scaling laws for a random feature model. *arXiv preprint arXiv:2602.04774*, 2026.

Blake Bordelon and Cengiz Pehlevan. Learning curves for sgd on structured features. *arXiv preprint arXiv:2106.02713*, 2021.

Blake Bordelon, Alexander Atanasov, and Cengiz Pehlevan. A dynamical model of neural scaling laws. *arXiv preprint arXiv:2402.01092*, 2024.

Lucas Caccia, Jing Xu, Myle Ott, Marcaurelio Ranzato, and Ludovic Denoyer. On anytime learning at macroscale. In *Conference on Lifelong Learning Agents*, pp. 165–182. PMLR, 2022.

Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

Andrea Cossu, Antonio Carta, Lucia Passaro, Vincenzo Lomonaco, Tinne Tuytelaars, and Davide Bacciu. Continual pre-training mitigates forgetting in language and vision. *Neural Networks*, 179: 106492, 2024.

George E Dahl, Frank Schneider, Zachary Nado, Naman Agarwal, Chandramouli Shama Sastry, Philipp Hennig, Sourabh Medapati, Runa Eschenhagen, Priya Kasimbeg, Daniel Suo, et al. Benchmarking neural network training algorithms. *arXiv preprint arXiv:2306.07179*, 2023.

Alex Damian, Eshaan Nichani, and Jason D. Lee. Self-stabilization: The implicit bias of gradient descent at the edge of stability. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=nhKHA59gXz.

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

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

Alexandre Défossez and Francis Bach. Averaged least-mean-squares: Bias-variance trade-offs and optimal sampling distributions. In *Artificial Intelligence and Statistics*, pp. 205–213. PMLR, 2015.

Aymeric Dieuleveut and Francis Bach. Nonparametric stochastic approximation with large step-sizes. *arXiv preprint arXiv:1408.0361*, 2014.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Roy Frostig, Rong Ge, Sham M Kakade, and Aaron Sidford. Competing with the empirical risk minimizer in a single pass. In *Conference on learning theory*, pp. 728–763. PMLR, 2015.

Rong Ge, Sham M Kakade, Rahul Kidambi, and Praneeth Netrapalli. The step decay schedule: A near optimal, geometrically decaying learning rate procedure for least squares. *Advances in neural information processing systems*, 32, 2019.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. Olmo: Accelerating the science of language models. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 15789–15809, 2024.

Kshitij Gupta, Benjamin Thérien, Adam Ibrahim, Mats L Richter, Quentin Anthony, Eugene Belilovsky, Irina Rish, and Timothée Lesort. Continual pre-training of large language models: How to (re) warm your model? *arXiv preprint arXiv:2308.04014*, 2023.

Alex Hägele, Elie Bakouch, Atli Kosson, Leandro Von Werra, Martin Jaggi, et al. Scaling laws and compute-optimal training beyond fixed training durations. *Advances in Neural Information Processing Systems*, 37:76232–76264, 2024.

Nicholas JA Harvey, Christopher Liaw, Yaniv Plan, and Sikander Randhawa. Tight analyses for non-smooth stochastic gradient descent. In *Conference on Learning Theory*, pp. 1579–1613. PMLR, 2019.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

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

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. Simple and scalable strategies to continually pre-train large language models. *arXiv preprint arXiv:2403.08763*, 2024.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Prateek Jain, Sham M Kakade, Rahul Kidambi, Praneeth Netrapalli, Venkata Krishna Pillutla, and Aaron Sidford. A markov chain theory approach to characterizing the minimax optimality of stochastic gradient descent (for least squares). *arXiv preprint arXiv:1710.09430*, 2017.

Prateek Jain, Sham M Kakade, Rahul Kidambi, Praneeth Netrapalli, and Aaron Sidford. Parallelizing stochastic gradient descent for least squares regression: mini-batching, averaging, and model misspecification. *Journal of machine learning research*, 18(223):1–42, 2018.

Prateek Jain, Dheeraj Nagaraj, and Praneeth Netrapalli. Making the last iterate of sgd information theoretically optimal. In *Conference on Learning Theory*, pp. 1752–1755. PMLR, 2019.

Amir Joudaki, Giulia Lanzillotta, Mohammad Samragh Razlighi, Iman Mirzadeh, Keivan Alizadeh, Thomas Hofmann, Mehrdad Farajtabar, and Fartash Faghri. Barriers for learning in an evolving world: Mathematical understanding of loss of plasticity. *arXiv preprint arXiv:2510.00304*, 2025.

Jean Kaddour. Stop wasting my time! saving days of imagenet and bert training with latest weight averaging. *arXiv preprint arXiv:2209.14981*, 2022.

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24174–24184, 2024.

Priya Kasimbeg, Frank Schneider, Runa Eschenhagen, Juhan Bae, Chandramouli Shama Sastry, Mark Saroufim, Boyuan Feng, Less Wright, Edward Z Yang, Zachary Nado, et al. Accelerating neural network training: An analysis of the algoperf competition. *arXiv preprint arXiv:2502.15015*, 2025.

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

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

Siyuan Li, Zicheng Liu, Juanxi Tian, Ge Wang, Zedong Wang, Weiyang Jin, Di Wu, Cheng Tan, Tao Lin, Yang Liu, et al. Switch ema: A free lunch for better flatness and sharpness. *arXiv preprint arXiv:2402.09240*, 2024.

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

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Kairong Luo, Haodong Wen, Shengding Hu, Zhenbo Sun, Maosong Sun, Zhiyuan Liu, Kaifeng Lyu, and Wenguang Chen. A multi-power law for loss curve prediction across learning rate schedules. In *International Conference on Learning Representations*, volume 2025, pp. 39973–40017, 2025.

Sanket Vaibhav Mehta, Darshan Patil, Sarath Chandar, and Emma Strubell. An empirical investigation of the role of pre-training in lifelong learning. *Journal of Machine Learning Research*, 24(214): 1–50, 2023.

Alexandru Meterez, Depen Morwani, Costin-Andrei Oncescu, Jingfeng Wu, Cengiz Pehlevan, and Sham Kakade. A simplified analysis of sgd for linear regression with weight averaging. *arXiv preprint arXiv:2506.15535*, 2025.

Bruno Mlodozeniec, Pierre Ablin, Louis Béthune, Dan Busbridge, Michal Klein, Jason Ramapuram, and Marco Cuturi. Completed hyperparameter transfer across modules, width, depth, batch and duration. *arXiv preprint arXiv:2512.22382*, 2025.

Daniel Morales-Brotons, Thijs Vogels, and Hadrien Hendrikx. Exponential moving average of weights in deep learning: Dynamics and benefits. *arXiv preprint arXiv:2411.18704*, 2024.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Rui Pan, Haishan Ye, and Tong Zhang. Eigencurve: Optimal learning rate schedule for sgd on quadratic objectives with skewed hessian spectrums. *arXiv preprint arXiv:2110.14109*, 2021.

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

Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. Effect of scale on catastrophic forgetting in neural networks. In *International conference on learning representations*, 2021.

Sunny Sanyal, Atula Neerkaje, Jean Kaddour, Abhishek Kumar, and Sujay Sanghavi. Early weight averaging meets high learning rates for llm pre-training. *arXiv preprint arXiv:2306.03241*, 2023.

Fabian Schaipp, Alexander Hägele, Adrien Taylor, Umut Simsekli, and Francis Bach. The surprising agreement between convex optimization theory and learning-rate scheduling for large model training. *arXiv preprint arXiv:2501.18965*, 2025.

Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International conference on machine learning*, pp. 71–79. PMLR, 2013.

Xian Shuai, Yiding Wang, Yimeng Wu, Xin Jiang, and Xiaozhe Ren. Scaling law for language models training considering batch size. *arXiv preprint arXiv:2412.01505*, 2024.

Vaibhav Singh, Paul Janson, Paria Mehrbod, Adam Ibrahim, Irina Rish, Eugene Belilovsky, and Benjamin Thérien. Beyond cosine decay: On the effectiveness of infinite learning rate schedule for continual pre-training. *arXiv preprint arXiv:2503.02844*, 2025.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Howe Tissue, Venus Wang, and Lu Wang. Scaling law with learning rate annealing. *Advances in Neural Information Processing Systems*, 38:91017–91053, 2026.

Tom Veniat, Ludovic Denoyer, and Marc'Aurelio Ranzato. Efficient continual learning with modular networks and task-driven priors. *arXiv preprint arXiv:2012.12631*, 2020.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5362–5383, 2024.

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

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. PMLR, 2022.

Jingfeng Wu, Difan Zou, Vladimir Braverman, Quanquan Gu, and Sham Kakade. Last iterate risk bounds of sgd with decaying stepsize for overparameterized linear regression. In *International conference on machine learning*, pp. 24280–24314. PMLR, 2022a.

Jingfeng Wu, Difan Zou, Vladimir Braverman, Quanquan Gu, and Sham Kakade. The power and limitation of pretraining-finetuning for linear regression under covariate shift. *Advances in Neural Information Processing Systems*, 35:33041–33053, 2022b.

Yasin Yazıcı, Chuan-Sheng Foo, Stefan Winkler, Kim-Hui Yap, Georgios Piliouras, and Vijay Chandrasekhar. The unusual effectiveness of averaging in gan training. *arXiv preprint arXiv:1806.04498*, 2018.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1204–1213. IEEE, 2022.

Haihan Zhang, Yuanshi Liu, Qianwen Chen, and Cong Fang. The optimality of (accelerated) sgd for high-dimensional quadratic optimization. *arXiv preprint arXiv:2409.09745*, 2024a.

Hanlin Zhang, Depen Morwani, Nikhil Vyas, Jingfeng Wu, Difan Zou, Udaya Ghai, Dean Foster, and Sham Kakade. How does critical batch size scale in pre-training? *arXiv preprint arXiv:2410.21676*, 2024b.

Difan Zou, Jingfeng Wu, Vladimir Braverman, Quanquan Gu, and Sham M Kakade. Benign overfitting of constant-stepsize sgd for linear regression. *Journal of Machine Learning Research*, 24(326):1–58, 2023.

Schedule	Averaging	Horizon-free	Rate	Regime	Reference
Constant η	yes	✓	$N^{-(b-1)/a}$	$1 < b \leq a$	Zhang et al. (2024b)
Constant $\eta \propto N^{a/b-1}$	yes	×	$N^{-(1-1/b)}$	$a < b \leq 2a$	Zhang et al. (2024b)
Horizon-optimal schedule	last iterate	×	$N^{-(b-1)/\max\{a,b\}}$	both	Bordelon & Mori (2026)
$1/t^{\gamma^\star}, \gamma^\star = 1 - a/b$	yes	✓	$N^{-(1-1/b)}$	$1 < a < b \leq 2a$	**this work** (Thm 1, Cor. 1)

Table 2: Excess-risk rates for SGD in infinite dimensional linear regression and closely related models. We use capacity exponent a and source exponent b. Horizon-free means that the learning rate scheduler does not depend on the training horizon N. Rates are presented up to constants and logarithmic factors.

A Related Work

Finite dimensional SGD analysis. There is a wide body of literature studying risk bounds in stochastic gradient descent, both in the finite dimensional regime and in the infinite dimensional/nonparametric setup. Polyak & Juditsky (1992) introduced averaged SGD as an algorithm to achieve improved SGD convergence rates, while Bach & Moulines (2013) later analyzed the rate obtained by it in least squares. Défossez & Bach (2015) analyzed constant learning rate with averaged iterates for least-squares, providing rates for the bias and variance terms, with similar proofs being shown in Jain et al. (2017); Dieuleveut & Bach (2014). This analysis has been extended to minibatch gradient descent (Jain et al., 2018) and streaming algorithms (Frostig et al., 2015). When the horizon is known in advance, Jain et al. (2019) have shown that a carefully designed step size sequence can be minimax optimal for last iterate SGD, building up on previous work by Shamir & Zhang (2013); Harvey et al. (2019). Ge et al. (2019) showed that the final iterate under polynomially decaying stepsizes is provably off by a condition-number factor in the strongly convex case, and no better than $d\sigma^2/\sqrt{T}$ against the minimax $d\sigma^2/T$ in the smooth case, while a geometrically decaying step-decay schedule is near-minimax-optimal up to logarithmic factors, with Pan et al. (2021) refining the schedule further.

Nonparametric Least Squares. Recent work by Zhang et al. (2024b) has shown that for power law spectra and under certain conditions on source and capacity exponents, averaging can be minimax optimal. Zhang et al. (2024a) have established minimax optimal rates for SGD-type methods under power-law source and capacity conditions, characterizing the regimes in which plain SGD attains the minimax rate and where acceleration is required. Other learning rate schedules have been analyzed using a bias variance decomposition technique (Zou et al., 2023; Wu et al., 2022a;b; Meterez et al., 2025). From a statistical physics point of view, Bordelon & Pehlevan (2021) have established precise asymptotics for SGD in the overparameterized regime, with a similar analysis being done by Atanasov et al. (2024; 2025) using tools from random matrix theory.

Theory and Practice for Learning Rate Scheduling. Defazio et al. (2023) have analyzed the linear decay schedule and have proposed further refinements for this schedule using gradient norms. Linearly decaying to zero has been studied empirically by Bergsma et al. (2025), achieving competitive performance to cosine annealing. Defazio et al. (2024) have proposed a schedule-free optimizer, which takes advantage of tail averaging to remove the need for learning rate scheduling. Recent works leverage power-law kernel and random-feature regression models to derive optimal learning-rate schedules for last-iterate performance (Li et al., 2026b; Bordelon & Mori, 2026). Closely related to our work, Hägele et al. (2024) show that a constant learning rate with a decay at the end (a schedule similar to the trapezoidal scheduler proposed by Zhai et al. (2022) and WSD (Hu et al., 2024)) performs close to cosine decay at matched training lengths. Moreover, they find that stochastic weight averaging (SWA) with a fixed window improves models throughout training, but does not fully close the gap to learning rate decay. Our results complement this finding: guided by the quadratic analysis, we use an EMA window that grows proportionally with elapsed training time, and ask whether a single fixed configuration can track the envelope of per-horizon-tuned cosine baselines at every intermediate checkpoint of runs extending to $32\times$ Chinchilla. In this regime, we find that averaging can track the cosine envelope with no decay phase. Schaipp et al. (2025) show that last-iterate bounds from convex optimization closely predict the behavior of practical schedules, explaining the benefit of cooldowns. Tissue et al. (2026); Luo et al. (2025); Li et al. (2026a) model the loss curve as a

function the scheduler, predicting behaviour across horizons and schedulers. More recently, stochastic weight averaging has obtained impressive results in the AlgoPerf competition (Kasimbeg et al., 2025; Dahl et al., 2023), with Ajroldi et al. (2025) providing a systematic study of when averaging helps. Weight averaging has also been commonly applied in image generation and diffusion (Yazıcı et al., 2018; Karras et al., 2024; Song et al., 2020). Other variants of weight averaging have been used in practice (Kaddour, 2022; Sanyal et al., 2023; Li et al., 2024; Morales-Brotons et al., 2024; Arpit et al., 2022; Wortsman et al., 2022). It has been empirically observed that averaging can also lead to flatter minima and improved generalization performance on multiple image tasks (Izmailov et al., 2018).

Continual Learning and Anytime Training. Generally, continual learning refers to training a model on a sequence of (possibly orthogonal) tasks, ensuring that the model does not forget any of them (Wang et al., 2024; Aljundi et al., 2019; Veniat et al., 2020; Cossu et al., 2024; Ramasesh et al., 2021; Mehta et al., 2023; Joudaki et al., 2025). Anytime training (Caccia et al., 2022; Ibrahim et al., 2024; Gupta et al., 2023) refers to training a model optimally without having access to the total amount of steps i.e. the training horizon. While the two themes appear separate, we believe they are closely related, as anytime pretraining is a necessary stepping stone towards continual learning. A possible technique for anytime pretraining involves re-warming and re-decaying the learning rate after each stage of training (Ibrahim et al., 2024), which has been explored by Wen et al. (2024) in the context of Warmup-Stable-Decay (WSD) (Hu et al., 2024) schedules. Learning dynamics of WSD have been studied empirically by (Belloni et al., 2026). Singh et al. (2025) show that infinite schedules which avoid re-warming altogether consistently improve continual pretraining over repeated cosine cycles.

B PROOFS OF SECTION 2

Helper lemmas. Before beginning the main proofs, we state a few helper lemmas. Also note that for the majority of the proofs, we will make heavy use of the inequality:

$$(1-x)^t \leq \exp(-tx) \tag{1}$$

Lemma 1. *Let $0 < \alpha \leq \frac{1}{2}$ and let $1 \leq j \leq i$. Then*

$$i^{-\alpha}(i-j) \;\lesssim\; i^{1-\alpha} - j^{1-\alpha} \;\lesssim\; j^{-\alpha}(i-j).$$

Proof. Let $f(x) = x^{1-\alpha}$. By the mean value theorem applied on $[j, i]$, there exists $c \in [j, i]$ such that

$$i^{1-\alpha} - j^{1-\alpha} = (1-\alpha)c^{-\alpha}(i-j).$$

Because $x \mapsto x^{-\alpha}$ is decreasing,

$$(1-\alpha)i^{-\alpha}(i-j) \leq i^{1-\alpha} - j^{1-\alpha} \leq (1-\alpha)j^{-\alpha}(i-j).$$

Since $0 < \alpha \leq \frac{1}{2}$, we have $\frac{1}{2} \leq 1-\alpha \leq 1$ which proves the claim. □

Lemma 2. *Let $\mathbf{\Lambda} = diag(\lambda_i)$ be a diagonal matrix of nonnegative eigenvalues, let λ be the vector corresponding to its diagonal, and suppose that $(\eta_t)_{t\geq 1}$ is nonincreasing and satisfies*

$$0 < \eta_t \leq \eta_1 \leq \frac{1}{2Tr(\mathbf{H})}.$$

Then, for the recurrence

$$\mathbf{m}_t = \left(\mathbf{I} - 2\eta_t\mathbf{\Lambda} + 2\eta_t^2\mathbf{\Lambda}^2 + \eta_t^2\lambda\lambda^\top\right)\mathbf{m}_{t-1} + \eta_t^2\sigma^2\lambda, \qquad t \geq 1,$$

with $\mathbf{m}_0 \geq 0$ componentwise, we have

$$\lambda^\top\mathbf{m}_t \leq 2\lambda^\top\mathbf{m}_0 + \sigma^2 \qquad \forall t \geq 0.$$

Proof. We first denote the diagonal matrices

$$\mathbf{D}_t = \mathbf{I} - 2\eta_t\mathbf{\Lambda} + 2\eta_t^2\mathbf{\Lambda}^2.$$

We rewrite the recurrence as

$$\mathbf{m}_t = \mathbf{D}_t\mathbf{m}_{t-1} + \eta_t^2\lambda(\lambda^\top\mathbf{m}_{t-1} + \sigma^2)$$

Unrolling the recurrence gives

$$\mathbf{m}_t = \left(\prod_{r=1}^{t}\mathbf{D}_r\right)\mathbf{m}_0 + \sum_{s=1}^{t}\eta_s^2\left(\prod_{r=s+1}^{t}\mathbf{D}_r\right)\lambda(\lambda^\top\mathbf{m}_{s-1} + \sigma^2),$$

where an empty product is understood to be $\mathbf{I}$.

Dotting with λ, we have

$$\lambda^\top\mathbf{m}_t = \lambda^\top\left(\prod_{r=1}^{t}\mathbf{D}_r\right)\mathbf{m}_0 + \sum_{s=1}^{t}\eta_s^2\lambda^\top\left(\prod_{r=s+1}^{t}\mathbf{D}_r\right)\lambda(\lambda^\top\mathbf{m}_{s-1} + \sigma^2).$$

First, note that for each diagonal entry of $\mathbf{D}_t$, we have

$$0 \leq \mathbf{D}_{t,i} = 1 - 2\eta_t\lambda_i + 2\eta_t^2\lambda_i^2 = (1-\eta_t\lambda_i)^2 + \eta_t^2\lambda_i^2 \leq 1.$$

This implies that, elementwise,

$$\lambda^\top\left(\prod_{r=1}^{t}\mathbf{D}_r\right)\mathbf{m}_0 \leq \lambda^\top\mathbf{m}_0. \tag{2}$$

Next, we look at the other term. Since

$$1 - \mathbf{D}_{s,i} = 2\eta_s\lambda_i(1-\eta_s\lambda_i),$$

and $\eta_s\lambda_i \le 1/2$, we have

$$\eta_s^2\lambda_i^2 = \frac{\eta_s\lambda_i}{2(1-\eta_s\lambda_i)}(1-\mathbf{D}_{s,i}) \le \eta_1\lambda_i(1-\mathbf{D}_{s,i}).$$

Therefore,

$$\begin{aligned}
&\sum_{s=1}^{t}\eta_s^2\lambda^\top\left(\prod_{r=s+1}^{t}\mathbf{D}_r\right)\lambda \\
&\le \eta_1\sum_i\lambda_i\sum_{s=1}^{t}(1-\mathbf{D}_{s,i})\prod_{r=s+1}^{t}\mathbf{D}_{r,i} \\
&= \eta_1\sum_i\lambda_i\left(1-\prod_{r=1}^{t}\mathbf{D}_{r,i}\right) &(3)\\
&\le \eta_1\mathrm{Tr}(\mathbf{H}) \le \frac{1}{2}. &(4)
\end{aligned}$$

We prove the final statement inductively. Define the hypothesis

$$P(t) : \lambda^\top\mathbf{m}_t \le 2\lambda^\top\mathbf{m}_0 + \sigma^2.$$

The base case $P(0)$ is trivially true. Now suppose $P(k)$ is true for all $k < t$. We want to prove $P(t)$. We have

$$\begin{aligned}
\lambda^\top\mathbf{m}_t &= \lambda^\top\left(\prod_{r=1}^{t}\mathbf{D}_r\right)\mathbf{m}_0 + \sum_{s=1}^{t}\eta_s^2\lambda^\top\left(\prod_{r=s+1}^{t}\mathbf{D}_r\right)\lambda(\lambda^\top\mathbf{m}_{s-1}+\sigma^2) \\
&\le \lambda^\top\mathbf{m}_0 + \sum_{s=1}^{t}\eta_s^2\lambda^\top\left(\prod_{r=s+1}^{t}\mathbf{D}_r\right)\lambda(\lambda^\top\mathbf{m}_{s-1}+\sigma^2) && \textit{equation 2}\\
&\le \lambda^\top\mathbf{m}_0 + (2\lambda^\top\mathbf{m}_0+2\sigma^2)\sum_{s=1}^{t}\eta_s^2\lambda^\top\left(\prod_{r=s+1}^{t}\mathbf{D}_r\right)\lambda && P(s-1)\\
&\le 2\lambda^\top\mathbf{m}_0+\sigma^2 && \textit{equation 3}.
\end{aligned}$$

This proves $P(t)$ and completes the induction. □

Setup. Before proceeding to the proofs, we briefly restate the notation and setup from Section 2. Denote the independent and identically distributed input output pairs $\{(\mathbf{x}_i, y_i)\}_{i\ge1}$ where each $\mathbf{x}_i$ and y_i are $\mathbf{x}_i \sim \mathcal{N}(0,\mathbf{H})$ and $y_i = \mathbf{x}_i^\top w^\star + \epsilon_i$ for independent $\epsilon_i \sim \mathcal{N}(0,\sigma^2)$. We take $\mathbf{w}_0 = 0$ as the initialization and number updates from $t=1$, so update t uses $(\mathbf{x}_t, y_t)$ and produces $\mathbf{w}_t$. Here, $s \ge 1$ is the step from which we start tail averaging. Define the risk to be $\mathcal{R}(\mathbf{w}) = \frac{1}{2}\mathbb{E}_{\mathbf{x},y}(\langle\mathbf{w},\mathbf{x}\rangle - y)^2$ where the expectation is over the joint $(\mathbf{x},y)\sim\mathcal{D}$. Consider the learning rate schedule

$$\eta_t = \frac{\eta}{t^\gamma} \qquad t \ge 1$$

for $\gamma \in (0, 1/2]$ and a constant $0 < \eta \le 1/2\mathrm{Tr}(\mathbf{H})$. Denote the eigendecomposition of $\mathbf{H} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$ and let $\mathbf{\Sigma}_i = \mathbb{E}[(\mathbf{w}_i - \mathbf{w}^\star)(\mathbf{w}_i-\mathbf{w}^\star)^\top]$ be the second moment of the iterates. We also introduce $\mathbf{M}_t = \mathbf{Q}^\top\mathbf{\Sigma}_t\mathbf{Q}$ as the covariance of the iterates rotated in the eigenbasis of the data, and $\mathbf{m}_t = \mathrm{diag}(\mathbf{M}_t)$ as its diagonal. Note that for $\mathbf{A},\mathbf{B}$ matrices we have $\langle\mathbf{A},\mathbf{B}\rangle = \mathrm{Tr}(\mathbf{A}^\top\mathbf{B})$. We also denote the induced norm $\|\mathbf{w}\|_{\mathbf{A}}^2 = \mathbf{w}^\top\mathbf{A}\mathbf{w}$. For a diagonal matrix we define its restriction to entries between i and j where $1 \le i \le j \le \infty$ as $\mathbf{\Lambda}_{i:j} = \mathrm{diag}(\lambda_i,\dots,\lambda_j)$. Notationally, for $\mathbf{w}$ and $\mathbf{v}$ multidimensional quantities (vectors or matrices), we mean $\mathbf{w} \le \mathbf{v}$ as componentwise comparison, and we will use $\preceq$ to denote the Loewner order comparisons.

B.1 Proof of Theorem 1

Proof of Theorem 1. We begin with deriving the expression for the risk:

$$\mathcal{R}(\bar{\mathbf{w}}_{s:s+N}) = \frac{1}{2}\mathbb{E}\left[\left(\langle \bar{\mathbf{w}}_{s:s+N} - \mathbf{w}^\star, \mathbf{x}\rangle - \epsilon\right)^2\right].$$

Expanding and using $\mathbb{E}[\epsilon] = 0$,

$$\mathcal{R}(\bar{\mathbf{w}}_{s:s+N}) = \frac{1}{2}\mathbb{E}\left[\langle \bar{\mathbf{w}}_{s:s+N} - \mathbf{w}^\star, \mathbf{x}\rangle^2\right] + \frac{\sigma^2}{2}.$$

Substituting $\bar{\mathbf{w}}_{s:s+N} = \frac{1}{N}\sum_{i=s}^{s+N-1}\mathbf{w}_i$,

$$\begin{aligned}\mathcal{R}(\bar{\mathbf{w}}_{s:s+N}) &= \frac{1}{2N^2}\sum_{i=s}^{s+N-1}\sum_{j=s}^{s+N-1}\mathbb{E}\left[\mathbf{x}^\top(\mathbf{w}_i - \mathbf{w}^\star)(\mathbf{w}_j - \mathbf{w}^\star)^\top\mathbf{x}\right] + \frac{\sigma^2}{2} \\ &\leq \frac{1}{N^2}\sum_{i=s}^{s+N-1}\sum_{j=i}^{s+N-1}\mathbb{E}\left[\mathbf{x}^\top(\mathbf{w}_i - \mathbf{w}^\star)(\mathbf{w}_j - \mathbf{w}^\star)^\top\mathbf{x}\right] + \frac{\sigma^2}{2}.\end{aligned}$$

To make progress, we invoke the tower rule:

$$\mathbb{E}\left[(\mathbf{w}_j - \mathbf{w}^\star)(\mathbf{w}_i - \mathbf{w}^\star)^\top\right] = \mathbb{E}\left[\mathbb{E}[\mathbf{w}_j - \mathbf{w}^\star \mid \mathbf{w}_i]\,(\mathbf{w}_i - \mathbf{w}^\star)^\top\right].$$

For SGD with stepsizes η_t we have:

$$\mathbb{E}[\mathbf{w}_j - \mathbf{w}^\star \mid \mathbf{w}_i] = \prod_{t=i+1}^{j}(\mathbf{I} - \eta_t\mathbf{H})(\mathbf{w}_i - \mathbf{w}^\star)$$

Since $\eta\lambda_i \leq 1$ for all λ_i we have the PSD inequality (the factors commute since they have the same $\mathbf{H}$ basis):

$$\prod_{t=i+1}^{j}(\mathbf{I} - \eta_t\mathbf{H}) \preceq \exp\left(-\mathbf{H}\sum_{t=i+1}^{j}\eta_t\right) \preceq \exp\left(-\eta[j^{1-\gamma} - i^{1-\gamma}]\mathbf{H}\right)$$

where the last inequality is from Lemma 1. Plugging this inside of the risk expression and diagonalizing we have the following risk inequality:

$$\mathcal{R}(\bar{\mathbf{w}}_{s:s+N}) - \frac{\sigma^2}{2} \lesssim \frac{1}{N^2}\sum_{i=s}^{s+N-1}\sum_{j=i}^{s+N-1}\left\langle \mathbf{m}_i,\ \exp\left(-\eta(j^{1-\gamma} - i^{1-\gamma})\mathbf{\Lambda}\right)\lambda\right\rangle \tag{5}$$

To get an expression for the bias and variance iterates, denoted as $\widetilde{\mathbf{m}}_t$ and $\overline{\mathbf{m}}_t$ respectively, we follow the derivation from Meterez et al. (2025). At batch size 1, the SGD update is

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \eta_t\mathbf{x}_t(\mathbf{x}_t^\top\mathbf{w}_{t-1} - y_t), \qquad t \geq 1, \qquad y_t = \mathbf{x}_t^\top\mathbf{w}^\star + \epsilon_t,$$

where ϵ_t is independent noise with $\mathbb{E}[\epsilon_t] = 0$ and $\mathbb{E}[\epsilon_t^2] = \sigma^2$. Subtracting $\mathbf{w}^\star$,

$$\mathbf{w}_t - \mathbf{w}^\star = (\mathbf{I} - \eta_t\mathbf{x}_t\mathbf{x}_t^\top)(\mathbf{w}_{t-1} - \mathbf{w}^\star) + \eta_t\epsilon_t\mathbf{x}_t.$$

Taking the covariance of this term we obtain a recursion on $\mathbf{\Sigma}_t$:

$$\begin{aligned}\mathbf{\Sigma}_t &\coloneqq \mathbb{E}[(\mathbf{w}_t - \mathbf{w}^\star)(\mathbf{w}_t - \mathbf{w}^\star)^\top] \\ &= \mathbb{E}\left[(\mathbf{I} - \eta_t\mathbf{x}\mathbf{x}^\top)\mathbf{\Sigma}_{t-1}(\mathbf{I} - \eta_t\mathbf{x}\mathbf{x}^\top)\right] + \eta_t^2\sigma^2\mathbb{E}[\mathbf{x}\mathbf{x}^\top].\end{aligned}$$

Taking expectation yields:

$$\mathbf{\Sigma}_t = \mathbf{\Sigma}_{t-1} - \eta_t\mathbf{\Sigma}_{t-1}\mathbf{H} - \eta_t\mathbf{H}\mathbf{\Sigma}_{t-1} + 2\eta_t^2\mathbf{H}\mathbf{\Sigma}_{t-1}\mathbf{H} + \eta_t^2\mathrm{Tr}(\mathbf{H}\mathbf{\Sigma}_{t-1})\mathbf{H} + \eta_t^2\sigma^2\mathbf{H}$$

Rotating in the $\mathbf{Q}$ basis and taking a diagonal operator through the whole equation we end up with a recursion on $\mathbf{m}_t$:

$$\begin{aligned}\mathbf{m}_t &= \left(\mathbf{I} - 2\eta_t\mathbf{\Lambda} + 2\eta_t^2\mathbf{\Lambda}^2 + \eta_t^2\lambda\lambda^\top\right)\mathbf{m}_{t-1} + \sigma^2\eta_t^2\lambda \\ &\leq \left(\mathbf{I} - 2\eta_t\mathbf{\Lambda} + \eta_t^2\mathbf{\Lambda}^2 + 2\eta_t^2\lambda\lambda^\top\right)\mathbf{m}_{t-1} + \sigma^2\eta_t^2\lambda && \mathbf{\Lambda}^2 \leq \lambda\lambda^\top \\ &\leq (\mathbf{I} - \eta_t\mathbf{\Lambda})^2\mathbf{m}_{t-1} + 2\eta_t^2\lambda(2\lambda^\top\mathbf{m}_0 + \sigma^2) + \sigma^2\eta_t^2\lambda && \text{Lemma 2} \\ &\lesssim (\mathbf{I} - \eta_t\mathbf{\Lambda})^2\mathbf{m}_{t-1} + 7\sigma^2\eta_t^2\lambda && \|\mathbf{w}_0 - \mathbf{w}^\star\|_{\mathbf{H}}^2 \lesssim \sigma^2\end{aligned}$$

For simplicity, we will absorb the constant 7 into the noise scale σ^2 for the remainder of this proof. Unrolling the recursion yields

$$\mathbf{m}_t \leq \left[\prod_{i=1}^{t}(I-\eta_i\mathbf{\Lambda})^2\right]\mathbf{m}_0 + \sum_{p=1}^{t}\eta_p^2\sigma^2\left[\prod_{r=p+1}^{t}(I-\eta_r\mathbf{\Lambda})^2\right]\lambda \tag{6}$$

$$\leq \exp\Big[-2\mathbf{\Lambda}\sum_{i=1}^{t}\eta_i\Big]\mathbf{m}_0 + \sigma^2\sum_{p=1}^{t}\eta_p^2\exp\Big[-2\mathbf{\Lambda}\sum_{r=p+1}^{t}\eta_r\Big]\lambda \tag{7}$$

Plugging in the scheduler for η_t, we have:

$$\mathbf{m}_t \lesssim \exp\big[-2\eta\mathbf{\Lambda}t^{1-\gamma}\big]\mathbf{m}_0 + \eta^2\sigma^2\sum_{p=1}^{t}\frac{1}{p^{2\gamma}}\exp\Big[-2\eta\mathbf{\Lambda}(t^{1-\gamma}-p^{1-\gamma})\Big]\lambda.$$

We therefore define the bias and variance iterates as

$$\widetilde{\mathbf{m}}_t \coloneqq \exp\big[-2\eta\mathbf{\Lambda}t^{1-\gamma}\big]\mathbf{m}_0, \tag{8}$$

$$\overline{\mathbf{m}}_t \coloneqq \eta^2\sigma^2\sum_{p=1}^{t}\frac{1}{p^{2\gamma}}\exp\Big[-2\eta\mathbf{\Lambda}(t^{1-\gamma}-p^{1-\gamma})\Big]\lambda. \tag{9}$$

To obtain the rates, we independently bound each of these quantities after plugging them back in equation 5.

Bias bound. We begin with the bias:

$$\begin{aligned}
\texttt{bias}_t &\lesssim \sum_k \frac{\mathbf{m}_{0,k}\lambda_k}{N^2}\sum_{i=s}^{s+N-1}\sum_{j=i}^{s+N-1}\exp\Big[-\eta(j^{1-\gamma}-i^{1-\gamma})\lambda_k - 2\eta\lambda_k i^{1-\gamma}\Big] \\
&\leq \sum_k \frac{\mathbf{m}_{0,k}\lambda_k}{N^2}\sum_{i=s}^{s+N-1}\sum_{j=i}^{s+N-1}\exp\Big[-\eta(j^{1-\gamma}-i^{1-\gamma})\lambda_k - \eta\lambda_k i^{1-\gamma}\Big] \\
&= \sum_k \frac{\mathbf{m}_{0,k}\lambda_k}{N^2}\sum_{i=s}^{s+N-1}\sum_{j=i}^{s+N-1}\exp\big[-\eta\lambda_k j^{1-\gamma}\big].
\end{aligned}$$

Choose

$$k^\star \coloneqq \max\Big\{k:\ \lambda_k \geq \tfrac{\log N}{\eta s^{1-\gamma}}\Big\},$$

which defines the split between head and tail eigenvalues.

$$\begin{aligned}
\texttt{bias}_t^{1:k^\star} &\lesssim \sum_{k\leq k^\star}\frac{\mathbf{m}_{0,k}\lambda_k}{N^2}\sum_{i=s}^{s+N-1}\sum_{j=i}^{s+N-1}\exp\big[-\eta\lambda_k j^{1-\gamma}\big] \\
&\leq \sum_{k\leq k^\star}\frac{\mathbf{m}_{0,k}\lambda_k}{N^2}\sum_{i=s}^{s+N-1}\sum_{j=i}^{s+N-1}\exp\Big[-\Big(\tfrac{j}{s}\Big)^{1-\gamma}\log N\Big] \\
&\lesssim \sum_{k\leq k^\star}\frac{\mathbf{m}_{0,k}\lambda_k}{N^2}\sum_{i=s}^{s+N-1}\sum_{j=i}^{s+N-1}N^{-1} \\
&\lesssim \sum_{k\leq k^\star}\frac{\mathbf{m}_{0,k}\lambda_k}{N}.
\end{aligned}$$

For the tail eigenvalues, we upper bound the exponential by 1, yielding

$$\texttt{bias}_t^{k^\star:\infty} \lesssim \sum_{k>k^\star}\mathbf{m}_{0,k}\lambda_k.$$

Setting $\mathbf{w}_0 = 0$, we have the final bias bound:

$$\texttt{bias}_t \lesssim \frac{1}{N}\|\mathbf{w}^\star\|^2_{\mathbf{\Lambda}_{1:k^\star}} + \|\mathbf{w}^\star\|^2_{\mathbf{\Lambda}_{k^\star:\infty}} \tag{10}$$

Variance bound. We now turn our attention to bounding the variance. Following a similar procedure as before, we have:

$$\begin{aligned}
\texttt{variance}_t &\lesssim \sum_k \frac{\eta^2\sigma^2\lambda_k^2}{N^2} \sum_{i=s}^{s+N-1} \sum_{j=i}^{s+N-1} \sum_{p=1}^{i} \frac{1}{p^{2\gamma}} \\
&\qquad \times \exp\left[-2\eta\lambda_k(i^{1-\gamma} - p^{1-\gamma}) - \eta\lambda_k(j^{1-\gamma} - i^{1-\gamma})\right] \\
&\leq \sum_k \frac{\eta^2\sigma^2\lambda_k^2}{N^2} \sum_{i=s}^{s+N-1} \sum_{j=i}^{s+N-1} \sum_{p=1}^{i} \frac{1}{p^{2\gamma}} \\
&\qquad \times \exp\left[-\eta\lambda_k(j^{1-\gamma} - p^{1-\gamma})\right] \\
&= \sum_k \frac{\eta^2\sigma^2\lambda_k^2}{N^2} \sum_{i=s}^{s+N-1} \sum_{j=i}^{s+N-1} \sum_{p=1}^{i} \frac{1}{p^{2\gamma}} \\
&\qquad \times \exp\left[-\eta\lambda_k(j^{1-\gamma} - i^{1-\gamma} + i^{1-\gamma} - p^{1-\gamma})\right] \\
&= \sum_k \frac{\eta^2\sigma^2\lambda_k^2}{N^2} \sum_{i=s}^{s+N-1} \sum_{j=i}^{s+N-1} \\
&\qquad \times \exp\left[-\eta\lambda_k(j^{1-\gamma} - i^{1-\gamma})\right] \sum_{p=1}^{i} \frac{1}{p^{2\gamma}} \\
&\qquad \times \exp\left[-\eta\lambda_k(i^{1-\gamma} - p^{1-\gamma})\right]
\end{aligned}$$

Now we can handle the last sum. We split it at $i/2$ and handle each sum independently. We begin with the first half:

$$\begin{aligned}
\sum_{p=1}^{i/2} \frac{1}{p^{2\gamma}} \exp\left[-\eta\lambda_k(i^{1-\gamma} - p^{1-\gamma})\right] &\lesssim \sum_{p=1}^{i/2} \frac{1}{p^{2\gamma}} \exp\left[-\eta\lambda_k i^{-\gamma}(i-p)\right] && \text{Lemma 1} \\
&\lesssim \exp\left(-\frac{1}{2}\eta\lambda_k i^{1-\gamma}\right) \sum_{p=1}^{i/2} \frac{1}{p^{2\gamma}} \\
&\lesssim \exp\left(-\frac{1}{2}\eta\lambda_k i^{1-\gamma}\right) \cdot \begin{cases} i^{1-2\gamma} & \gamma < 1/2 \\ \log i & \gamma = 1/2 \end{cases}
\end{aligned}$$

Now we look at the second half. We have:

$$\begin{aligned}
\sum_{p=i/2}^{i} \frac{1}{p^{2\gamma}} \exp\left[-\eta\lambda_k(i^{1-\gamma} - p^{1-\gamma})\right] &\lesssim i^{-2\gamma} \exp\left(-\eta\lambda_k i^{1-\gamma}\right) \sum_{p=i/2}^{i} \exp\left(\eta\lambda_k i^{-\gamma} p\right) \\
&\lesssim \frac{i^{-\gamma}}{\eta\lambda_k}\left[1 - \exp\left(-\frac{1}{2}\eta\lambda_k i^{1-\gamma}\right)\right]
\end{aligned}$$

Now that we have obtained the bounds for the sum over p, remains to bound the sum over j:

$$\begin{aligned}\sum_{j=i}^{s+N-1} \exp\left[-\eta\lambda_k(j^{1-\gamma}-i^{1-\gamma})\right] &\lesssim \sum_{j=i}^{s+N-1} \exp\left[-\eta\lambda_k j^{-\gamma}(j-i)\right] && \text{Lemma 1}\\ &\lesssim \sum_{j=i}^{s+N-1} \exp\left[-\eta\lambda_k (s+N)^{-\gamma}(j-i)\right]\\ &\le \sum_{q=0}^{N-1} \exp\left[-\eta\lambda_k (s+N)^{-\gamma} q\right]\\ &\lesssim \frac{(s+N)^\gamma}{\eta\lambda_k}\left(1-\exp\left[-\eta\lambda_k \frac{N}{(s+N)^\gamma}\right]\right)\end{aligned}$$

Assembling everything together we end up with:

$$\begin{aligned}\texttt{variance}_t &\lesssim \sum_k \frac{\eta^2\sigma^2\lambda_k^2}{N^2} \sum_{i=s}^{s+N-1}\sum_{j=i}^{s+N-1}\\ &\quad\times \exp\left[-\eta\lambda_k\left(j^{1-\gamma}-i^{1-\gamma}\right)\right]\sum_{p=1}^{i}\frac{1}{p^{2\gamma}}\exp\left[-\eta\lambda_k\left(i^{1-\gamma}-p^{1-\gamma}\right)\right]\\ &\lesssim \sum_k \frac{\eta^2\sigma^2\lambda_k^2}{N^2}\sum_{i=s}^{s+N-1}\frac{(s+N)^\gamma}{\eta\lambda_k}\left(1-\exp\left[-\eta\lambda_k\frac{N}{(s+N)^\gamma}\right]\right)\\ &\quad\times\left[\exp\left(-\frac{1}{2}\eta\lambda_k i^{1-\gamma}\right)s_i(\gamma)\right.\\ &\quad\left.+\frac{i^{-\gamma}}{\eta\lambda_k}\left(1-\exp\left(-\frac{1}{2}\eta\lambda_k i^{1-\gamma}\right)\right)\right].\end{aligned}$$

where we have defined:

$$s_i(\gamma)=\begin{cases} i^{1-2\gamma} & \gamma<1/2\\ \log i & \gamma=1/2\end{cases}$$

Recall the cutoff

$$k^\star := \max\left\{k:\ \lambda_k \ge \tfrac{\log N}{\eta s^{1-\gamma}}\right\}.$$

We split the sum over k into head and tail components.

Variance head. For $k\le k^\star$ and all $i\ge s$, we have

$$\eta\lambda_k i^{1-\gamma}\ge \eta\lambda_k s^{1-\gamma}\ge \log N,$$

so the exponential term in the p-sum decays rapidly. Consequently, the sum over p is dominated by its second half, yielding the bound

$$\sum_{p=1}^{i}\frac{1}{p^{2\gamma}}\exp\Big[-\eta\lambda_k(i^{1-\gamma}-p^{1-\gamma})\Big]\ \lesssim\ \frac{i^{-\gamma}}{\eta\lambda_k}.$$

Moreover, $1-\exp\Big[-\eta\lambda_k\frac{N}{(s+N)^\gamma}\Big]\le 1$. Thus, we get:

$$\begin{aligned}\texttt{variance}_t^{1:k^\star} &\lesssim \sum_{k\le k^\star}\frac{\eta^2\sigma^2\lambda_k^2}{N^2}\sum_{i=s}^{s+N-1}\frac{(s+N)^\gamma}{\eta\lambda_k}\cdot\frac{i^{-\gamma}}{\eta\lambda_k}\\ &= \sigma^2\sum_{k\le k^\star}\frac{(s+N)^\gamma}{N^2}\sum_{i=s}^{s+N-1} i^{-\gamma}.\end{aligned}$$

Using $\sum_{i=s}^{s+N-1} i^{-\gamma} \lesssim N s^{-\gamma}$, we conclude

$$\texttt{variance}_t^{1:k^\star} \lesssim \sigma^2 \frac{(s+N)^\gamma}{N s^\gamma} k^\star$$

Variance tail. For $k > k^\star$, we upper bound all exponentials by 1 and use $1 - e^{-x} \leq x$ with $x = \eta\lambda_k \frac{N}{(s+N)^\gamma}$. Starting from the assembled bound,

$$\texttt{variance}_t \lesssim \sum_k \frac{\eta^2\sigma^2\lambda_k^2}{N^2} \sum_{i=s}^{s+N-1} \frac{(s+N)^\gamma}{\eta\lambda_k}\Big(1 - \exp\Big[-\eta\lambda_k \tfrac{N}{(s+N)^\gamma}\Big]\Big) \Big[\exp\left(-\tfrac{1}{2}\eta\lambda_k i^{1-\gamma}\right) s_i(\gamma) + \frac{i^{-\gamma}}{\eta\lambda_k}\left[1 - \exp\left(-\tfrac{1}{2}\eta\lambda_k i^{1-\gamma}\right)\right]\Big],$$

we obtain, for $k > k^\star$,

$$\begin{aligned}
\texttt{variance}_t^{k^\star:\infty} &\lesssim \sum_{k>k^\star} \frac{\eta^2\sigma^2\lambda_k^2}{N^2} \sum_{i=s}^{s+N-1} \frac{(s+N)^\gamma}{\eta\lambda_k}\Big(\eta\lambda_k \tfrac{N}{(s+N)^\gamma}\Big)\left[s_i(\gamma) + \frac{i^{-\gamma}}{\eta\lambda_k}\right] \\
&= \sum_{k>k^\star} \frac{\eta^2\sigma^2\lambda_k^2}{N} \sum_{i=s}^{s+N-1} \left[s_i(\gamma) + \frac{i^{-\gamma}}{\eta\lambda_k}\right] \\
&= \sigma^2 \sum_{k>k^\star} \left(\eta^2\lambda_k^2 \sum_{i=s}^{s+N-1} s_i(\gamma) + \eta\lambda_k \sum_{i=s}^{s+N-1} i^{-\gamma}\right) \frac{1}{N}.
\end{aligned}$$

Bounding the remaining sums as $\frac{1}{N}\sum_{i=s}^{s+N-1} i^{-\gamma} \lesssim s^{-\gamma}$ and $\frac{1}{N}\sum_{i=s}^{s+N-1} s_i(\gamma) \lesssim s^{1-2\gamma}$ (for $\gamma = \frac{1}{2}$ this incurs an additional $\log N$ factor), we conclude

$$\texttt{variance}_t^{k^\star:\infty} \lesssim \sigma^2 \sum_{k>k^\star} \left(\eta^2\lambda_k^2\, s^{1-2\gamma} + \eta\lambda_k\, s^{-\gamma}\right) \tag{11}$$

Combining the 2 bounds we get the final variance bound:

$$\texttt{variance}_t \lesssim \sigma^2 \frac{(s+N)^\gamma}{N s^\gamma} k^\star + \sigma^2 \sum_{k>k^\star} \left(\eta^2\lambda_k^2\, s^{1-2\gamma} + \eta\lambda_k\, s^{-\gamma}\right) \tag{12}$$

□

B.2 Proof of Corollary 1

Proof of Corollary 1. From Theorem 1 we know that for $s = \Theta(N)$, for $k^\star := \max\left\{k : \lambda_k \geq \frac{\log N}{\eta N^{1-\gamma}}\right\}$ the bias and variance decay rates are:

$$\begin{aligned}
\texttt{bias}_t &\lesssim \frac{1}{N}\|\mathbf{w}^\star\|^2_{\mathbf{\Lambda}_{1:k^\star}} + \|\mathbf{w}^\star\|^2_{\mathbf{\Lambda}_{k^\star:\infty}} \\
\texttt{variance}_t &\lesssim \frac{\sigma^2 k^\star}{N} + \sigma^2 \sum_{k>k^\star} \left(\eta^2\lambda_k^2\, N^{1-2\gamma} + \eta\lambda_k\, N^{-\gamma}\right)
\end{aligned}$$

We first specialize the bias bound to the power law setting. Plugging in the source and capacity assumptions, we have:

$$\begin{aligned}
\|\mathbf{w}^\star\|^2_{\mathbf{\Lambda}_{1:k^\star}} &\leq \sum_{k\leq k^\star} \lambda_k (w_k^\star)^2 \lesssim \sum_{k\leq k^\star} k^{-b} \lesssim \sum_{k=1}^{\infty} k^{-b} \eqsim 1 \qquad b > 1 \\
\|\mathbf{w}^\star\|^2_{\mathbf{\Lambda}_{k^\star:\infty}} &\lesssim \sum_{k>k^\star} k^{-b} \leq \int_{k^\star}^{\infty} k^{-b} dk \lesssim (k^\star)^{1-b}
\end{aligned}$$

For the variance term, we can see that the variance head is the dominating term. By definition, we have $\lambda_k \eqsim k^{-a}$ with $a > 1$. Then:

$$\sum_{k>k^\star} \lambda_k \lesssim (k^\star)^{1-a}, \qquad \sum_{k>k^\star} \lambda_k^2 \lesssim (k^\star)^{1-2a}.$$

Moreover, by definition of $k^\star$ we have $\lambda_{k^\star} \eqsim (k^\star)^{-a} \eqsim \frac{\log N}{\eta N^{1-\gamma}}$, where the $\eqsim$ notation absorbs constants and $\log$ factors. Thus, for the 2 terms in the variance tail, we have:

$$\sigma^2 \eta N^{-\gamma} \sum_{k>k^\star} \lambda_k \lesssim \sigma^2 \eta N^{-\gamma} (k^\star)^{1-a} = \frac{\sigma^2 k^\star}{N} \left(\eta N^{1-\gamma} (k^\star)^{-a}\right) \lesssim \frac{\sigma^2 k^\star}{N} \log N \lesssim \frac{\sigma^2 k^\star}{N}$$

$$\sigma^2 \eta^2 N^{1-2\gamma} \sum_{k>k^\star} \lambda_k^2 \lesssim \sigma^2 \eta^2 N^{1-2\gamma} (k^\star)^{1-2a} = \frac{\sigma^2 k^\star}{N} \left(\eta N^{1-\gamma} (k^\star)^{-a}\right)^2 \lesssim \frac{\sigma^2 k^\star}{N} \log^2 N \lesssim \frac{\sigma^2 k^\star}{N}$$

where we absorbed the $\log$ factors into the $\lesssim$ notation. Plugging everything in, we have the bound on the risk:

$$\mathbb{E}\mathcal{R}(\bar{\mathbf{w}}_{s:s+N}) - \frac{\sigma^2}{2} \lesssim \frac{1}{N} + (k^\star)^{1-b} + \frac{\sigma^2 k^\star}{N}$$

Optimizing the bound over γ is equivalent to optimizing over $k^\star$, since $k^\star$ is a function of γ. We can optimize up to constants by balancing the $k^\star$ dependent terms in the bound:

$$(k^\star)^{1-b} \eqsim \frac{\sigma^2 k^\star}{N} \implies (k^\star)^b \eqsim \frac{N}{\sigma^2} \implies k^\star \eqsim N^{1/b}$$

Combining with the definition of $k^\star$ we have $\lambda_{k^\star} \eqsim (k^\star)^{-a} \eqsim \frac{\log N}{\eta N^{1-\gamma}}$. Plugging in the expression for $k^\star$ we have and dropping the $\log$ factor and the constant η we have:

$$N^{(1-\gamma)/a} \eqsim k^\star \eqsim N^{1/b} \implies \gamma^\star = 1 - \frac{a}{b} \in (0, \frac{1}{2}]$$

for $1 < a < b \leq 2a$. Thus the optimal exponent is $\gamma^\star$ for the rate. This yields:

$$(k^\star)^{1-b} \eqsim N^{-(b-1)/b}$$

Since $b > 1$, we have that $1/N \leq N^{-(b-1)/b}$, and thus the final rate is:

$$\mathbb{E}\mathcal{R}(\bar{\mathbf{w}}_{s:s+N}) - \frac{\sigma^2}{2} \lesssim N^{-(b-1)/b}$$

□

C Additional Figures

C.1 Figure 2 at optimal hyperparameters

We provide the losses of Figure 2 at the optimal hyperparameters for every intermediate point. Note that this is not a single run - we plot the optimal loss (over our sweep) at each intermediate point from $1\times$ to $32\times$ (and $16\times$ respectively for the 300M model) Chinchilla and interpolate between the points.

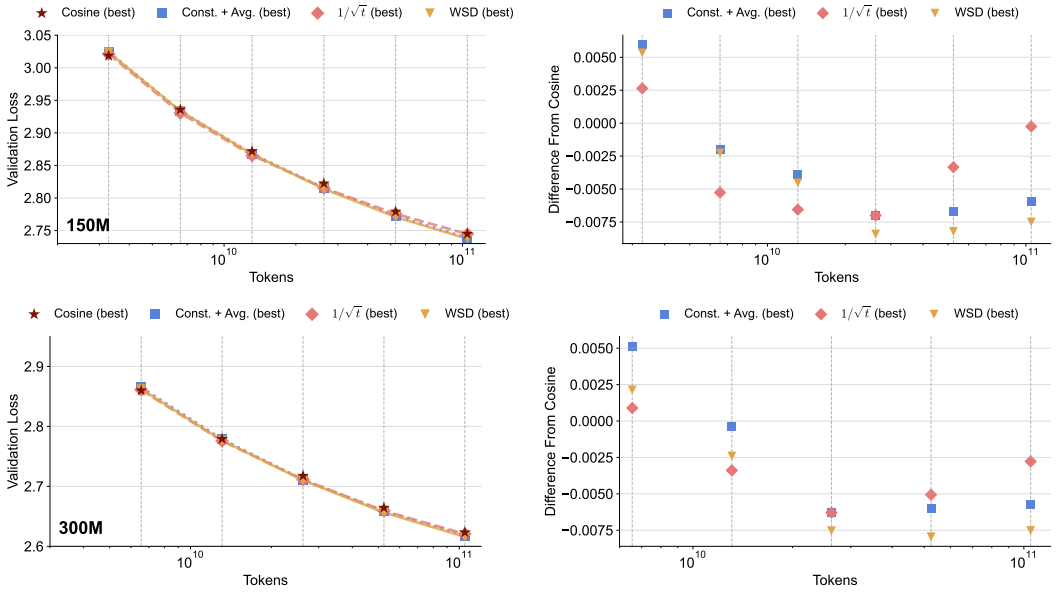


Figure 6: Top plots correspond to 150M parameter, and bottom plots correspond to 300M models. At each intermediate point, we plot the best loss out of the whole hyperparameter sweep, then we linearly interpolate between the points. Note that for long training durations, constant with averaging and $1/\sqrt{t}$ offers a substantial improvement over cosine decay.

C.2 Comparison at Optimal HPs

C.2.1 Optimal Hyperparameters by Endpoint

	Constant LR + Avg.		$1/\sqrt{t}$ + Avg.		
Endpoint	η	β_2	η	β_2	α
$1\times$	0.003	0.98	0.003	0.99	12,800
$2\times$	0.003	0.98	0.003	0.99	12,800
$4\times$	0.003	0.98	0.003	0.98	51,200
$8\times$	0.001	0.99	0.003	0.99	51,200
$16\times$	0.001	0.99	0.001	0.99	51,200
$32\times$	0.001	0.99	0.001	0.99	51,200

Table 3: Per-endpoint optimal hyperparameters for the 150M plots, where optimal means the lowest 4% EMA validation loss among the configurations ran at each checkpoint.

	Constant LR + Avg.		$1/\sqrt{t}$ + Avg.		
Endpoint	η	β_2	η	β_2	α
$1\times$	0.003	0.95	0.003	0.95	12,800
$2\times$	0.003	0.95	0.003	0.95	12,800
$4\times$	0.001	0.99	0.001	0.99	51,200
$8\times$	0.001	0.99	0.001	0.99	51,200
$16\times$	0.001	0.99	0.001	0.99	51,200

Table 4: Per-endpoint optimal hyperparameters for the 300M plots, where optimal means the lowest 4% EMA validation loss among the configurations ran at each checkpoint.

C.2.2 Inverse-Square-Root Offset Sensitivity

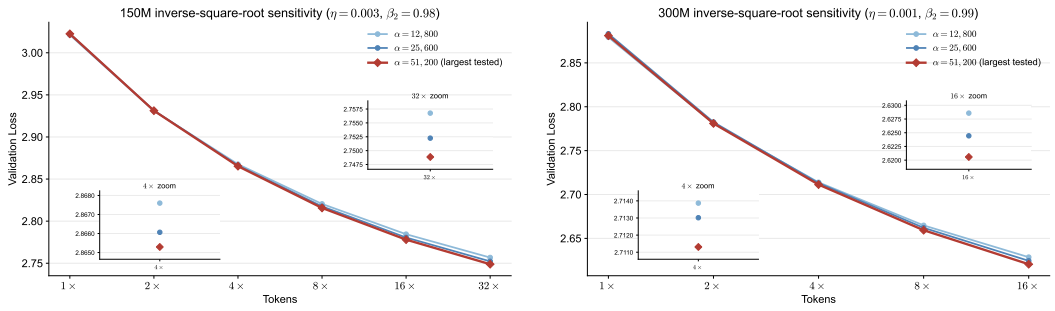


Figure 7: Sensitivity of the inverse-square-root offset α with every other hyperparameter fixed to the configuration selected at $4\times$. Left: the 150M model with $(\eta, \beta_2) = (0.003, 0.98)$, evaluated from $1\times$ through $32\times$. Right: the 300M model with $(\eta, \beta_2) = (0.001, 0.99)$, evaluated from $1\times$ through $16\times$. Lines connect exact checkpoint losses for $\alpha \in \{12{,}800, 25{,}600, 51{,}200\}$.

C.3 FIGURE 5 AT OPTIMAL HYPERPARAMETERS

We similarly provide the losses from the experiment in Figure 5 for the optimal hyperparameters at each intermediate point.

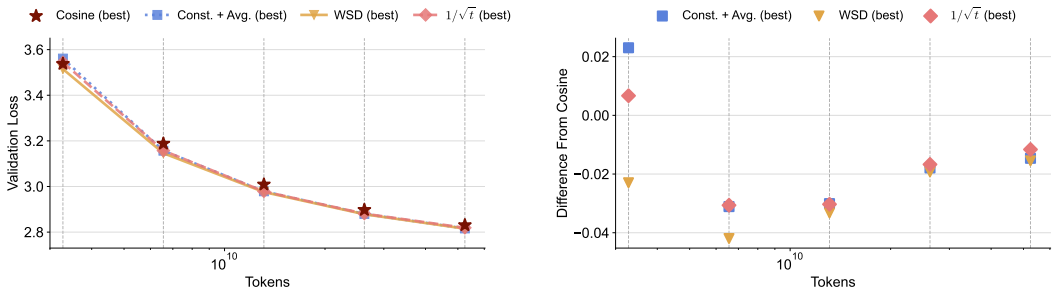


Figure 8: (Left) Validation loss comparison between cosine decay, constant learning rate with averaging, $1/\sqrt{t}$ with averaging and WSD on 150M models trained on $1\times$ to $16\times$ Chinchilla size datasets, at batch size 4096 with a 500-step warmup. (Right) The difference between the loss achieved by each schedule and cosine at each multiple of Chinchilla, with negative values meaning better than cosine. Traces are made by interpolating between the losses of the optimal hyperparameter runs at each of the intermediate points from $1\times$ to $16\times$ Chinchilla.

C.4 SYNTHETIC SENSITIVITY TO LABEL NOISE

Figure 9 repeats the exact-recursion experiment from Figure 3 at a lower label-noise variance. This isolates the expected finite-horizon effect of the noise scale: reducing σ^2 delays the variance-dominated regime in which the faster asymptotic decay of $1/\sqrt{t}$ becomes visible.

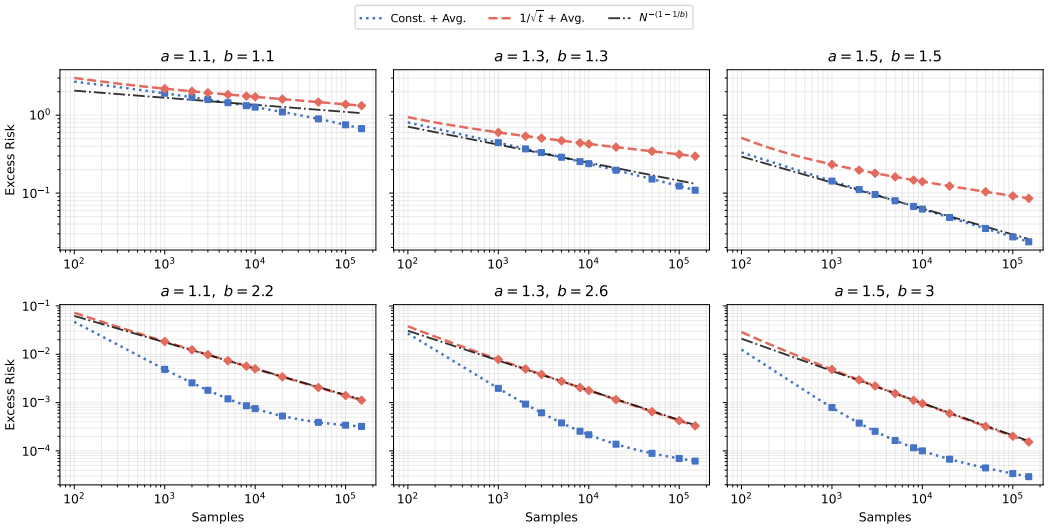
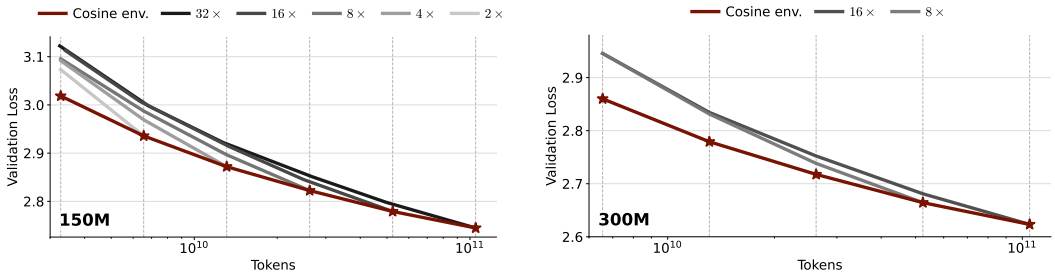


Figure 9: Similar setup as Figure 3, but with $\sigma^2 = 0.01$.

C.5 Optimal cosine envelope

Figure 10 compares two ways of using cosine learning-rate schedules across token budgets from $1\times$ to $32\times$ Chinchilla. First, we train *separately tuned* cosine baselines for each budget and connect the best result at every horizon; this defines the *optimal cosine envelope* (red curve). Second, we tune a cosine schedule for a long run ($8\times$, $16\times$, or $32\times$) and evaluate it at earlier checkpoints along the same training trajectory. Across both model sizes, these intermediate points lie substantially above the optimal envelope, showing that a cosine schedule tuned for a long horizon does not transfer well to shorter budgets.

Note that we plot again Figure 1 to aid in side by side comparison.



(a) 150M (b) 300M

Figure 10: Comparison between the optimal cosine envelope and checkpoints from cosine schedules tuned for longer budgets. The optimal envelope (red) is formed by independently tuning a cosine decay for each training horizon ($1\times$–$32\times$ Chinchilla) and taking the best validation value at that horizon. We also plot cosine schedules tuned for $8\times$, $16\times$, and $32\times$ and evaluated at smaller budgets using intermediate checkpoints from the same run. The gap to the envelope is substantial, indicating limited transfer from long-horizon cosine tuning to shorter budgets. In the 150M experiment, checkpoints were not logged exactly at $1\times$–$32\times$, and we plot the closest recorded validation point.