Anytime Pretraining: Horizon-Free Learning-Rate Schedules with Weight Averaging

author names withheld

Under Review for the Workshop on High-dimensional Learning Dynamics, 2026

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 at 1–32× Chinchilla scale, comparing constant learning rates with weight averaging 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

Anytime learning rate schedules such as $1/t^\gamma$ for $\gamma < 1$ have been studied in the literature [22], however unless weight averaging is used these schedules generally do not achieve minimax optimal rates for SGD in linear regression [39, 53, 56]. In this work we investigate, theoretically and empirically, how $1/t^\gamma$, constant learning rate and WSD compare to cosine decay in long training runs, and which of these schedules provides a viable anytime, or almost-anytime, alternative to cosine annealing. Concretely, we require two properties from an anytime scheduler: (i) the schedule should not depend on the planned number of training steps, 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.

Figure 1 highlights why this is a nontrivial requirement: cosine schedules tuned for a single terminal horizon are far from optimal when evaluated at intermediate checkpoints. Put differently, standard training recipes without knowing the stopping time do not yield an anytime procedure, because the choice of horizon implicitly determines the entire trajectory of losses. To the best of our knowledge, this envelope perspective has not been explicitly studied or used as an evaluation target in prior work. This motivates the central question of this paper: *when can a horizon-free (or nearly horizon-free) training procedure match—or even improve upon—the cosine envelope across training time?* Our goal is to propose alternatives that are competitive with a single cosine run at

© .

a fixed endpoint, and to characterize (theoretically and empirically) when matching the envelope is achievable. We show that simple anytime schedules such as constant or $1/t^\gamma$ with appropriate averaging, can closely follow the envelope over long runs.

1.1. Main Contributions

We first state an informal version of our main theoretical result:

Theorem *[Informal version of Theorem 1] For an SGD process run on N samples, a polynomially decaying learning rate of the form $\eta_t = 1/t^\gamma$ with tail averaging matches the rates of well-tuned SGD with averaging, where $0 < \gamma < 1$ and the exponent γ depends on the spectral properties of the data.*

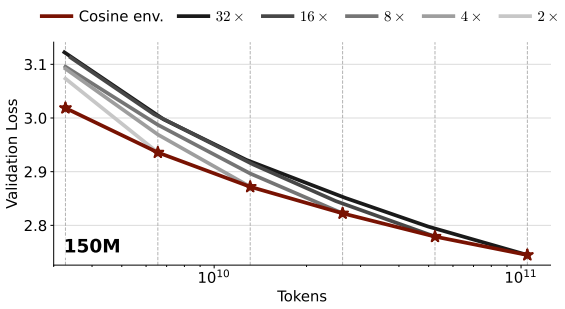


Figure 1: 150M model: **Cosine schedules do not transfer across horizons.** The *cosine envelope* (red) is formed by independently tuning a horizon-aware cosine schedule for each terminal compute budget ($1\times$–$32\times$ Chinchilla) and taking the best validation loss at that horizon. Gray curves show the same cosine schedule evaluated at intermediate checkpoints when tuned for a single fixed terminal budget. An analogous plot for the 300M model appears in Figure 9 (Appendix D).

This result shows that an anytime learning-rate schedule can achieve the same rate as well-tuned SGD. In contrast, Zhang et al. [68] show that while a constant learning rate with weight averaging can also attain these rates, for certain source and capacity exponents the learning rate must be scaled as a function of the training horizon (i.e., it depends on the end time) to achieve minimax rates, and is therefore not an anytime scheme. Guided by our theoretical analysis, we empirically compare three anytime schemes—$1/\sqrt{t}$, WSD, and a constant learning rate with weight averaging—against cosine decay. We train 150M- and 300M-parameter models at power-of-two multiples of the Chinchilla compute budget: from $1\times$ to $32\times$ for 150M, and from $1\times$ to $16\times$ for 300M. For cosine decay, each model is trained separately at each compute budget, whereas the anytime methods are trained once at the largest Chinchilla multiple and evaluated at intermediate checkpoints, as shown in Figure 2. Across all intermediate points, including very long training regimes, the anytime methods closely match cosine annealing, paying only a negligible performance hit near the start and end of training.

2. Empirical Findings

Empirically, we can see from Figure 2 that anytime schedules such as $1/\sqrt{t}$ with tail averaging, and constant learning rate with tail averaging can provide anytime alterantives to cosine in a horizon independent manner. We also compare with WSD, a commonly used alternative to cosine. To reiterate, we implement WSD following Hu et al. [28], Team et al. [59], doing warmup for 40% of the $1\times$ Chinchilla duration of the respective training run, followed by a constant learning rate until

90% of the run, then implementing a linear decay over the last 10% of the run, decaying the learning rate to 10% of its original value. In practice, we have saved at 90% of each intermediate point for the constant learning rate experiments, then decayed from each, to ensure a fair comparison. We emphasize that WSD is not strictly horizon-free, as it relies on checkpointing and a subsequent decision of when to initiate the decay stage. It is important to note that a cosine run tuned for a long duration will far underperform at smaller compute budgets - we ellaborate upon this point in Figure 9 (Appendix D). To further our understanding, we provide a theoretical analysis of WSD in Section 3 in the power law linear regression setting, showing that it achieves a similar rate as a constant learning rate with stochastic weight averaging [68]. We provide large batch ablations in Appendix A. We provide further experimental details in Section F.

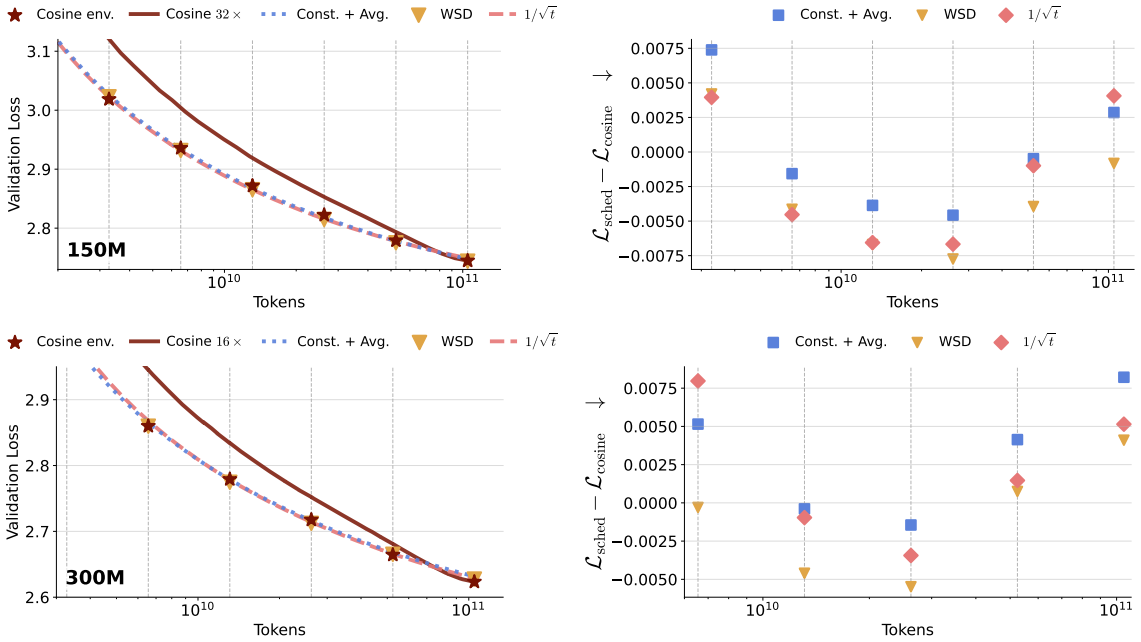


Figure 2: Validation loss for 150M (top) and 300M (bottom) models. **Left:** cosine decay versus constant LR, WSD, and $\sqrt{\alpha/(t+\alpha)}$ schedules with weight averaging across Chinchilla multiples. Cosine baselines are tuned separately at each horizon, while anytime schedules use a *single long run*. Red stars show the cosine envelope. **Right:** loss relative to the cosine envelope, with negative values better than cosine.

3. Theoretical Analysis

Setup and notation. Throughout the theory section of the manuscript, we study linear regression on Gaussian data over N total samples and batch size 1, with 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 $0 \leq i \leq j \leq \infty$ as $\mathbf{\Lambda}_{i:j} = \text{diag}(\lambda_i, \ldots, \lambda_j)$. We denote the independent covariates as $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ where each $\mathbf{x} \in \mathbb{R}^d$ and $y \in \mathbb{R}$ such that $\mathbf{x} \sim \mathcal{N}(0, \mathbf{H})$ and $y|\mathbf{x} \sim \mathcal{N}(\mathbf{x}^\top \mathbf{w}^\star, \sigma^2)$ 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 by η_t and in general $\eta_t \coloneqq \eta \cdot f(t, N)$ 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 also denote $\bar{\mathbf{w}}_{s:s+T} \coloneqq \frac{1}{T}\sum_{i=s}^{s+T-1} \mathbf{w}_i$ to be the average of the last T iterates starting from step s. Without loss of generality, we will always consider $\mathbf{w}_0 = 0$. The main proof technique used to establish the rates is based on the bias-variance decomposition of the risk [31, 33, 43, 64, 65, 69]. All the proofs are deferred to Appendix C and we provide synthetic regression plots in E.

3.1. Main results

The main assumption used in deriving our risk rate states that at any time t, the risk of the SGD training process is at most a constant factor larger than the noise. Formally, this means the following:

Assumption 1 *There exists a constant $c > 1$ such that for any time $t > 0$ we have $\mathcal{R}(\mathbf{w}_t) \leq c\sigma^2$.*

Assumption 1 is a mild assumption, since in general we expect a scheduler to only start decaying the learning rate once the risk is variance dominated. 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 $\mathbf{\Lambda} = \text{diag}(\lambda_i)_{i\geq 1}$ be the eigenvalues of $\mathbf{H}$ and learning rate $\eta_t = \eta/t^\gamma$ for $\gamma \in (0, 1)$ and some constant $\eta \lesssim 1/\text{Tr}(\mathbf{H})$ independent of N. Under assumption 1, we have:*

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

for $k^\star \coloneqq \max\left\{k : \lambda_k \geq \frac{\log N}{\eta N^{1-\gamma}}\right\}$, where $\lesssim$ absorbs absolute constant and log *factors.*

Following Jain et al. [31], 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 Corollary 2, we will specialize the covariance spectrum to a power law setting, using source and capacity exponents [11, 13, 41, 50], and compute the optimal choice of gamma as a function of the source and capacity.

Corollary 2 *Consider the setting of Theorem 1. We call $a, b > 1$ capacity and source exponents such that $\lambda_i \eqsim i^{-a}$ and $\mathbb{E}\lambda_i(\mathbf{w}_i^\star)^2 \eqsim i^{-b}$. Then, we have that the optimal choice for γ is $\gamma^\star = \max\left\{1 - \frac{a}{b}, 0\right\}$ Moreover, for $\gamma = \gamma^\star$ we have:*

$$\mathcal{R}(\mathbf{w}) - \sigma^2 \lesssim \left(\frac{\sigma^2}{N}\right)^{1-\frac{1}{b}}$$

where $\lesssim$ absorbs absolute constant and log *factors*

From Corollary 2 we see that the optimal $\gamma^\star$ depends on the spectral properties of the data, with the necessary condition that $a < b$. We can interpret b as quantifying how the signal in the target vector $\mathbf{w}^\star$ is spread in the eigenvectors of the data covariance $\mathbf{H}$, with higher b values meaning that most of the signal is located in the top few directions of $\mathbf{H}$. If $b \geq a$, the rate established achieves the infinite dimensional minimax optimal rate established by Zhang et al. [67]. Intuitively, for $b \gg a > 1$, the effective dimension of the space is low, and we are close to the strongly-convex case, and we see that the optimal scheduler would be $1/t$ with averaging, in line with existing literature [18, 39, 53, 56]. Conversely, for $b \leq a$, most of the signal is in the tail direction and thus a large learning rate is needed in order to fully decay the bias. Matching this intuition, from Corollary 2 we see that a constant learning rate with averaging would be the optimal scheduler, recovering the results of Zhang et al. [68], Zou et al. [69].

WSD. WSD learning rate schedules have shown great promise empirically, being adopted by frontier labs [59]. Recall that the WSD schedule, as introduced by Hu et al. [28], consists of a warmup stage, a constant learning rate stage and a decay over the last 10% of the training run.

Theorem 3 *Consider $t_0 = \rho N$ for some constant $\rho \in (0, 1)$. Assume a power law spectrum on $\mathbf{H}$, with capacity exponent $a \in (1, 2)$ and source exponent $b > 1$ (defined as in Corollary 2). Consider the two-phase learning rate schedule:*

$$\eta_t = \begin{cases} \eta & 1 \leq t \leq t_0, \\ \eta\left(1 - \frac{t - t_0}{N - t_0}\right) & t_0 < t \leq N \end{cases}$$

for constant $\eta \lesssim 1/Tr(\mathbf{H})$ independent of N. Under Assumption 1, we have the excess risk bound:

$$\mathcal{R}(\mathbf{w}_N) - \sigma^2 \lesssim \left(\frac{1}{N}\right)^{\frac{b}{a} - \frac{1}{a}} + \sigma^2 \left(\frac{1}{N}\right)^{1 - \frac{1}{a}}$$

where $\lesssim$ absorbs absolute constant and $\log$ *factors.*

In particular, if $b > a$, the variance term dominates, since most of the signal is contained in the top eigendirections, thus we can fit the bias in a finite number of steps. If $b < a$, then bias term dominates with most of the signal being contained in the bottom eigendirections and requiring a larger learning rate in order to fit them. For $b = a$, both terms decay with exponent $1 - 1/b$ (up to logarithmic factors), which matches the rate obtained in Corollary 2.

4. Discussion and Conclusions

We study horizon-free learning-rate schedules for LLM pretraining, focusing on constant and $1/\sqrt{t}$ step sizes combined with weight averaging. Empirically, these anytime schedules closely match well-tuned cosine baselines across 150M and 300M models over long Chinchilla-scale training horizons. Theoretically, we show that polynomially decaying step sizes with averaging achieve optimal rates for linear regression under power-law spectra, with the optimal decay exponent determined by source and capacity conditions. We also analyze WSD, showing that it has comparable asymptotic behavior in the same quadratic setting, although it remains only nearly anytime because it requires choosing a decay point. Overall, our results suggest that simple horizon-free schedules with averaging are a practical alternative to horizon-dependent cosine decay for continual or open-ended pretraining.

References

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

[2] 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.

[3] 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.

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

[5] 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.

[6] 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.

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

[8] 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.

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

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

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

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

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

[14] 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.

[15] 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.

[16] 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.

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

[18] 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.

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

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

[21] 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*, pages 728–763. PMLR, 2015.

[22] 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.

[23] 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)*, pages 15789–15809, 2024.

[24] 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.

[25] 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.

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

[27] 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.

[28] 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.

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

[30] 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.

[31] 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.

[32] Prateek Jain, Sham M Kakade, Rahul Kidambi, Praneeth Netrapalli, and Aaron Sidford. Accelerating stochastic gradient descent for least squares regression. In *Conference On Learning Theory*, pages 545–604. PMLR, 2018.

[33] 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.

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

[35] 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.

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

[37] 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*, pages 24174–24184, 2024.

[38] 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.

[39] Simon Lacoste-Julien, Mark Schmidt, and Francis Bach. A simpler approach to obtaining an o (1/t) convergence rate for the projected stochastic subgradient method. *arXiv preprint arXiv:1212.2002*, 2012.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] Depen Morwani, Nikhil Vyas, Hanlin Zhang, and Sham Kakade. Connections between schedule-free optimizers, ademamix, and accelerated sgd variants. *arXiv preprint arXiv:2502.02431*, 2025.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

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

[52] 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.

[53] Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. *arXiv preprint arXiv:1109.5647*, 2011.

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

[55] 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.

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

[57] 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.

[58] 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.

[59] 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.

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

[61] 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.

[62] 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.

[63] 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*, pages 23965–23998. PMLR, 2022.

[64] 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*, pages 24280–24314. PMLR, 2022.

[65] 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, 2022.

[66] 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.

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

[68] 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*, 2024.

[69] 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.

Appendix A. Large Batch Ablations

A.1. Large Batch Setting

We repeat the experiments from Figure 2 in a very large-batch regime (batch size 4096), well beyond the critical batch size (CBS). This regime is not our main focus since operating past the critical batch size (CBS) is generally inefficient (increasing the batch size does not reduce the serial runtime further) [68]. We therefore treat the large-batch setting as an ablation, in order to understand the limits of the quadratic model view.

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. Figure 3 confirms this behavior. With $B = 4096$, a constant learning rate with averaging substantially outperforms cosine for all horizons beyond $1\times$ Chinchilla, and the learning rate that is near-optimal for long runs remains near-optimal throughout training (unlike the CBS setting in Figure 2, which requires trading off short and long run performance). Note that the difference between constant with averaging and WSD decreases with N. We think this is due to a higher order effect caused by the deterministic edge of stability as studied in Damian et al. [16], where gradient descent is unable to learn features in the top subspace. However, we leave the precise study of this phenomenon to future work. The $1/\sqrt{t}$ schedule remains competitive and improves over cosine, but underperforms constant, which is again consistent with unnecessary decay in this regime.

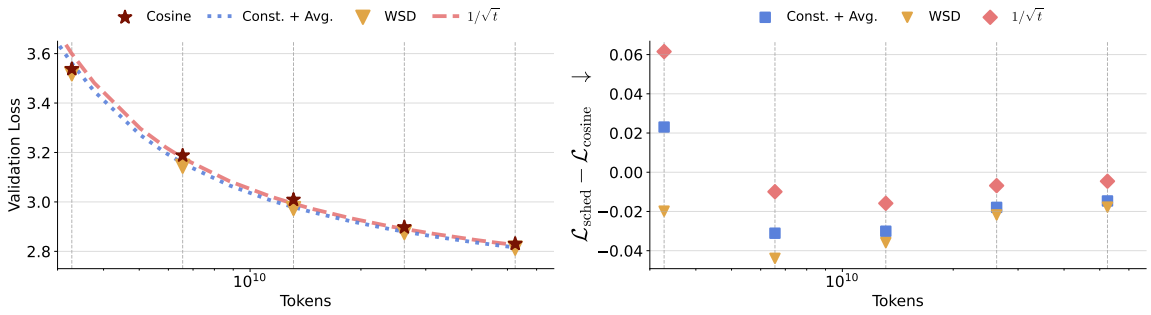


Figure 3: For a 150M-parameter model trained with batch size 4096, we compare cosine decay to constant learning rate with averaging, a $\sqrt{\alpha/(t+\alpha)}$ schedule with averaging, and WSD across end times ranging from $1\times$ to $16\times$ Chinchilla compute. **Left:** Validation loss versus 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 5 (Appendix D).

Appendix B. 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 - data covariance has finite rank, and in the infinite dimensional/nonparametric setup. Bach and Moulines [6], Polyak and Juditsky [51] introduced averaged SGD as an algorithm to achieve improved SGD convergence rates. Défossez and Bach [19] analyzed constant learning rate with averaged iterates in the strongly convex case, providing rates for the bias and variance terms, with similar proofs being shown in Dieuleveut and Bach [20], Jain et al. [31]. This analysis has been extended to minibatch gradient descent [33] and streaming algorithms [21]. When the horizon is known in advance, Jain et al. [34] have shown that a carefully designed step size sequence can be minimax optimal for last iterate SGD, building up on previous work by Harvey et al. [26], Shamir and Zhang [56]. Ge et al. [22] have shown that geometrically decaying step sizes are only $\log$ condition number suboptimal. Pan et al. [49] have shown that a more nuanced step size design can remove this suboptimality.

Nonparameteric Least Squares. Recent work by Zhang et al. [67] have shown that for power law spectra and under certain conditions on source and capacity exponents, averaging can be minimax optimal. Other schedules have been analyzed in a similar way [43, 64, 65, 69]. From a statistical physics point of view, Bordelon and Pehlevan [10] have established precise asymptotics for SGD in the overparameterized regime, with a similar analysis being done by Atanasov et al. [4, 5] using tools from random matrix theory. More recently, [68] have proposed a critical batch size scaling in pretraining, by choosing the batch size that balances the bias and variance rates in the quadratic analysis.

Learning rate scheduling and averaging in practice. These analyses have given rise to several new algorithms with practical impact in neural network training. Defazio et al. [17] have analyzed the linear decay schedule and have proposed further refinements for this schedule using the gradient norms. Linearly decaying to zero has been studied empirically by Bergsma et al. [8], achieving competitive performance to cosine annealing. Defazio et al. [18] have proposed a schedule-free optimizer, which takes advantage of tail averaging to remove the need for learning rate scheduling and an improved variant of momentum [32], as shown by Morwani et al. [46]. Hägele et al. [25] have shown that using stochastic weight averaging removes the need for learning rate scheduling, being comparable empirically to cosine annealing. It has been empirically observed that averaging can also lead to flatter minima and improved generalization performance on multiple image tasks [30]. More recently, stochastic weight averaging has obtained impressive results in the AlgoPerf competition [15, 38], across multiple downstream tasks [1]. Weight averaging has also been commonly applied in image generation and diffusion [37, 58, 66]. Other variants of weight EMA have been used in practice [3, 36, 40, 45, 55, 63].

Continual learning and anytime training. There is a growing body of literature studying continual learning and pretraining. 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 [2, 14, 35, 42, 54, 60, 61]. Anytime training [12, 24, 29] 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 [29], which has been explored by Belloni et al. [7], Wen

et al. [62] in the context of Warmup-Stable-Decay (WSD) [28] schedules. WSD has been introduced by Hu et al. [28] and has been used since in training recent frontier models [59].

Appendix C. Proofs of Section 3

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 4 *Let* $0 < \alpha \leq 1$ *and* $0 \leq j \leq i$. *Then*

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

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

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

Since $f'(x) = (1-\alpha)x^{-\alpha}$ and $x^{-\alpha}$ is decreasing for $\alpha > 0$, we have

$$(1-\alpha)i^{-\alpha} \;\leq\; f'(c) \;\leq\; (1-\alpha)j^{-\alpha}.$$

Multiplying by $(i-j)$ and absorbing the constant $(1-\alpha)$ into the $\lesssim$ notation yields the claim. ■

Setup. Before proceeding to the proofs, we briefly restate the notation and setup from Section 3. Let the independent and identically distributed covariates $\{(\mathbf{x}_i, y_i)\}_{i=1}^{s+N-1}$ where each $\mathbf{x}_i$ and y are $\mathbf{x} \sim \mathcal{N}(0, \mathbf{H})$ and $y = \mathbf{x}^\top w^\star + \epsilon$ for independent $\epsilon \sim \mathcal{N}(0, \sigma^2)$, where s is the step from which we start the 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 $\eta_t = \eta/t^\gamma$ for $\gamma < 1$ for some constant $1/\mathrm{Tr}(\mathbf{H}) \geq \eta > 0$. 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 covariance of the iterates. We also introduce $\mathbf{M}_t = \mathbf{Q}\mathbf{\Sigma}_t\mathbf{Q}^\top$ 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 $0 \leq i \leq j \leq \infty$ as $\mathbf{\Lambda}_{i:j} = \mathrm{diag}(\lambda_i, \ldots, \lambda_j)$.

C.1. Proof of Theorem 1

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

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

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

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

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

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

To make progress, we invoke the tower rule:

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

For SGD with stepsizes $\eta_t = \eta t^{-\gamma}$, $0<\gamma<1$,

$$\begin{aligned}\mathbb{E}[\mathbf{w}_j-\mathbf{w}^\star\mid\mathbf{w}_i] &= \prod_{t=i+1}^{j}(\mathbf{I}-\eta_t\mathbf{H})(\mathbf{w}_i-\mathbf{w}^\star)\\ &\preceq \exp\Big(-\mathbf{H}\sum_{t=i+1}^{j}\eta_t\Big)(\mathbf{w}_i-\mathbf{w}^\star)\\ &\lesssim \exp\Big(-\eta(j^{1-\gamma}-i^{1-\gamma})\mathbf{H}\Big)(\mathbf{w}_i-\mathbf{w}^\star) \qquad \text{Lemma 4}\end{aligned}$$

Diagonalizing inside the inner product we finally obtain the expression for the excess risk:

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

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. [43]. At batch size 1, the SGD update is

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

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

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

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

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

Taking expectation yields:

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

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

$$\begin{aligned}\mathbf{m}_t &= (\mathbf{I} - 2\eta_t\mathbf{\Lambda} + \mathbf{\Lambda}^2 + \lambda\lambda^\top)\mathbf{m}_{t-1} + \sigma^2\eta_t^2\lambda \\ &\leq (\mathbf{I} - \eta_t\mathbf{\Lambda})^2 + c\sigma^2\eta_t^2\lambda \qquad \text{Assumption 1}\end{aligned}$$

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

$$\mathbf{m}_{t+1} = \left[\prod_{i=1}^{t}(I - \eta_i\mathbf{\Lambda})^2\right]\mathbf{m}_0 + \sum_{p=0}^{t}\eta_p^2\sigma^2\left[\prod_{s=p+1}^{t}(I - \eta_s\mathbf{\Lambda})^2\right]\lambda \tag{3}$$

$$\leq \exp\Big[-2\mathbf{\Lambda}\sum_{i=1}^{t}\eta_i\Big]\mathbf{m}_0 + \sigma^2\sum_{p=0}^{t}\eta_p^2\exp\Big[-2\mathbf{\Lambda}\sum_{s=p+1}^{t}\eta_s\Big]\lambda \tag{4}$$

For $\eta_t = \eta t^{-\gamma}$ with $0 < \gamma < 1$,

$$\mathbf{m}_{t+1} \lesssim \exp\big[-2\eta\mathbf{\Lambda}t^{1-\gamma}\big]\mathbf{m}_0 + \eta\sigma^2\sum_{p=0}^{t}\frac{1}{p^{2\gamma}}\exp\Big[-2\eta\mathbf{\Lambda}(t^{1-\gamma} - p^{1-\gamma})\Big]\lambda.$$

We therefore define the bias and variance iterates as

$$\widetilde{\mathbf{m}}_{t+1} \coloneqq \exp\big[-2\eta\mathbf{\Lambda}t^{1-\gamma}\big]\mathbf{m}_0, \tag{5}$$

$$\overline{\mathbf{m}}_{t+1} \coloneqq \eta\sigma^2\sum_{p=0}^{t}\frac{1}{p^{2\gamma}}\exp\Big[-2\eta\mathbf{\Lambda}(t^{1-\gamma} - p^{1-\gamma})\Big]\lambda. \tag{6}$$

To obtain the rates, we independently bound each of these quantities after plugging them back in Equation (2).

Bias bound. We begin with the bias:

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

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.

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

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

$$\texttt{bias}_{t+1}^{k^\star:\infty} \lesssim \sum_{k>k^\star} \mathbf{m}_{0,k}\lambda_k.$$

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

$$\texttt{bias}_{t+1} \lesssim \frac{1}{N}\|\mathbf{w}^\star\|_{\mathbf{\Lambda}_{1:k^\star}} + \|\mathbf{w}^\star\|_{\mathbf{\Lambda}_{k^\star:\infty}} \tag{7}$$

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

$$\begin{aligned}
\texttt{variance}_{t+1} &\lesssim \sum_k \frac{\eta\sigma^2\lambda_k^2}{N^2} \sum_{i=s}^{s+N-1}\sum_{j=i}^{s+N-1}\sum_{p=0}^{i} \frac{1}{p^{2\gamma}} \exp\big[-2\eta\lambda_k(i^{1-\gamma}-p^{1-\gamma}) - \eta\lambda_k(j^{1-\gamma}-i^{1-\gamma})\big] \\
&\le \sum_k \frac{\eta\sigma^2\lambda_k^2}{N^2} \sum_{i=s}^{s+N-1}\sum_{j=i}^{s+N-1}\sum_{p=0}^{i} \frac{1}{p^{2\gamma}} \exp\big[-\eta\lambda_k(j^{1-\gamma}-p^{1-\gamma})\big] \\
&= \sum_k \frac{\eta\sigma^2\lambda_k^2}{N^2} \sum_{i=s}^{s+N-1}\sum_{j=i}^{s+N-1}\sum_{p=0}^{i} \frac{1}{p^{2\gamma}} \exp\big[-\eta\lambda_k(j^{1-\gamma}-i^{1-\gamma}+i^{1-\gamma}-p^{1-\gamma})\big] \\
&= \sum_k \frac{\eta\sigma^2\lambda_k^2}{N^2} \sum_{i=s}^{s+N-1}\sum_{j=i}^{s+N-1} \exp\big[-\eta\lambda_k(j^{1-\gamma}-i^{1-\gamma})\big] \sum_{p=0}^{i} \frac{1}{p^{2\gamma}} \exp\big[-\eta\lambda_k(i^{1-\gamma}-p^{1-\gamma})\big]
\end{aligned}$$

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

$$\sum_{p=0}^{i/2} \frac{1}{p^{2\gamma}} \exp\left[-\eta\lambda_k(i^{1-\gamma} - p^{1-\gamma})\right] \lesssim \sum_{p=0}^{i/2} \frac{1}{p^{2\gamma}} \exp\left[-\eta\lambda_k i^{-\gamma}(i-p)\right] \qquad \text{Lemma 4}$$

$$\lesssim \exp\left(-\frac{1}{2}\eta\lambda_k i^{1-\gamma}\right) \sum_{p=0}^{i/2} \frac{1}{p^{2\gamma}}$$

$$\lesssim \exp\left(-\frac{1}{2}\eta\lambda_k i^{1-\gamma}\right) \cdot \begin{cases} i^{1-2\gamma} & \gamma < 1/2 \\ \log i & \gamma = 1/2 \\ 1 & \gamma > 1/2 \end{cases}$$

Now we look at the second half. We have:

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

$$\lesssim \frac{i^{-\gamma}}{\eta\lambda_k}\left[1 - \exp\left(-\frac{1}{2}\eta\lambda_k i^{1-\gamma}\right)\right]$$

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

$$\sum_{j=i}^{s+N-1} \exp\left[-\eta\lambda_k(j^{1-\gamma} - i^{1-\gamma})\right] \lesssim \sum_{j=i}^{s+N-1} \exp\left[-\eta\lambda_k j^{-\gamma}(j-i)\right] \qquad \text{Lemma 4}$$

$$\lesssim \sum_{j=i}^{s+N-1} \exp\left[-\eta\lambda_k (s+N)^{-\gamma}(j-i)\right]$$

$$= \sum_{q=0}^{N-1} \exp\left[-\eta\lambda_k (s+N)^{-\gamma} q\right]$$

$$\lesssim \frac{(s+N)^{\gamma}}{\eta\lambda_k}\left(1 - \exp\left[-\eta\lambda_k \frac{N}{(s+N)^{\gamma}}\right]\right)$$

Assembling everything together we end up with:

$$\texttt{variance}_{t+1} \lesssim \sum_k \frac{\eta\sigma^2\lambda_k^2}{N^2} \sum_{i=s}^{s+N-1} \sum_{j=i}^{s+N-1} \exp\left[-\eta\lambda_k\left(j^{1-\gamma} - i^{1-\gamma}\right)\right] \sum_{p=0}^{i} \frac{1}{p^{2\gamma}} \exp\left[-\eta\lambda_k\left(i^{1-\gamma} - p^{1-\gamma}\right)\right]$$

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

$$\left. + \frac{i^{-\gamma}}{\eta\lambda_k}\left(1 - \exp\left(-\frac{1}{2}\eta\lambda_k i^{1-\gamma}\right)\right)\right].$$

where we have defined:

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

Recall the cutoff

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

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

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

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

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

$$\sum_{p=0}^{i} \frac{1}{p^{2\gamma}} \exp\Big[-\eta\lambda_k(i^{1-\gamma} - p^{1-\gamma})\Big] \ \lesssim\ \frac{i^{-\gamma}}{\eta\lambda_k}.$$

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

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

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

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

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,

$$\begin{aligned}
\texttt{variance}_{t+1} \lesssim \sum_k \frac{\eta\sigma^2\lambda_k^2}{N^2} \sum_{i=s}^{s+N-1} \frac{(s+N)^\gamma}{\eta\lambda_k} \Big(1 - \exp\Big[-\eta\lambda_k \tfrac{N}{(s+N)^\gamma}\Big]\Big) \\
\Big[\exp\big(-\tfrac{1}{2}\eta\lambda_k i^{1-\gamma}\big) s_i(\gamma) + \frac{i^{-\gamma}}{\eta\lambda_k}\big[1 - \exp\big(-\tfrac{1}{2}\eta\lambda_k i^{1-\gamma}\big)\big]\Big],
\end{aligned}$$

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

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

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 $\log s$ factor), we conclude

$$\texttt{variance}_{t+1}^{k^\star:\infty} \lesssim \sigma^2 \sum_{k>k^\star} \left(\eta\lambda_k^2\, s^{1-2\gamma} + \lambda_k\, s^{-\gamma}\right) \tag{8}$$

Combining the 2 bounds we get the final variance bound:

$$\texttt{variance}_{t+1} \lesssim \frac{\sigma^2}{\eta}\frac{(s+N)^\gamma}{Ns^\gamma}\, k^\star + \sigma^2 \sum_{k>k^\star} \left(\eta\lambda_k^2\, s^{1-2\gamma} + \lambda_k\, s^{-\gamma}\right) \tag{9}$$

■

C.2. Proof of Corollary 1

Proof [Proof of Corollary 2] Recall that we call $a, b > 1$ capacity and source exponents such that:

$$\lambda_i \eqsim i^{-a} \qquad\qquad \mathbb{E}\lambda_i(\mathbf{w}_i^\star)^2 \eqsim i^{-b}$$

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

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

In order to compute the optimal $\gamma^\star$, we have to balance out the rates. First note that in the bias, the bias tail dominates:

$$\frac{1}{N}\|\mathbf{w}^\star\|_{\mathbf{\Lambda}_{1:k^\star}} = \frac{1}{N}\sum_{k\leq k^\star} k^{-b} \lesssim 1 \lesssim \|\mathbf{w}^\star\|_{\mathbf{\Lambda}_{k^\star:\infty}}$$

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

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

Thus, we can now balance the 2 rates:

$$\frac{\sigma^2 k^\star}{\eta N} \eqsim \|\mathbf{w}^\star\|_{\mathbf{\Lambda}_{k^\star:\infty}}$$

We now compute each side as a function of $k^\star$ and then solve for $\gamma^\star$. By the source and capacity assumption,

$$\|\mathbf{w}^\star\|_{\mathbf{\Lambda}_{k^\star:\infty}} \;=\; \sum_{k>k^\star} \lambda_k (w_k^\star)^2 \;\eqsim\; \sum_{k>k^\star} k^{-b} \;\eqsim\; (k^\star)^{1-b}.$$

Thus balancing bias and variance gives

$$\frac{\sigma^2 k^\star}{\eta N} \;\eqsim\; (k^\star)^{1-b} \qquad \Longrightarrow \qquad (k^\star)^b \;\eqsim\; \frac{\eta N}{\sigma^2} \qquad \Longrightarrow \qquad k^\star \;\eqsim\; N^{1/b},$$

where $\eqsim$ absorbs constants (including η, σ^2) and logarithmic factors.

On the other hand, by definition of $k^\star$ and $\lambda_k \eqsim k^{-a}$,

$$\lambda_{k^\star} \;\eqsim\; (k^\star)^{-a} \;\eqsim\; \frac{1}{\eta N^{1-\gamma}} \qquad \Longrightarrow \qquad k^\star \;\eqsim\; N^{(1-\gamma)/a}.$$

Equating the two expressions for $k^\star$ yields

$$N^{(1-\gamma)/a} \;\eqsim\; N^{1/b} \qquad \Longrightarrow \qquad \gamma^\star \;=\; 1 - \frac{a}{b}.$$

This choice is feasible iff $b > a$ (so that $\gamma^\star \in (0,1)$). In this case,

$$\texttt{bias}_{t+1} \;\eqsim\; \texttt{variance}_{t+1} \;\eqsim\; (k^\star)^{1-b} \;\eqsim\; N^{-(b-1)/b}$$

■

C.3. Proof of Theorem 3

To prove Theorem 3, we follow a similar approach as before, beginning with deriving the expression for the risk for the last iterate case. Recall the setup from Appendix C.1. Following the derivation of [43, 64, 64, 69], the last iterate risk is:

$$\begin{aligned}\mathcal{R}(\mathbf{w}_t) &= \frac{1}{2}\mathbb{E}[((\mathbf{w}_t - \mathbf{w}^\star)^\top \mathbf{x} + \epsilon)^2] \\ &= \frac{1}{2}\mathbb{E}[(\mathbf{w}_t - \mathbf{w}^\star)\mathbf{x}_t\mathbf{x}_t^\top(\mathbf{w}_t - \mathbf{w}^\star) + \epsilon^2] \\ &= \frac{1}{2}\operatorname{Tr}(\mathbf{\Lambda}\mathbf{M}_t) + \frac{\sigma^2}{2}\end{aligned}$$

Using the same notation of $\lambda = \operatorname{diag}(\mathbf{\Lambda})$ and $\mathbf{m}_t = \operatorname{diag}(\mathbf{M}_t)$, we end up with the formula for the last iterate excess risk at time N:

$$\mathcal{R}(\mathbf{w}_N) - \sigma^2 \eqsim \langle \lambda, \mathbf{m}_N \rangle$$

As in Appendix C.1, unrolling the recursion on $\mathbf{m}_N$ we get:

$$\mathbf{m}_N \lesssim \exp\Big[-2\mathbf{\Lambda}\sum_{i=1}^{N}\eta_i\Big]\mathbf{m}_0 + \sigma^2\sum_{p=0}^{t}\eta_p^2\exp\Big[-2\mathbf{\Lambda}\sum_{s=p+1}^{N}\eta_s\Big]\lambda$$

with the corresponding bias and variance iterates:

$$\begin{aligned}\widetilde{\mathbf{m}}_N &= \exp\Big[-2\mathbf{\Lambda}\sum_{i=1}^{N}\eta_i\Big]\mathbf{m}_0 \\ \overline{\mathbf{m}}_N &= \sigma^2\sum_{p=0}^{t}\eta_p^2\exp\Big[-2\mathbf{\Lambda}\sum_{s=p+1}^{N}\eta_s\Big]\lambda\end{aligned}$$

Similarly, if we dot these quantities into λ we obtain the bias and variance contributions to the excess risk:

$$\mathcal{R}(\mathbf{w}_N) - \sigma^2 \eqsim \langle \widetilde{\mathbf{m}}_N, \lambda \rangle + \langle \overline{\mathbf{m}}_N, \lambda \rangle$$

and we will refer to these quantities as:

$$\begin{aligned}\texttt{bias}_N &\coloneqq \langle \widetilde{\mathbf{m}}_N, \lambda \rangle \\ \texttt{variance}_N &\coloneqq \langle \overline{\mathbf{m}}_N, \lambda \rangle\end{aligned}$$

Recall the schedule from the statement of Theorem 3. The schedule we are analyzing:

$$\eta_t = \begin{cases}\eta & 1 \le t \le t_0, \\ \eta\Big(1 - \frac{t - t_0}{N - t_0}\Big) & t_0 < t \le N\end{cases}$$

for $\eta \lesssim 1/\operatorname{Tr}(\mathbf{H})$, and $t_0 = \rho N$ for some constant $\rho \in (0, 1)$. Assume a power law spectrum on $\mathbf{H}$, with source capacity $a \in (1, 2)$ and source exponent $b > 1$ (defined as in Corollary 2). We first analyze the bias component.

Bias bound. By definition,

$$\texttt{bias}_N = \frac{1}{2}\sum_k \lambda_k m_{0,k} \exp\Big(-2\lambda_k \sum_{i=1}^{N-1} \eta_i\Big).$$

Since $\sum_{i=1}^{N-1} \eta_i \geq \sum_{i=1}^{t_0} \eta = \eta t_0 \gtrsim \eta N$, we obtain

$$\texttt{bias}_N \lesssim \sum_k \lambda_k \mathbf{m}_{0,k} \exp(-c\,\eta\lambda_k N).$$

where $c > 0$ is absorbing all the constants inside the exponential. Define the cut-off:

$$k^\star := \max\left\{k : \lambda_k \geq \frac{\log N}{\eta N}\right\}$$

Then in the bias head we have:

$$\texttt{bias}_N^{1:k^\star} \lesssim \frac{1}{N^c}\sum_{k\leq k^\star} \lambda_k \mathbf{m}_{0,k}$$

For the bias tail, we can upper bound the exponential by 1. Thus, the total bias bound is:

$$\texttt{bias}_N \lesssim \frac{1}{N^c}\sum_{k\leq k^\star} \lambda_k \mathbf{m}_{0,k} + \sum_{k>k^\star} \lambda_k \mathbf{m}_{0,k}$$

Now that we have finished bounding the bias we can proceed to bounding the variance.

Variance bound. From the unrolled variance iterate we have that,

$$\texttt{variance}_N = \frac{\sigma^2}{2}\sum_k \lambda_k^2 \sum_{p=1}^{N-1} \eta_p^2 \exp\Big(-2\lambda_k \sum_{s=p+1}^{N-1} \eta_s\Big).$$

We split the inner sum indexed by p into the constant phase ($p \leq t_0$) and the decay phase ($p > t_0$):

$$\texttt{variance}_N = \texttt{variance}_{N,\text{const}} + \texttt{variance}_{N,\text{dec}}$$

Constant learning rate phase. For $p \leq t_0$ we have $\eta_p = \eta$ and

$$\begin{aligned}
\sum_{s=p+1}^{N-1} \eta_s &= \eta(t_0 - p) + \sum_{s=t_0+1}^{N-1} \eta_s \\
&= \eta(t_0 - p) + \eta \sum_{s=t_0+1}^{N-1} \left(1 - \frac{s - t_0}{N - t_0}\right) \\
&\gtrsim \eta(t_0 - p) + \eta N
\end{aligned}$$

Plugging back in the variance iterate we have:

$$\begin{aligned}
\texttt{variance}_{N,\text{const}} &= \frac{\sigma^2}{2}\sum_k \lambda_k^2 \sum_{p=1}^{t_0} \eta^2 \exp\Big(-2\lambda_k \sum_{s=p+1}^{N-1} \eta_s\Big) \\
&\lesssim \sigma^2 \sum_k \lambda_k^2 \eta^2 e^{-c\eta\lambda_k N} \sum_{p=1}^{t_0} \exp\big(-2\eta\lambda_k (t_0 - p)\big) \\
&= \sigma^2 \sum_k \lambda_k^2 \eta^2 e^{-c\eta\lambda_k N} \sum_{r=0}^{t_0-1} e^{-2\eta\lambda_k r} \qquad (r = t_0 - p).
\end{aligned}$$

We can bound the inner sum by the series and we have:

$$\sum_{r=0}^{t_0-1} e^{-2\eta\lambda_k r} \le \sum_{r\ge 0} e^{-2\eta\lambda_k r} = \frac{1}{1-e^{-2\eta\lambda_k}} \lesssim \frac{1}{\eta\lambda_k}$$

where in the last step we have used the fact that $\eta\lambda_k \le 1$ (under our condition that $\eta \lesssim 1/\text{Tr}(\mathbf{H})$) and that for $x \in [0,1]$ we have $1 - e^{-x} \ge (1 - e^{-1})x \gtrsim x$. Plugging back in and splitting the sum over k at the same cutoff $k^\star$ we have:

$$\texttt{variance}_{N,\text{const}} \;\lesssim\; \sigma^2\,\eta\,N^{-c'} \sum_{k\le k^\star} \lambda_k \;+\; \sigma^2\,\eta \sum_{k>k^\star} \lambda_k$$

Linear decay learning rate phase. Now we shift our focus to the decay phase of the variance bound. For $p > t_0$ we have $\eta_p = \eta\left(1 - \frac{p-t_0}{N-t_0}\right)$ which yields:

$$\sum_{s=p+1}^{N} \eta_s \gtrsim \eta \frac{(N-p)^2}{N-t_0}$$

Plugging back in we have:

$$\begin{aligned}
\texttt{variance}_{N,\text{dec}} &\coloneqq \frac{\sigma^2}{2}\sum_k \lambda_k^2 \sum_{p=t_0+1}^{N-1} \eta_p^2 \exp\Big(-2\lambda_k \sum_{s=p+1}^{N-1} \eta_s\Big) \\
&\lesssim \eta^2\sigma^2 \sum_k \lambda_k^2 \sum_{u=1}^{N-t_0} \frac{u^2}{(N-t_0)^2} \exp\Big(-c\,\eta\lambda_k \frac{u^2}{(N-t_0)}\Big) \qquad u = N-p \\
&= \eta^2\sigma^2 \sum_k \lambda_k^2 \sum_{u=1}^{M} \frac{u^2}{M^2} \exp\Big(-c\,\eta\lambda_k \frac{u^2}{M}\Big) \qquad M = N - t_0 \\
&= \eta^2\sigma^2 \sum_k \lambda_k^2 S_k
\end{aligned}$$

where $c > 0$ is a constant and we have defined:

$$S_k = \sum_{u=1}^{M} \frac{u^2}{M^2} \exp\Big(-c\,\eta\lambda_k \frac{u^2}{M}\Big)$$

For the tail eigenvalues $k > k^\star$ we upper bound the exponential by 1 and thus we have:

$$S_{k>k^\star} \lesssim \sum_{u=1}^{M} \frac{u^2}{M^2} \lesssim M$$

For the head eigenvalues $k \leq k^\star$, we upper bound the sum by the integral:

$$S_{k\leq k^\star} \lesssim \frac{1}{M^2}\int_0^\infty u^2 \exp\Big(-c\,\eta\lambda_k \frac{u^2}{M}\Big)\,du \;\lesssim\; \frac{1}{M^2}\Big(\frac{M}{\eta\lambda_k}\Big)^{3/2} = \eta^{-3/2}M^{-1/2}\lambda_k^{-3/2}$$

Assembling everything, we end up with the variance bound:

$$\texttt{variance}_N \;\lesssim\; \sigma^2\,\eta\,N^{-c'}\sum_{k\leq k^\star}\lambda_k \;+\; \sigma^2\,\eta\sum_{k>k^\star}\lambda_k \;+\; \sigma^2\left(\eta^{1/2}M^{-1/2}\sum_{k\leq k^\star}\lambda_k^{1/2} \;+\; \eta^2 M\sum_{k>k^\star}\lambda_k^2\right)$$

Power law spectrum. Finally, we specialize the Hessian spectrum to the power law setting with capacity exponent $a \in (1,2)$ and source exponent $b > 1$, as well as fixing $\mathbf{w}_0 = 0$. Note that in the $\lesssim$ notation we now absorb $\log$ factors and constant factors.

Under $\lambda_k \eqsim k^{-a}$, we have:

$$(k^\star)^{-a} \eqsim \frac{\log N}{\eta N} \qquad \Longrightarrow \qquad k^\star \eqsim (\eta N)^{1/a}$$

In the bias term, the tail dominates, and substituting the expression for $k^\star$ yields:

$$\texttt{bias}_N \;\lesssim\; N^{-\frac{b-1}{a}}.$$

For the variance, in the constant-phase tail term we have:

$$\sigma^2\,\eta\sum_{k>k^\star}\lambda_k \;\eqsim\; \sigma^2\,\eta\sum_{k>k^\star}k^{-a} \;\lesssim\; \sigma^2\,\eta\,(k^\star)^{1-a} \;\lesssim\; \sigma^2\,N^{-\frac{a-1}{a}}$$

For the other 2 terms we have the following bounds. Since $\lambda_k^{1/2} \eqsim k^{-a/2}$ and $a/2 \in (1/2, 1)$, we have

$$\sum_{k\leq k^\star}\lambda_k^{1/2} \;\eqsim\; \sum_{k\leq k^\star}k^{-a/2} \;\lesssim\; (k^\star)^{1-a/2}.$$

Therefore,

$$\begin{aligned}\eta^{1/2}M^{-1/2}\sum_{k\leq k^\star}\lambda_k^{1/2} \lesssim \eta^{1/2}N^{-1/2}(k^\star)^{1-a/2} &= \eta^{1/2}N^{-1/2}\Big(\frac{\eta N}{\log N}\Big)^{\frac{1-a/2}{a}} \\ &= \eta^{\frac{1}{2}+\frac{1}{a}-\frac{1}{2}}N^{-\frac{1}{2}+\frac{1}{a}-\frac{1}{2}}\,(\log N)^{-\frac{1-a/2}{a}} \\ &\lesssim\; N^{-(a-1)/a}\end{aligned}$$

Since $\lambda_k^2 \eqsim k^{-2a}$ and $2a > 1$, we have

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

Therefore,

$$\eta^2 M \sum_{k>k^\star} \lambda_k^2 \lesssim \eta^2 N \,(k^\star)^{1-2a} = \eta^2 N \Big(\frac{\eta N}{\log N}\Big)^{\frac{1-2a}{a}}$$
$$= \eta^{2+\frac{1}{a}-2}\, N^{1+\frac{1}{a}-2}\, (\log N)^{\frac{2a-1}{a}} \;\lesssim\; N^{-(a-1)/a}$$

Appendix D. Additional Figures

D.1. Figure 2 at optimal hyperparamters

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.

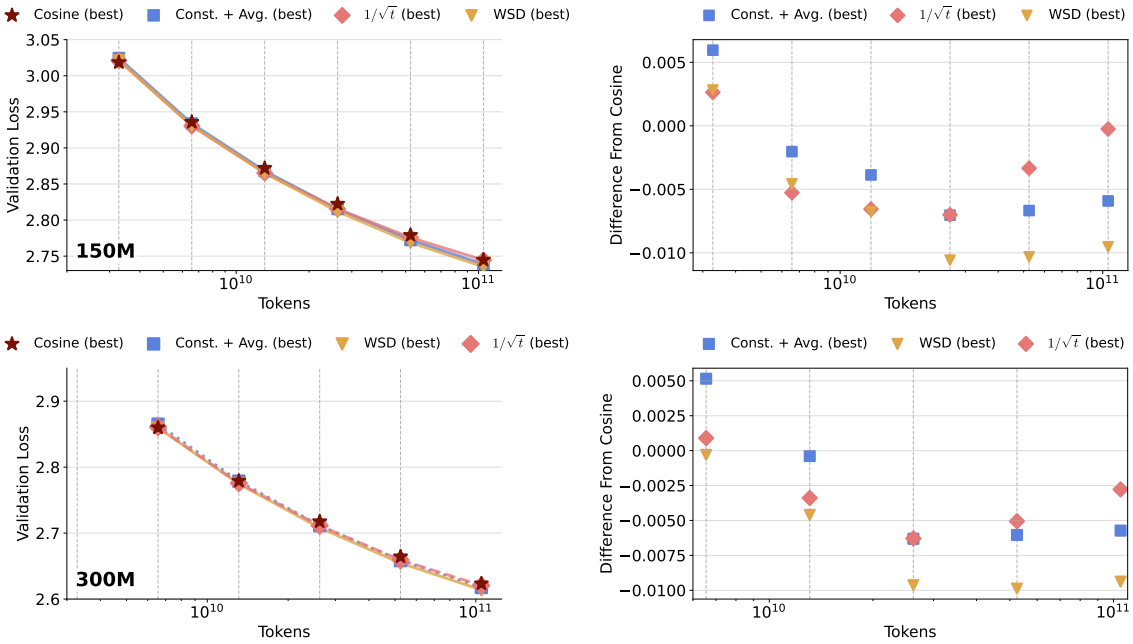


Figure 4: Top plots correspond to 150M parameter, and bottom plots correspond to 300M models. At each intermediate point, we plot the best loss out of the whole hyperparameter sweep, then we linearly interpolate between the points. Note that for long training durations, constant with averaging and $1/\sqrt{t}$ offers a substantial improvement over cosine decay.

D.2. Figure 3 at optimal hyperparameters

We similarly provide the losses from the experiment in Figure 3 for the optimal hyperparameters at each intermediate point.

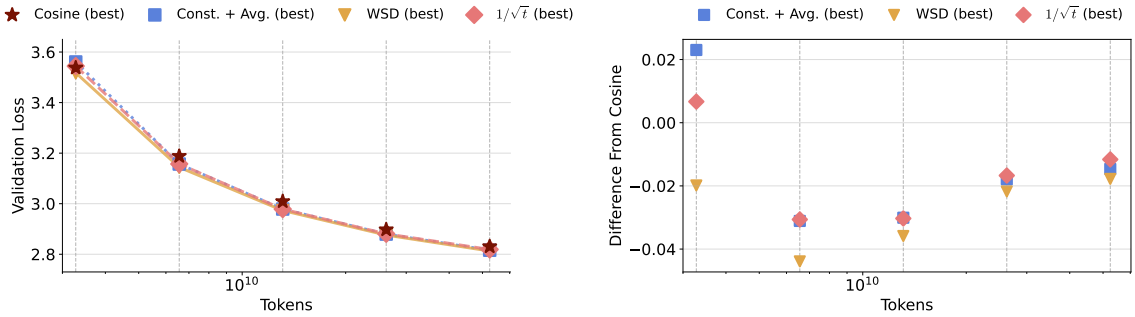


Figure 5: (Left) Validation loss comparison between cosine decay, constant learning rate with averaging, $1/\sqrt{t}$ with averaging and WSD on 150M models trained on $1\times$ to $16\times$ Chinchilla size datasets, at batch size 4096. (Right) The difference between the loss achieved by each schedule and cosine at each multiple of Chinchilla, with negative values meaning better than cosine. These 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. Note that the longer the training run, the more similar the schedules become indicating that at such large batch sizes learning rate decay is not needed.

D.3. Additional σ^2 values for synthetic experiments

We report additional synthetic linear-regression experiments under power-law spectra at different label-noise levels σ^2 using the parameterization used in the LLM experiments:

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

and sweep over α to assess how this horizon-free family interpolates between effectively constant and decaying step sizes. When α satisfies $\alpha = \Theta(N)$, the factor $\sqrt{\alpha/(t+\alpha)}$ remains roughly constant throughout the run, so the schedule behaves like a constant learning rate up to an $\mathcal{O}(1)$ rescaling. Thus, after tuning, the method can match both the qualitative rate and the final loss of constant learning rate with averaging in Figure 6 ($\sigma^2 = 0.001$), Figure 8 ($\sigma^2 = 0.0001$) and and Figure 7 ($\sigma^2 = 0.01$).

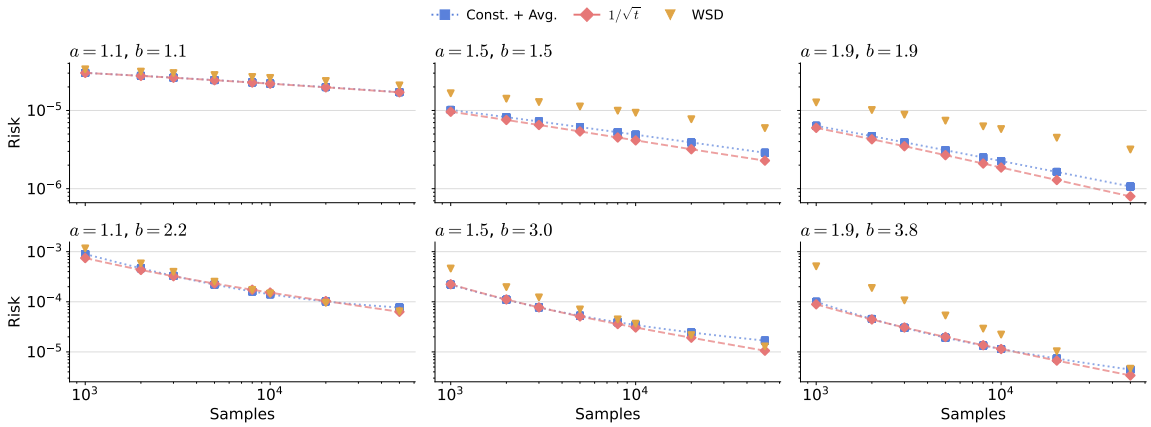


Figure 6: Risk comparison between schedulers in SGD on linear regression. We plot the exact risk recursion from Equation (4). The problem dimension is $d = 100000$ and we train for a maximum of $N = 50000$ samples at batch size 1, with label noise $\sigma^2 = 0.001$. We plot source exponents $a = 1.1$, 1.5 and 1.9 on the columns, and the top row corresponds to the capacity exponent $b = a$, and the bottom row corresponds to $b = 2a$. We sweep over learning rates $\eta/\mathrm{Tr}(\mathbf{H})$ for $\eta \in \{0.1, 0.25, 0.5, 1.0, 1.25, 1.5, 1.9\}$. For constant with averaging and $1/\sqrt{t}$ we average over the last fraction $f \in \{1.0, 0.5, 0.25, 0.125, 0.0625\}$ of iterates. For $1/\sqrt{t}$ we use the practical implementation of $\sqrt{\frac{\alpha}{t+\alpha}}$ and sweep over $\alpha \in \{400, 800, 1600, 3200, 6400, 12800, 25600\}$. For WSD, we fix intermediate points during the run at $1000, 2000, 3000, 5000, 8000, 10000, 20000, 50000$ samples and run until a fraction p of each with constant learning rate, followed by a linear decay, where $p \in \{0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. For each run, hyperparameters are chosen such that they are close to anytime optimal. We provide plots at other values of σ^2 in Figures 7 and 8 (Appendix D).

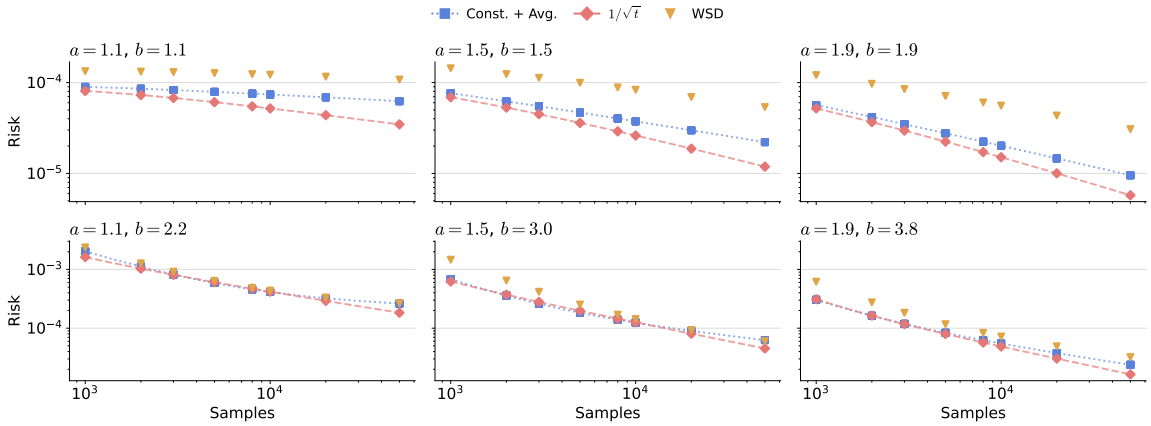


Figure 7: Same setup as Figure 6 using $\sigma^2 = 0.01$

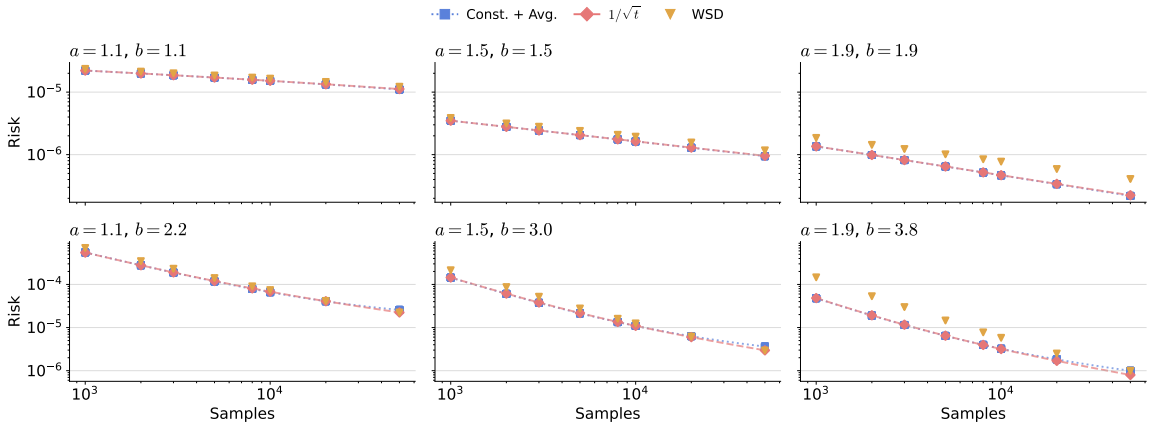
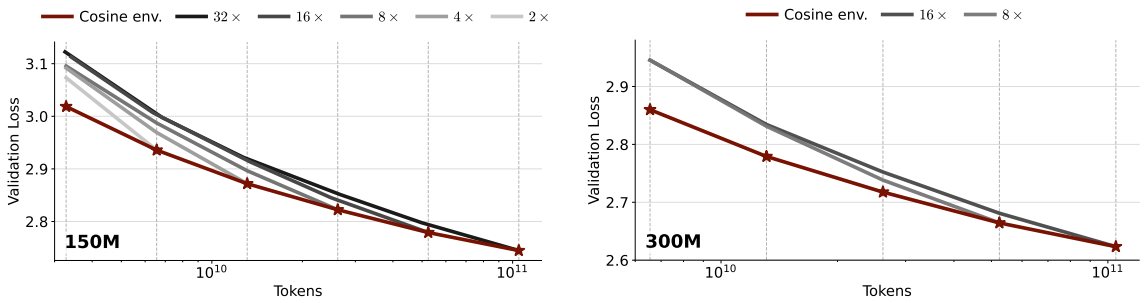


Figure 8: Same setup as Figure 6 using $\sigma^2 = 0.0001$

D.4. Optimal cosine envelope

Figure 9 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 below 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 9: 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 $1\times$–$32\times$; we plot the closest recorded validation point.

Appendix E. Synthetic Plots

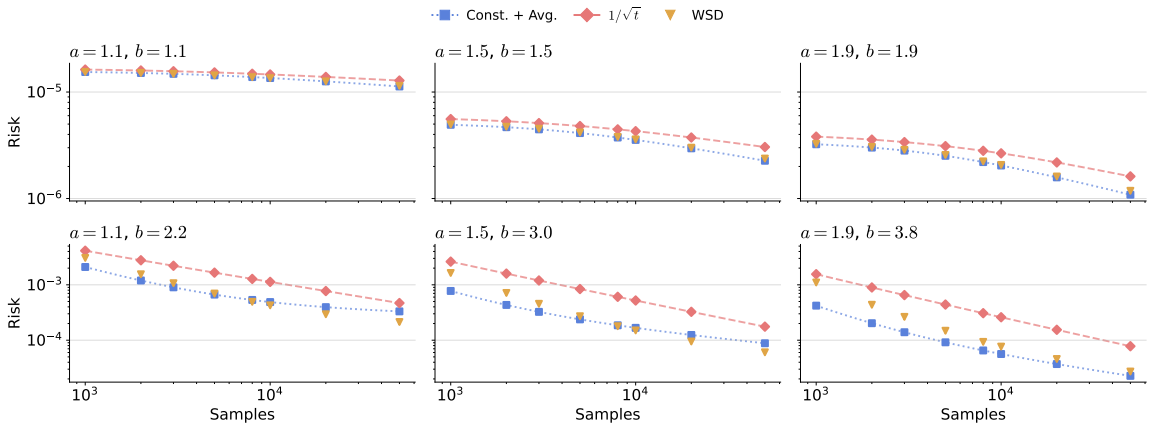


Figure 10: Risk comparison between schedulers in SGD on linear regression. We plot the exact risk recursion from (4). The problem dimension is $d = 500000$ and we train for a maximum of $N = 50000$ samples at batch size 1, with label noise $\sigma^2 = 0.01$. We plot source exponents $a = 1.1$, 1.5 and 1.9 on the columns, and the top row corresponds to the capacity exponent $b = a$, and the bottom row corresponds to $b = 2a$. We sweep over learning rates $\eta \in \{0.0001, 0.0002, 0.0005, 0.0007, 0.001, 0.002, 0.005, 0.01, 0.02, 0.03, 0.05, 0.075, 0.1, 0.2, 0.3, 0.5, 0.8, 1.0\}$. For constant with averaging and $1/\sqrt{t}$ we average over the whole duration of the run and we only use the last iterate for WSD. For WSD, we fix intermediate points during the run at $1000, 2000, 3000, 5000, 8000, 10000, 20000, 50000$ samples and run until a fraction p of each with constant learning rate, followed by a linear decay, where $p \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$. For each run, hyperparameters are chosen such that they are close to anytime optimal by minimizing the average loss over intermediate points.

Appendix F. Empirical Details

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. 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, Appendix D includes additional loss curves obtained by selecting the best hyperparameter setting separately for each training budget, spanning $1\times$ to $32\times$ Chinchilla tokens.

Architecture and Dataset. We pretrain 150M and 300M models based on the OLMo codebase [23, 47, 48]. 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 $\beta_1 = 0.9$, and we sweep over learning rate $\eta \in \{0.0001, 0.0003, 0.001, 0.003, 0.01\}$ for 150M (and $\eta \in \{0.0003, 0.001, 0.003\}$ for 300M), preconditioner $\beta_2 \in \{0.95, 0.98, 0.99\}$ for 150M (and $\beta_2 \in \{0.95, 0.99\}$ for 300M). We train without z-loss and with a sequence length $L = 1024$. We train our models on the C4 dataset [52], 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 [27] calculation of 20 tokens-per-parameter (TPP), we take as $1\times$ 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 at the critical batch size for $1\times$ Chinchilla, based on Zhang et al. [68], 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 40% of the respective $1\times$ Chinchilla number of tokens for each of the models, for all training runs. For the 150M model, we train a cosine baseline at each power-of-2 multiple of Chinchilla from $1\times - 32\times$. For the $1\sqrt{t}$ and constant with averaging runs, we train directly for the $32\times$ 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 due to computational constraints we stop at $16\times$.

Averaging. We evaluate using weight averaging based on EMA with parameter τ_t i.e. $\bar{\mathbf{w}}_{t+1} = (1-\tau_t)\bar{\mathbf{w}}_t + \tau_t\theta_t$ for parameters θ_t. We choose the schedule for $\tau_t = 1/2^{f/t}$ such that the half life of the EMA is equal to some fraction f of the current time t (i.e. for half life h meaning $\tau_t^h = 1/2$ we want $h = t/f$). This schedule ensures that at time t, the EMA is averaged over approximately the last $1/f$ iterates. For all runs, we maintain multiple EMAs for $f \in \{0.0, 6.25, 12.5, 25.0, 50.0, 100.0\}$ where for $f = 0.0$ we refer to only using the last iterate, without any averaging.

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. Bjorck et al. [9], Mlodozeniec et al. [44] 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

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

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 [9, 57], but for all our experiments, we chose to keep $1/2$. We formalize this bias–variance decomposition and the resulting recursion in Appendix C, equation (4). Note that while theoretically we establish guarantees for $1/\sqrt{t}$, these guarantees hold only up to constant factors. In order to account for these constants, in practice we parameterize our scheduler as $\eta\sqrt{\frac{\alpha}{t+\alpha}}$ for a tunable positive constant α, which we tune over $\alpha \in \{400, 800, 1600, 3200, 6400, 12800, 25600, 51200\}$. While α does introduce a dependence on the total number of steps, and thus implicitly on the time horizon, this dependency is weak, and we show in Figure 2 that we can find an α value that is close to optimal across the cosine envelope from $1\times$ to $32\times$, and $16\times$ for 150M and 300M models, respectively. While this parameterization is not predicted by the quadratic analysis, we will refer to this schedule as $1/\sqrt{t}$ for the remainder of this manuscript in line with our theoretical predictions and specify case by case which parameterization we are refering to.