Theory of Optimal Learning Rate Schedules and Scaling Laws for a Random Feature Model

Blake Bordelon[1,2*], **Francesco Mori**[1*]
[1]Center for Mathematical Sciences and Applications, Harvard University
[2]Oden Institute for Computational Engineering and Sciences & Dept. of Neuroscience, UT Austin
{ blake , francesco }@cmsa.fas.harvard.edu

Abstract

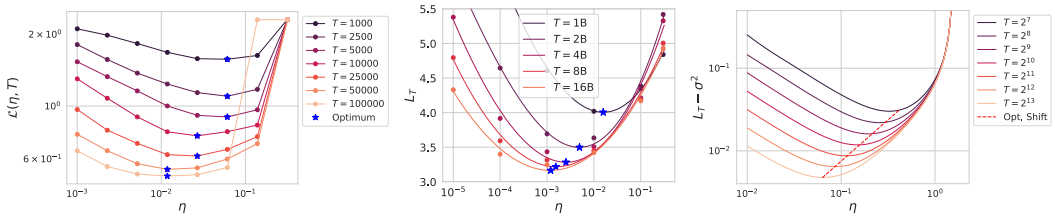
Setting the learning rate (LR) for a deep learning model is a critical part of successful training. Choosing LRs is often done empirically with trial and error. In this work, we explore a solvable model of optimal LR schedules for a powerlaw random feature model trained with stochastic gradient descent (SGD). We consider the optimal schedule $\eta_T^\star(t)$ where t is the current iterate and T is the training horizon. This schedule is computed both as a numerical optimization problem and also analytically using optimal control theory. Our analysis reveals two regimes which we term the easy phase and hard phase. In the easy phase the optimal schedule is a polynomial decay $\eta_T^\star(t) \simeq T^{-\xi}(1-t/T)^\delta$ where ξ and δ depend on the properties of the features and task. In the hard phase, the optimal schedule features an initial period of constant LR, followed by annealing performed over a vanishing fraction of training steps, similar to warmup-stable-decay. We investigate joint optimization of LR and batch size and find batch ramps can improve the wall-clock time in the easy phase. Beyond SGD, we derive optimal schedules for momentum parameter $\beta(t)$ and show that it improves the loss-scaling exponent in the hard phase. We compare our optimal schedule to various benchmarks including (1) optimal constant learning rates $\eta_T(t) \sim T^{-\xi}$ (2) optimal power laws $\eta_T(t) \sim T^{-\xi}t^{-\chi}$, finding that our schedule achieves better rates than either of these. Our theory suggests that LR transfer across training horizon depends on the structure of the model and task. For ResNet image classification on CIFAR-5M, the learning curves exhibit hard-phase behavior where optimal base LRs are constant under sufficient annealing. GPT-2 style transformers trained in language modeling exhibit easy-phase behavior where optimal LRs shift even under annealing.

1 Introduction

Training deep learning models requires choosing many hyperparameters such as the learning rate (LR), batch size, training horizon, total data, and architectural details, resulting in a complicated decision space. To make matters worse, it is challenging to characterize how these various hyperparameters jointly interact as one scales up the model size or training horizon. One strategy to reduce this complexity is to identify scaling protocols that allow for *transfer* of optimal hyperparameters, where small models can provide reliable proxies for tuning for large models. Such approximate hyperparameter transfer across model sizes can be achieved by principled parameterization and optimizer design Yang et al. [2022], Bordelon et al. [2023], Yang et al. [2023], Dey et al. [2025]. However, these schemes do not automatically lead to transfer over training horizons Bjorck et al. [2024], Everett et al. [2024]. In Figure 1 we illustrate the failure of transfer over training horizons T despite having successful transfer over model sizes (widths N). This failure motivates theory that

*Equal contribution

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) ResNet on CIFAR-5M (b) Transformer on C4 (c) Random Feature Model

Figure 1: Learning rates do not automatically transfer over training horizons T. This motivates theory that can identify not only how to scale η with T, but also how to set the entire learning rate schedule $\eta(t)$ with T. (a) The loss of a deep ResNet trained on CIFAR-5M. (b) GPT-style transformer trained on next word prediction on C4 dataset. (c) Test loss of a random feature model trained with SGD as a function of fixed learning rate η.

can account for the behavior of optimal learning rates of SGD as the training horizon varies. Further, beyond optimal base learning rates themselves, one can also adopt a **learning rate schedule**. While many schedules such as linear decay, warmup-stable-decay, and cosine annealing are commonly used in practice, it is unclear under which conditions one should favor one type of schedule over the other.

To address these questions in a theoretical framework, we make the following contributions:

- We study SGD dynamics in a powerlaw random feature model Bordelon and Pehlevan [2022], Bordelon et al. [2024], Paquette et al. [2024]. We show SGD fluctuations can cause shifts in optimal learning rates across training horizons T that are **task and model dependent**.
- We characterize both numerically and analytically the **optimal learning rate schedule** for SGD in our model using optimal control, minimizing the final test loss at step T. We prove that the exact loss is controlled, up to constant factors, by a continuous-time functional of the schedule. This analysis reveals two phases of tasks, easy and hard. For easy tasks, our schedule takes the form $\eta(t) = T^{-\xi} f(t/T)$, while for hard tasks our schedule features an initial period of constant learning rate, followed by annealing performed over a vanishing fraction of training steps, similar to warmup-stable-decay.
- We compare optimal schedules to benchmarks including optimized constant learning rate and optimal powerlaw schedules, finding that our schedules can have better scaling exponents.
- This analysis is generalized to optimal **batch size** schedules and optimal **momentum** schedules. Optimizing batch size and learning rate jointly enables a reduction in total wall-clock time, while optimizing the momentum parameter can provide an improvement in the scaling exponent for hard tasks beyond SGD.
- We empirically validate both phases in deep learning settings: ResNets on CIFAR-5M exhibit hard-phase behavior (constant optimal base learning rate under annealing), while GPT-style transformers pretrained on C4 Raffel et al. [2020] exhibit easy-phase behavior (optimal learning rate shifts with horizon T).

1.1 Related Works

Based on the correspondence between neural networks in the lazy training regime Chizat et al. [2019], many works have attempted to characterize neural scaling laws Kaplan et al. [2020], Hoffmann et al. [2022] by computing the generalization of random feature models with powerlaw features Varre et al. [2021], Bordelon and Pehlevan [2022], Bahri et al. [2024], Bordelon et al. [2024], Paquette et al. [2024], Lin et al. [2024], Ferbach et al. [2025]. Several works have examined easy and hard phases of this model in the noise dominated regime Dieuleveut and Bach [2015], Lin and Rosasco [2017]. Notably, Pillaud-Vivien et al. [2018] demonstrated that optimal rates for SGD can be achieved in the hard phase with data repetition and model averaging strategies. Meterez et al. [2025] recently analyzed this model to motivate invariances between learning rate and batch sizes when designing schedules. Qiu et al. [2025] identified a scaling collapse phenomenon where the compute optimal models exhibit universal loss dynamics across scales for a variety of learning rate schedules. Our

work similarly utilizes this powerlaw random feature model, however we study the *optimal learning rate schedule* for SGD using optimal control theory.

Optimal control theory provides a principled framework for deriving hyperparameter schedules in reduced theoretical models. Pioneering work in the 1990s characterized optimal learning rate schedules for two-layer neural networks within the teacher-student formalism Saad and Rattray [1997], Rattray and Saad [1998]. More recently, these techniques have been generalized to several hyperparameter protocols Mori et al. [2025], Mignacco and Mori [2025]. Analogous optimization procedures have also recently been studied in deep linear networks Carrasco-Davis et al. [2023] and cognitive science Njaradi et al. [2026].

Most closely related to our work is Li et al. [2025], which examined learning rate schedules within the context of powerlaw random feature models. However, their analysis was restricted to pre-defined, parametric schedules and did not address the theoretical derivation of the optimal control policy.

2 Solvable Model and Test Loss Dynamics

We first define the model and describe the evolution of the test error under SGD before formulating the optimal control problem that we aim to solve.

2.1 Power Law Linear Random Feature Model

We consider a random-feature model widely studied in the literature of scaling laws Bordelon and Pehlevan [2022], Bordelon et al. [2024], Lin et al. [2024], Paquette et al. [2024]. Let the input data $\boldsymbol{x} \in \mathbb{R}^D$ be drawn from a probability distribution $p(\boldsymbol{x})$. The target function generates noisy labels as

$$y(\boldsymbol{x}) = \boldsymbol{w}^* \cdot \boldsymbol{\psi}(\boldsymbol{x}) + \sigma_0 z\,, \tag{1}$$

where $\sigma_0 > 0$ is the amplitude of the label noise and $z \sim \mathcal{N}(0,1)$. The vector $\boldsymbol{\psi}(\boldsymbol{x}) \in \mathbb{R}^M$ (with M possibly infinite) is a feature map and $\boldsymbol{w}^* \in \mathbb{R}^M$ a vector of parameters. We work in the feature eigenbasis, so that the correlation matrix is diagonal

$$\langle \psi_i(\boldsymbol{x})\psi_j(\boldsymbol{x})\rangle_{\boldsymbol{x}\sim p(\boldsymbol{x})} = \delta_{ij}\lambda_i\,. \tag{2}$$

We consider a linear student model $\hat{y}(\boldsymbol{x}) = \boldsymbol{w}\cdot\tilde{\boldsymbol{\psi}}(\boldsymbol{x})$, with parameters $\boldsymbol{w} \in \mathbb{R}^N$ and projected features $\tilde{\boldsymbol{\psi}}(\boldsymbol{x}) = \boldsymbol{G}\boldsymbol{\psi}(\boldsymbol{x})$, where $\boldsymbol{G} \in \mathbb{R}^{N\times M}$ is a projection matrix. To describe model-size effects, we assume $N \le M$ and that the top-N features are selected, i.e., $\boldsymbol{G} = [\boldsymbol{I}_N, \boldsymbol{0}_{N\times(M-N)}]$(a Gaussian $\boldsymbol{G}$ yields approximately the same loss-scaling exponents in T, see Appendix H). We train via online stochastic gradient descent (SGD) with learning rates $\{\eta_t\}$ and batch sizes $\{m_t\}$ on the square loss

$$\text{SGD Dynamics:}\quad \boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \frac{\eta_t}{m_t}\sum_{\mu=1}^{m_t}\tilde{\boldsymbol{\psi}}(\boldsymbol{x}_{\mu,t})(\boldsymbol{w}_t\cdot\tilde{\boldsymbol{\psi}}(\boldsymbol{x}_{\mu,t}) - y(\boldsymbol{x}_{\mu,t}))\,. \tag{3}$$

The samples $\boldsymbol{x}_{\mu,t}$ and the label noise variables $z_{\mu,t}$ are sampled independently.

2.2 Test Loss Dynamics of the Model

Because this model generates linear dynamics for the parameters, the expected test loss (over random draws of SGD minibatches) can be computed from a simple recurrence.

Theorem 1 *Under Gaussian features ψ, the expected test loss L_t over the SGD datastream $\mathcal{D}_{t-1}$ up to time t obeys the following dynamics*

$$L_t = \left\langle (\hat{y}_t(\boldsymbol{x}) - y(\boldsymbol{x}))^2 \right\rangle_{\boldsymbol{x},z,\mathcal{D}_{t-1}} = \sum_{k=1}^{N}\lambda_k \underbrace{\left\langle (w_{t,k} - w_k^\star)^2\right\rangle_{\mathcal{D}_{t-1}}}_{:=\,c_{t,k}} + \underbrace{\sum_{k=N+1}^{M}\lambda_k (w_k^\star)^2 + \sigma_0^2}_{:=\,\sigma^2} \tag{4}$$

*where the mode errors $c_{t,k}$ obey the following **linear recurrence** that depends on effective noise σ^2*

$$c_{t+1,k} = \left(1 - 2\eta_t\lambda_k + \eta_t^2\,\frac{m_t+1}{m_t}\,\lambda_k^2\right)c_{t,k} + \frac{\eta_t^2}{m_t}\lambda_k\sum_{\ell=1}^{N}\lambda_\ell\, c_{t,\ell} + \frac{\eta_t^2}{m_t}\sigma^2\lambda_k. \tag{5}$$

This recurrence defines L_t as a function of learning rate and batch size schedules $\{(\eta_t, m_t)\}$.

This result merely extends the main theorem from Bordelon and Pehlevan [2022] to arbitrary learning rate and batch schedules and adds a projection cutoff that generates the effective noise σ^2. We provide a detailed proof and discussion in Appendix E.

Power-law spectra While the result above holds for arbitrary spectra, we will often specialize to a particular structural assumption on the features. Motivated by empirical observations that natural data often exhibit power-law spectral decay [Bordelon and Pehlevan, 2022, Bahri et al., 2024], we assume power-law scalings for the data covariance eigenvalues and the teacher weights:

$$\lambda_k \sim k^{-b} \ , \ (w_k^*)^2 \lambda_k \sim k^{-a} \tag{6}$$

with exponents $a, b > 1$. These exponents (a, b) characterize the effective dimension of the features and difficulty of the learning task. These power-law assumptions correspond to standard source-capacity conditions in the kernel-regression literature, under which the minimax-optimal scaling of the excess risk at fixed batch size is $L_T - \sigma^2 \sim T^{-(a-1)/a}$ Caponnetto and De Vito [2007].

3 Optimal Control for Optimal Hyperparameters

From the closed form loss dynamics, we aim to choose the optimal hyperparameter schedules that **minimize the final loss** L_T after T steps. We can formulate and compare many optimal control strategies with this method. Let $L_t(\boldsymbol{\eta}, \boldsymbol{m})$ represent the loss for learning rates $\boldsymbol{\eta} = \text{Vec}\{\eta_t\}_{t=1}^T$ and $\boldsymbol{m} = \text{Vec}\{m_t\}_{t=1}^T$. Let $\mathbf{1} \in \mathbb{R}^T$ be the vector of all ones so that $\eta \mathbf{1}$ represents a constant learning rate schedule. Throughout this work, we will define and compare three different optimization problems that correspond to optimizing over different subsets of variables

$$\begin{aligned}
&\text{Optimal Constant LR:} \quad \eta_T^\star = \text{argmin}_{\eta \in \mathbb{R}} \quad L_T(\eta \mathbf{1}, m\mathbf{1}) \\
&\text{Optimal LR Schedule:} \quad \boldsymbol{\eta}_T^\star = \text{argmin}_{\boldsymbol{\eta} \in \mathbb{R}^T} \ L_T(\boldsymbol{\eta}, m\mathbf{1}) \\
&\text{Optimal LR and Batch Schedule:} \quad \boldsymbol{\eta}_T^\star, \boldsymbol{m}_T^\star = \text{argmin}_{\boldsymbol{\eta} \in \mathbb{R}^T, \boldsymbol{m} \in \mathbb{R}^T} \ L_T(\boldsymbol{\eta}, \boldsymbol{m}).
\end{aligned} \tag{7}$$

In the joint LR+batch optimization, the total data budget $B_{\text{tot}} = \sum_{t=1}^T m_t$ is held fixed. The optimal hyperparameter schedules depend on the total training horizon T. We stress that each of these optimization problems is finite dimensional for finite training horizon T and can be computed efficiently due to the linear recurrence for the $c_{t,k}$ variables that define L_T using Theorem 1. We will take two approaches to these optimization problems. First, we will solve numerically for the optimal learning rate using the *exact recurrence for the loss* L_T. Second, we will provide an approximate (yet accurate) description of the optimal control scaling laws under the powerlaw feature assumption.

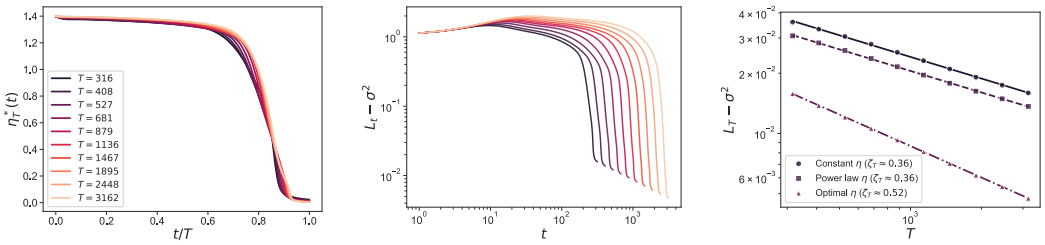
3.1 Numerical optimal control

We first approach these optimization problems numerically by optimizing the evolution equation in Theorem 1 using a single-shooting method implemented in CasADi Andersson et al. [2019], see Appendix G for details. This procedure allows us to identify the optimal learning rate schedules $\eta_T^*(t)$ for various training horizons T, as illustrated in Figures 2a and 2d. Crucially, we observe that across different regimes of data structure (parameterized by power-law exponents a and b), the optimal schedules achieve a higher scaling exponent compared to the optimal constant-η baseline.

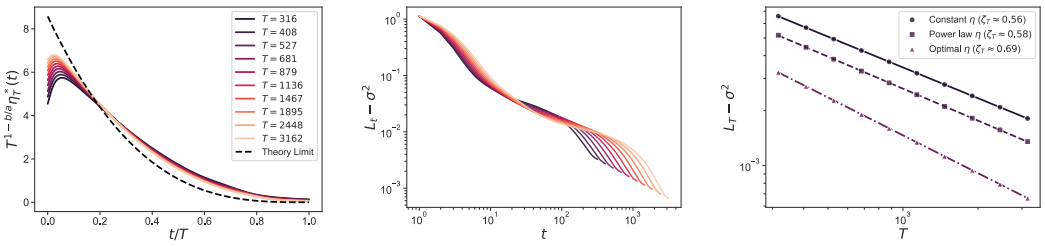
3.2 Analytical derivation of approximate optimal schedules and scaling laws

To determine the analytical form of the optimal schedule, we consider the continuous-time approximation for large T (when taking this limit, we will often use the notation $\eta(t)$ for a learning rate as a function of time $t \in \mathbb{R}$). We first show that this approximation is controlled. We embed the schedule in continuous time as $\eta(t) = \eta_j$ for $t \in [j, j+1)$ and define the integrated learning rate $\chi(t) \coloneqq \int_t^T ds\, \eta(s)$.

Theorem 2 (Controlled continuous-time loss) *Under Gaussian features ψ with power-law spectra $\lambda_k = k^{-b}$ and $\lambda_k (w_k^\star)^2 = k^{-a}$ $(a, b > 1)$, label noise $\sigma_0 > 0$, initialization $\boldsymbol{w}_0 = \mathbf{0}$, constant batch size m, and learning rates $0 \leq \eta_t \leq \eta_{\max} < 2m/(m + 1 + \zeta(b))$, where ζ is the Riemann zeta*



(a) Optimal Schedule ($b > a$) (b) Loss vs. Time ($b > a$) (c) Final Loss ($b > a$)



(d) Optimal Schedule ($b < a$) (e) Loss vs. Time ($b < a$) (f) Final Loss ($b < a$)

Figure 2: **Comparison of optimal learning rate schedules in the hard** ($b > a$) **and easy** ($b < a$) **phases. (a, d)**. Profile of the optimal learning rate $\eta_T^*(t)$, obtained by numerically optimizing the exact discrete dynamics of Theorem 1 with CasADi. In the hard phase **(a)**, the schedule maintains an approximately constant maximum value followed by a rapid annealing phase. In the easy phase **(d)**, the schedule collapses onto the scaling form $\eta_T^*(t) \approx T^{b/a-1} f(t/T)$ (dashed theoretical curve, see Result 2). **(b, e)** Evolution of the loss over training time t for the optimal schedule. **(c, f)** Scaling of the loss with T. The optimal schedule improves scaling exponent compared to the constant and power-law baselines. Exponents obtained from numerical fit match theoretical predictions: for the constant and power-law case, $\zeta_T = \frac{a-1}{a+b-1}$ ($\zeta_T = 1/3$ for panel (c) and $\zeta_T = 5/9 \approx 0.56$ for panel (f)), for the optimal schedule $\zeta_T = \min(\frac{a-1}{a}, \frac{a-1}{b})$ ($\zeta_T = 0.5$ for panel (c) and $\zeta_T = 5/7 \approx 0.71$ for panel (f)). **Parameters:** $N = 500$, $\sigma = 0.5$, $a = 3.5$, $m = 5$, with $b = 5$ (top) and $b = 2$ (bottom).

function, the expected test loss of Theorem 1 satisfies

$$L_T - \sigma^2 \asymp (1 + \chi(0))^{-\frac{a-1}{b}} + \frac{\sigma^2}{m} \int_0^T dt\, \eta(t)^2 \, (1 + \chi(t))^{-\frac{2b-1}{b}} \tag{8}$$

whenever $N \geq 4\,(1 + \chi(0))^{1/b}$. Here $\asymp$ denotes equality up to multiplicative constants independent of T, N and the schedule.

The proof is given in Appendix F. For the optimal schedules derived below, $\chi(0) \sim T^{b/a}$ in the easy phase and $\chi(0) \sim T$ in the hard phase, so the condition for the validity of Eq. (8) becomes $N \gg T^{1/\max(a,b)}$. Theorems 1 and 2 are rigorous, whereas Results 1–3 below are derived at leading order in large T. For constant batch size (Results 1 and 2), Theorem 2 implies that, for a given schedule, the scaling exponents of the loss are not affected by the approximations, while the prefactors and precise schedule shapes are leading-order predictions that we validate numerically. We will provide a comparison of each of the different optimal control problems, starting with optimal constant learning rate (see Appendix C for details on the derivation).

Result 1 (Optimal Constant LR) *Assuming powerlaw features with exponents (a, b) and nonzero effective noise $\sigma^2 > 0$, the late time $T \gg 1$ value of the loss under constant η, m is*

$$L_T - \sigma^2 \sim \underbrace{(\eta\, T)^{-\frac{a-1}{b}}}_{\textit{Bias}} + \underbrace{\frac{\sigma^2 \eta}{m}}_{\textit{Variance}} . \tag{9}$$

As a consequence the optimal learning rate and loss scale as $\eta \sim T^{-\frac{(a-1)}{a-1+b}}$ and $L_T - \sigma^2 \sim T^{-\frac{a-1}{a-1+b}}$.

This scaling law can be outperformed by optimizing the entire learning rate schedule.

Result 2 (Optimal LR, Constant Batch) *Under a continuous time approximation of the loss in Theorem 1 and assuming (1) power-law spectra with exponents (a, b) (2) non-zero effective noise $\sigma^2 > 0$ (3) the batchsize $m(t) = m$ is constant and (4) a maximum learning rate stability condition $\eta(t) \leq \eta_{\max}$ [2], the loss for large $T \gg 1$ under the optimal learning rate schedule minimizes the following functional of the integrated learning rate $\chi(t)$*

$$L_T - \sigma^2 \sim \underbrace{\chi(0)^{-\frac{a-1}{b}}}_{\textit{Bias}} + \underbrace{\frac{\sigma^2}{m} \int_0^T dt\, \dot{\chi}(t)^2\, \chi(t)^{-2+\frac{1}{b}}}_{\textit{Variance}} \qquad \underbrace{\chi(t) \coloneqq \int_t^T ds\, \eta(s)}_{\textit{Integrated Learning Rate}} \tag{10}$$

*From Noether's theorem Gel'fand and Fomin [2000], the optimal schedule satisfies $\dot{\chi}(t) \propto -\chi(t)^{(2b-1)/2b}$ with boundary condition $\chi(T) = 0$. The optimal solution exhibits two phases. In the **Easy Phase** ($a > b$), the optimal learning rate schedule is*

$$\textit{Easy Phase:} \quad \eta_T^*(t) \sim c(a,b)\ T^{b/a-1}\ f\left(\frac{t}{T}\right)\ ,\ f(z) = (1-z)^{2b-1} \tag{11}$$

*where $c(a, b)$ is a constant. In the **Hard Phase** ($a < b$), the optimal learning rate saturates the stability bound $\eta = \eta_{\max}$ for early times $t < t_s$ before subsequently annealing*

$$\textit{Hard Phase:} \quad \eta_T^*(t) \sim \begin{cases} \eta_{\max} & t < t_s\,, \\ \eta_{\max}\left(\frac{1-t/T}{1-t_s/T}\right)^{2b-1} & t > t_s\,, \end{cases} \tag{12}$$

where the fraction of time spent annealing scales sublinearly with T as $1 - t_s/T \sim T^{-\frac{b-a}{2b-1}}$. For derivations and exact constants, consult Appendix A.

Our analysis revealed two distinct scaling regimes depending on the parameters a and b. Following the terminology of Pillaud-Vivien et al. [2018], we distinguish between an *easy phase* ($b < a$), where the constraint is asymptotically inactive, and a *hard phase* ($b > a$), where it plays a critical role[3]. The type of schedule that we identify as optimal in the easy phase is commonly known as *polynomial decay* and has been used in a variety of applications Wu and Liu [2023], while the optimal protocol in the hard phase is a *constant-then-decay* schedule. It corresponds to the stable and decay phases of the *warmup-stable-decay* (WSD) schedule Hu et al. [2024] but has no warmup. Within our model, adding a warmup over a fixed fraction of training does not change the scaling exponent (Appendix H). We next show that the scaling exponents for optimized schedules are improved relative to the constant-η baseline. In Figure 2 we illustrate these two phases and plot both the exact solution to the optimal control problem, obtained numerically, and show that our scaling exponents derived from large T asymptotics accurately capture the scaling laws for each schedule.

Corollary 1 (Scaling Law Under the optimal LR Schedule) *The scaling law of the loss with T is*

$$L_T - \sigma^2 \sim T^{-(a-1)/\max(a,b)} \tag{13}$$

which outperform the scaling exponents for optimal constant LR. In the easy phase, this rate matches the minimax-optimal exponent $T^{-(a-1)/a}$ Caponnetto and De Vito [2007].

Interpretation In the easy phase ($a > b$), the optimal schedule balances the contribution of the bias and variance components. In the hard phase ($b > a$), the slow decay of the bias implies that the unconstrained optimal learning rate would grow as $T^{b/a-1}$ for large T. This divergence activates the stability constraint $\eta_T(t) \leq \eta_{\max}$, resulting in the constant-then-decay profile. As shown in Figure 5, while the easy phase schedule gradually decays to minimize both error components simultaneously, the hard phase strategy dedicates the majority of the training budget to maximizing bias suppression via a constant, maximal learning rate, compressing the variance-reducing annealing into a vanishing

[2]The bound $\eta_{\max} = \Theta(\lambda_1^{-1})$ reflects the stability of the leading modes of the exact recursion in Theorem 1. CasADi enforces it through the dynamics rather than as an explicit constraint, yielding smooth schedules that approximate the analytic plateau-then-decay shape and recover the predicted annealing-fraction scaling (see Fig. 6).

[3]We note that this easy/hard distinction is focused on the behavior of label noise SGD and differs from other notions of easy and hard phases such as diverging RKHS norm Paquette et al. [2024], Bordelon et al. [2025].

fraction of the total training time. In both regimes, a critical implication is that the optimal schedule relies on a fixed training horizon T. Anytime protocols, which are agnostic to the total budget, cannot effectively postpone variance minimization. We summarize all of the exponents for these settings in Table 1. The advantage of horizon-dependent schedules has been previously observed in convex optimization Jain et al. [2019], Defazio et al. [2023]. Previous works have rationalized the effectiveness of the WSD schedule by conjecturing a "river valley" loss landscape Wen et al. [2024], Liu et al. [2025]. In this picture, the landscape contains flat directions (the river) and steep directions (the hillsides). The stable phase makes progress along the river, while the decay phase suppresses the oscillations across the hillsides and reveals the progress made. Here, we show that a constant-then-decay schedule is optimal in the hard phase of our model, providing an alternative explanation for its effectiveness based on balancing demands of bias and variance reduction. The scaling laws in Eq. (13) can be combined with the dependence of σ^2 on the model size N to derive the compute-optimal allocation of N and T at fixed compute $C = mNT$; see Appendix B.2.

3.3 Comparison with benchmarks

Phase	Constant LR	Powerlaw	Opt. LR Schedule	Opt. LR + Batch Schedule
Easy ($a > b$)	$\frac{a-1}{a-1+b}$	$\frac{a-1}{a-1+b}$	$\frac{a-1}{a}$	$\frac{a-1}{b}$
Hard ($a < b$)	$\frac{a-1}{a-1+b}$	$\frac{a-1}{a-1+b}$	$\frac{a-1}{b}$	$\frac{a-1}{b}$

Table 1: Comparison of scaling exponents ζ_T for the loss as a function of training horizon $L_T - \sigma^2 \sim T^{-\zeta_T}$ for different types of schedules. Constant LR refers to $\eta_T(t) \sim T^{-\xi}$ and powerlaw refers to $\eta_T(t) \sim T^{-\xi} t^{-\delta}$, where the exponents ξ and δ are chosen optimally in each case, and optimal is the schedule we derive from optimal control. In the easy phase the batch schedule improves the wall-clock exponent from $\frac{a-1}{a}$ to $\frac{a-1}{b}$, while leaving the sample-budget exponent $\frac{a-1}{a}$ unchanged.

Having established that the optimal schedule improves the scaling exponent relative to any constant learning rate strategy, we extend our comparison to a broader class of protocols. In Appendix C, we analyze power-law schedules of the form $\eta_T(t) \sim T^{-\xi} t^{-\delta}$, optimizing the exponents ξ and δ to minimize the final loss. We demonstrate that even the best-performing power-law schedule asymptotically recovers the same scaling behavior as the optimal constant-η strategy. Consequently, our derived optimal schedule strictly outperforms the entire family of power-law benchmarks (see Table 1). Our schedules saturate information theoretically optimal rates in the easy phase, but not the hard phase where $b > a$ Fischer and Steinwart [2020]. Pillaud-Vivien et al. [2018] showed that a combination of data repetition and model averaging can improve the rates in the hard phase to saturate the theoretically optimal rate.

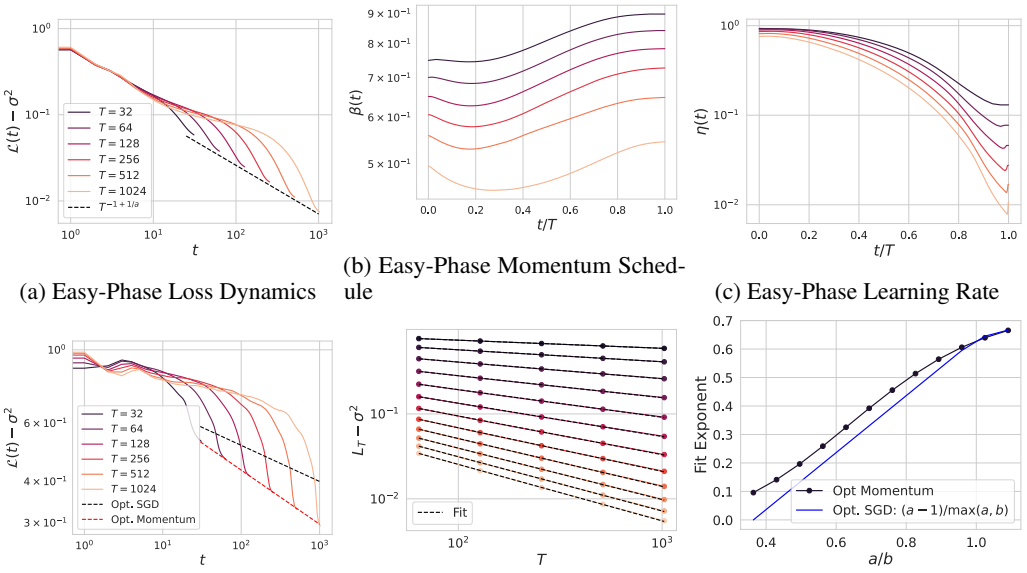
Generalizing beyond the specific optimal schedule derived in Result 2, we investigate the sensitivity of the performance to the choice of the profile function within the general scaling $\eta_T(t) = T^{-\xi} g\left(t/T\right)$. In Appendix C, we show that the specific shape of $g(z)$ does not change the scaling exponent, provided it decays faster than $(1-z)^{b-1}$ as $z \to 1$. The prefactor exponent ξ must be chosen as $\xi = \max[0, 1 - b/a]$ (using a different value would result in suboptimal scaling). Under these conditions, the generic schedule recovers the optimal scaling exponents in both the easy and hard regimes, differing from the optimal error only by a constant prefactor.

3.4 Joint optimization of learning rate and batch size

Our analysis extends naturally to the joint optimization of the learning rate $\eta(t)$ and minibatch size $m(t)$ under a fixed data budget $B_{\rm tot} = \int_0^T dt\, m(t)$, with the wall-clock time T as a free variable.

Result 3 (Joint LR and Batch Schedules) *Under the assumptions of Result 2, optimizing L_T jointly over $\eta(t)$ and $m(t)$ subject to $B_{\rm tot} = \int_0^T dt\, m(t)$ yields a degenerate family of optima with $m(t) \propto -\dot{\chi}(t)\, \chi(t)^{-1+1/(2b)}$, all reaching the same loss*

$$L_T - \sigma^2 \sim B_{\rm tot}^{-(a-1)/a} \;\; (b < a)\,, \qquad L_T - \sigma^2 \sim B_{\rm tot}^{-(a-1)/b} \;\; (b > a)\,. \tag{14}$$



(a) Easy-Phase Loss Dynamics (b) Easy-Phase Momentum Schedule (c) Easy-Phase Learning Rate

(d) Hard-Phase Loss Dynamics (e) Opt. Momentum Varying a/b (f) Improved Scaling in Hard Phase

Figure 3: Optimal Schedule and loss dynamics for SGD + momentum. (a) For the easy task regime $(a, b) = (2.3, 1.5)$, the numerically optimized schedule achieves the same scaling law as SGD with optimal schedule $L_T - \sigma^2 \sim T^{-1+1/a}$. (b) The optimal momentum dynamics vary significantly across T but only weakly vary with t. (c) The learning rate for optimal momentum schedules anneals similarly to SGD in the easy phase. (d) In the hard phase $(a, b) = (1.3, 2.8)$, the scaling law for the loss obtained by jointly optimizing momentum and learning rate is better than the SGD rate $T^{-(a-1)/b}$. (e)-(f) We fit scaling laws for the optimal momentum schedule $L_T - \sigma^2 \sim T^{-\xi}$ for varying (a, b) and plot the fit exponent compared to our predictions for SGD. For $a < b$, momentum outperforms SGD.

The degeneracy is broken, following Meterez et al. [2025], by demanding minimum wall-clock time, which fixes $\eta(t) = \eta_{\max}$ and yields an increasing batch schedule

$$m_T^*(t) = \frac{B_{\text{tot}}}{2bT}\left(1 - \frac{t}{T}\right)^{1/(2b)-1}, \qquad T \sim B_{\text{tot}}^{b/a} \;\; (b < a)\,, \quad T \sim B_{\text{tot}} \;\; (b > a)\,. \tag{15}$$

This type of procedure, known as a *batch ramp*, has been empirically proposed Smith et al. [2017] but is derived here through optimal control principles. In the easy phase, the batch ramp accelerates training in wall-clock time: the loss decays as $L_T - \sigma^2 \sim T^{-(a-1)/b}$ rather than $T^{-(a-1)/a}$ at constant batch, while the sample-budget exponent $(a-1)/a$ is unchanged (see Fig. 8). In the hard phase, both the wall-clock and sample-budget exponents coincide with the fixed-batch optimum. Since Theorem 1 holds for any batch schedule, a maximum batch size $m_{\max}$ can be imposed in the same framework, which simply truncates the batch ramp.

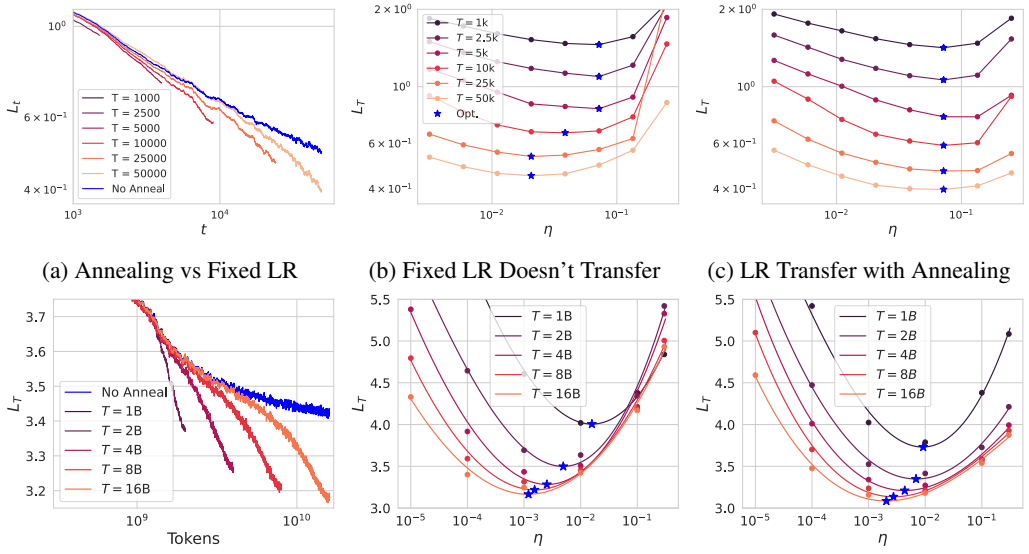
3.5 Optimal Momentum Schedules Compared to SGD

We also study momentum with time varying learning rates $\{\eta_t\}$ and momentum $\{\beta_t\}$ parameters which generate the update equations for parameters $\boldsymbol{w}_t$

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \eta_t \boldsymbol{v}_t\,,\; \boldsymbol{v}_t = (1-\beta_t)\boldsymbol{v}_{t-1} + \beta_t \boldsymbol{g}_t\,,\; \boldsymbol{g}_t \coloneqq \frac{1}{m_t}\sum_{\mu=1}^{m_t} \tilde{\boldsymbol{\psi}}(\boldsymbol{x}_{\mu,t})\left(y(\boldsymbol{x}_{\mu,t}) - \hat{y}(\boldsymbol{x}_{\mu,t})\right)\,.$$

We optimize the terminal loss L_T over the choices of $\{(\eta_t, \beta_t)\}_{t=1}^T$. Using similar methods to the analysis we used for SGD in Theorem 1, we can compute the test loss by tracking three correlation variables $\{C_{k,t}, R_{k,t}, V_{k,t}\}$ for each eigendirection k. These variables are defined as

$$C_{k,t} = \left\langle (w_{k,t} - w_k^\star)^2 \right\rangle\,, \qquad R_{k,t} = \left\langle v_{k,t}(w_k^\star - w_{k,t}) \right\rangle\,, \qquad V_{k,t} = \left\langle (v_{k,t})^2 \right\rangle \tag{16}$$



(a) Annealing vs Fixed LR (b) Fixed LR Doesn't Transfer (c) LR Transfer with Annealing

(d) LLM Constant vs Annealing (e) Fixed LR (f) Annealing Needs Tuned Base LR

Figure 4: Hard and easy phase behaviors in different deep learning settings. (a) ResNet CNNs trained with SGD on CIFAR-5M. Cross-entropy loss dynamics as a function of training time t for fixed learning rate (blue) and polynomial annealing schedule set to training horizon T. The final loss follows a better trend when using the annealing schedule. (b) Loss as a function of (η, T) and for fixed learning rate. The optimal learning rate decreases with T. (c) For polynomial annealing schedules, the optimal base learning rate is approximately constant across horizons T, consistent with our model in the hard phase. Further, the best performance achieved at T iterations is noticeably improved compared to constant learning rate. (d)-(f) The same plots but for GPT-2 transformer pretraining on C4. In this case, both the fixed learning rate and annealing strategies require decreasing the base learning rate with T.

which collectively evolve as a linear dynamical system that depends on $\lambda_k, w_k^\star, \eta_t, \beta_t$ and m. From this dynamical system, we can obtain the loss dynamics $L_T - \sigma^2 = \sum_k \lambda_k C_{k,T}$, enabling use of optimal control. Recent work has demonstrated that allowing β to depend on time can improve scaling laws in this powerlaw random feature model Ferbach et al. [2025]. Instead of a preset schedule for β_t, we investigate the *optimal control* policy for η_t and β_t. We plot the result for easy and hard tasks in Figure 3, finding it differs from the anytime powerlaw schedule used by Ferbach et al. [2025]. We find that the optimal control learning rate for easy tasks exhibits phenomenology remarkably similar to optimal SGD. However, for hard tasks, we find that optimizing the momentum schedule β_t in addition to the learning rate schedule η_t provides an improvement to the scaling law with T in the hard phase ($a < b$). The optimal momentum variables β_t in either case show an initial transient decrease before a final increase as $t/T \to 1$.

4 Experiments in Deep Networks

These experiments use nonlinear networks and, for language modeling, Adam, and should be read as qualitative illustrations of the predicted phenomenology.

Computer Vision We study a simple setup of CIFAR-5M (an extended version of CIFAR-10 that allows one-pass training) image classification in a deep convolutional ResNet Nakkiran et al. [2020]. We train with SGD using either fixed learning rate or a polynomial decay schedule, consistent with our optimal schedule. In Figure 4, we plot a comparison of the learning rate transfer for fixed learning rate schedules compared to polynomial decay. In Figure 4 (a) we show that the loss dynamics L_t for fixed learning rate (blue) can be outperformed at the same base learning rate by annealing with a schedule that decays to zero as $t/T \to 1$. In Figure 4 (b), we plot the optimal learning rate as a function of T for constant learning rate, which shifts left. Lastly, when using an annealing schedule

we find that the optimal learning rate is approximately constant and the best performance at each value of T is improved compared to fixed learning rate. These observations are consistent with the behavior of our theoretical model in the hard phase since the optimal base learning rate of an annealing schedule is approximately constant. Based on the spectral analysis of the NTK for ResNets on CIFAR-5M from Bordelon et al. [2024], it is reasonable to expect this problem-architecture pair to fall in the hard phase. We note that our theory *does not generally predict transfer of base learning rates*, since in the easy phase, the optimal base learning rate will still scale as a powerlaw in T.

Language Modeling We also find evidence for easy-phase behavior in GPT-style transformers trained on next word prediction on natural language. The models have 125M non-embedding parameters and are trained on the C4 corpus Raffel et al. [2020] with Adam with varying learning rate and default β. In Figure 4 (d) we plot the influence of cosine annealing schedules on the loss dynamics at the near optimal learning rate. In Figure 4 (e)-(f) we illustrate that for both fixed learning rate and annealing, the optimal base learning rate decreases with T. The persistence of a shift in the optimal base learning rate under annealing is compatible the *easy phase* in our theory.

5 Conclusions

We theoretically analyzed the optimal learning rate schedule for a simple random feature model. Our results utilized optimal control methods, which enabled optimization over the full trajectory of the learning dynamics. This model recapitulated schedules often observed in practice including polynomial decay and constant-then-decay schedules. Further, we explored optimal batch size and optimal momentum schedules, finding improvements in wall-clock scaling and sample-budget scaling, respectively.

Limitations and Future Directions While our analysis provided some insights into the behavior of SGD under different annealing schedules and different phases of task difficulty, there are several limitations to the present work. First, our current theory is focused on describing linear models trained with SGD. This can only capture optimization behaviors of neural networks operating in the lazy/kernel regime and also fails to capture the behavior of more sophisticated optimizers such as Adam or preconditioned optimizers like Muon, Shampoo, or SOAP Kingma and Ba [2014], Jordan et al. [2024], Gupta et al. [2018], Vyas et al. [2024]. Second, while our numerical solution to optimal momentum revealed an improvement in the scaling law exponent, we have not yet identified analytically what this improved exponent is for the hard phase. Third, we restricted to the top-N projection $\boldsymbol{G} = [\boldsymbol{I}_N, \boldsymbol{0}]$, and a full analysis of Gaussian sketches is left for future work. Lastly, while we provided experimental evidence for easy and hard phase problems in different deep learning settings, it would be interesting to develop ways to empirically access the correct spectral statistics to predict the optimal schedule shape to achieve better performance.

Broader Impacts This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

Acknowledgements

The authors thank Alex Atanasov, Francesca Mignacco, and Clarissa Lauditi for useful feedback on the manuscript. This work was supported by the Center of Mathematical Sciences and Applications (CMSA) at Harvard University. B.B. acknowledges the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing resources that have contributed to the research results reported within this paper, specifically allocations DMS26007 and DMS26010 on the Lonestar6-GPU system. The authors declare no competing interests.

Use of AI Tools

We used AI assistants for minor editing of the text and to produce a first version of the proof of Theorem 2. The authors have checked all AI-generated content and take full responsibility for it.

References

Joel AE Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. Casadi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences*, 121(27):e2311878121, 2024.

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 Cengiz Pehlevan. Learning curves for SGD on structured features. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=WPI2vbkAl3Q`.

Blake Bordelon, Lorenzo Noci, Mufan Bill Li, Boris Hanin, and Cengiz Pehlevan. Depthwise hyperparameter transfer in residual networks: Dynamics and scaling limit. *arXiv preprint arXiv:2309.16620*, 2023.

Blake Bordelon, Alexander Atanasov, and Cengiz Pehlevan. A dynamical model of neural scaling laws. In *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=nbOY1OmtRc`.

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

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

Rodrigo Carrasco-Davis, Javier Masís, and Andrew M Saxe. Meta-learning strategies through value maximization in neural networks. *arXiv preprint arXiv:2310.19919*, 2023.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019.

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

Nolan Dey, Bin Claire Zhang, Lorenzo Noci, Mufan Li, Blake Bordelon, Shane Bergsma, Cengiz Pehlevan, Boris Hanin, and Joel Hestness. Don't be lazy: Completep enables compute-efficient deep transformers. *arXiv preprint arXiv:2505.01618*, 2025.

Aymeric Dieuleveut and Francis Bach. Non-parametric stochastic approximation with large step sizes. *Annals of Statistics*, 44(4), 2015.

Katie Everett, Lechao Xiao, Mitchell Wortsman, Alexander A Alemi, Roman Novak, Peter J Liu, Izzeddin Gur, Jascha Sohl-Dickstein, Leslie Pack Kaelbling, Jaehoon Lee, et al. Scaling exponents across parameterizations and optimizers. *arXiv preprint arXiv:2407.05872*, 2024.

Damien Ferbach, Katie Everett, Gauthier Gidel, Elliot Paquette, and Courtney Paquette. Dimension-adapted momentum outscales sgd. *arXiv preprint arXiv:2505.16098*, 2025.

Simon Fischer and Ingo Steinwart. Sobolev norm learning rates for regularized least-squares algorithms. *Journal of Machine Learning Research*, 21(205):1–38, 2020.

I. M. Gel'fand and S. V. Fomin. *Calculus of Variations*. Dover Publications, 2000.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pages 1842–1850. PMLR, 2018.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, DDL Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 10, 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.

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.

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

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 22, 2014.

Binghui Li, Fengling Chen, Zixun Huang, Lean Wang, and Lei Wu. Unveiling the role of learning rate schedules via functional scaling laws. *arXiv e-prints*, 2025.

Junhong Lin and Lorenzo Rosasco. Optimal rates for multi-pass stochastic gradient methods. *Journal of Machine Learning Research*, 18(97):1–47, 2017.

Licong Lin, Jingfeng Wu, Sham M. Kakade, Peter Bartlett, and Jason D. Lee. Scaling laws in linear regression: Compute, parameters, and data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=PH7sdEanXP`.

Ziming Liu, Yizhou Liu, Jeff Gore, and Max Tegmark. Neural thermodynamic laws for large language model training. *arXiv preprint arXiv:2505.10559*, 2025.

Alexandru Meterez, Depen Morwani, Jingfeng Wu, Costin-Andrei Oncescu, Cengiz Pehlevan, and Sham Kakade. Seesaw: Accelerating training by balancing learning rate and batch size scheduling. *arXiv preprint arXiv:2510.14717*, 2025.

Francesca Mignacco and Francesco Mori. A statistical physics framework for optimal learning. *arXiv preprint arXiv:2507.07907*, 2025.

Francesco Mori, Stefano Sarao Mannelli, and Francesca Mignacco. Optimal protocols for continual learning via statistical physics and control theory. *Journal of Statistical Mechanics: Theory and Experiment*, 2025(8):084004, 2025.

Preetum Nakkiran, Behnam Neyshabur, and Hanie Sedghi. The deep bootstrap framework: Good online learners are good offline generalizers. *arXiv preprint arXiv:2010.08127*, 2020.

Valentina Njaradi, Rodrigo Carrasco-Davis, Peter E Latham, and Andrew Saxe. Optimal learning rate schedule for balancing effort and performance. *arXiv preprint arXiv:2601.07830*, 2026.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Michal Guerquin, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2 olmo 2 furious, 2024. URL `https://arxiv.org/abs/2501.00656`.

Elliot Paquette, Courtney Paquette, Lechao Xiao, and Jeffrey Pennington. 4+3 phases of compute-optimal neural scaling laws. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=aVSxwicpAk`.

Loucas Pillaud-Vivien, Alessandro Rudi, and Francis Bach. Statistical optimality of stochastic gradient descent on hard learning problems through multiple passes. *Advances in Neural Information Processing Systems*, 31, 2018.

Shikai Qiu, Lechao Xiao, Andrew Gordon Wilson, Jeffrey Pennington, and Atish Agarwala. Scaling collapse reveals universal dynamics in compute-optimally trained neural networks. *arXiv preprint arXiv:2507.02119*, 2025.

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`.

Magnus Rattray and David Saad. Analysis of on-line training with optimal learning rates. *Physical Review E*, 58(5):6379, 1998.

David Saad and Magnus Rattray. Globally optimal parameters for on-line learning in multilayer neural networks. *Physical review letters*, 79(13):2578, 1997.

Samuel L Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V Le. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017.

Aditya Vardhan Varre, Loucas Pillaud-Vivien, and Nicolas Flammarion. Last iterate convergence of sgd for least-squares in the interpolation regime. *Advances in Neural Information Processing Systems*, 34:21581–21591, 2021.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Mujin Kwun, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. Soap: Improving and stabilizing shampoo using adam. *arXiv preprint arXiv:2409.11321*, 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.

Yanzhao Wu and Ling Liu. Selecting and composing learning rate policies for deep neural networks. *ACM Transactions on Intelligent Systems and Technology*, 14(2):1–25, 2023.

Greg Yang, Edward J Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv preprint arXiv:2203.03466*, 2022.

Greg Yang, Dingli Yu, Chen Zhu, and Soufiane Hayou. Feature learning in infinite-depth neural networks. In *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*, 2023.

A Long-time dynamics

This appendix derives the closed-form expression for the test loss used in the optimal-control analysis of the main text. The derivation is carried out at leading order in large T and underlies Results 1–3. Its controlled counterpart, which keeps all terms of the exact recursion, is Theorem 2, proved in Appendix F.

Assumptions. The starting point is the exact recursion for $c_{t,k}$ given in Eq. (5). Throughout this appendix we use:

- **(A1) Gaussian features.** The features $\boldsymbol{\psi}(\boldsymbol{x})$ are zero-mean Gaussian. Eq. (5) follows rigorously under this assumption Bordelon and Pehlevan [2022] and extends to non-Gaussian features under mild moment conditions.
- **(A2) Power-law spectra.** $\lambda_k = k^{-b}$ and $(w_k^*)^2\lambda_k = k^{-a}$, with $a, b > 1$.
- **(A3) Zero initialization.** $\boldsymbol{w}_0 = \mathbf{0}$, hence $c_{0,k} = (w_k^*)^2$.
- **(A4) Late-time / small-eigenvalue regime.** The horizon T is large and we restrict to times t for which the loss is dominated by modes with $\eta_t\lambda_k \ll 1$.
- **(A5) Bounded label noise.** σ_0 is bounded away from zero, so that σ^2 defined in Eq. (4) stays of order one as $T \to \infty$.
- **(A6) Stability bound.** $\eta(t) \le \eta_{\max}$, where $\eta_{\max} = \mathcal{O}(\lambda_1^{-1})$ is set by the stability of the exact recursion in Eq. (5) and does not depend on T (Theorem 2 gives an explicit sufficient condition). This bound is automatic when optimizing the exact dynamics; it must be reintroduced explicitly here because (A4) drops the leading-λ_k modes that would otherwise enforce it.

From Eq. (5),

$$c_{t+1,k} = \left(1 - 2\eta_t\lambda_k + \eta_t^2\frac{m_t+1}{m_t}\lambda_k^2\right)c_{t,k} + \frac{\eta_t^2}{m_t}\lambda_k\sum_{\ell=1}^{N}\lambda_\ell c_{t,\ell} + \frac{\eta_t^2}{m_t}\sigma^2\lambda_k\,. \tag{17}$$

Under (A4), the term $\eta_t^2\lambda_k^2\,c_{t,k}$ is subleading compared to $\eta_t\lambda_k\,c_{t,k}$ and is dropped. Using the identity $\sum_\ell \lambda_\ell c_{t,\ell} = L_t - \sigma^2$ to absorb the second and third terms into a single L_t, we obtain the linear recursion

$$c_{t+1,k} \approx (1 - 2\eta_t\lambda_k)\,c_{t,k} + \frac{\eta_t^2}{m_t}\lambda_k\,L_t\,. \tag{18}$$

Equation (18) is a linear recursion in $c_{t,k}$ with t-dependent coefficients. Its solution reads Bordelon and Pehlevan [2022]

$$c_{T,k} = c_{0,k}\prod_{i=0}^{T-1}(1 - 2\eta_i\lambda_k) + \lambda_k\sum_{n=0}^{T-1}\frac{\eta_n^2}{m_n}L_n\prod_{i=n+1}^{T-1}(1 - 2\eta_i\lambda_k)\,. \tag{19}$$

Under (A4), $\log(1 - 2\eta_i\lambda_k) \approx -2\eta_i\lambda_k$, so that

$$\prod_{i=n+1}^{T-1}(1 - 2\eta_i\lambda_k) \approx \exp\left(-2\lambda_k\sum_{i=n+1}^{T-1}\eta_i\right) \approx e^{-2\lambda_k\chi(t_n)}\,, \tag{20}$$

where we have introduced the integrated learning rate

$$\chi(t) := \int_t^T \eta(t')\,dt'\,, \qquad \chi_0 := \chi(0)\,. \tag{21}$$

Replacing the sum over n by an integral and using (A3) to set $c_{0,k} = (w_k^*)^2$,

$$c_{T,k} \approx (w_k^*)^2\,e^{-2\lambda_k\chi_0} + \lambda_k\int_0^T dt\;\frac{\eta(t)^2}{m(t)}\,L_t\,e^{-2\lambda_k\chi(t)}\,. \tag{22}$$

Define

$$f_s(\chi) := \sum_{k=1}^{N} k^{-s} e^{-k^{-b}\chi}\,. \tag{23}$$

Using (A2), $\sum_k \lambda_k (w_k^*)^2 e^{-2\lambda_k \chi_0} = f_a(2\chi_0)$ and $\sum_k \lambda_k^2 e^{-2\lambda_k \chi(t)} = f_{2b}(2\chi(t))$. Therefore

$$L_T - \sigma^2 = \sum_k \lambda_k c_{T,k} \approx f_a(2\chi_0) + \int_0^T dt\, \frac{\eta(t)^2}{m(t)}\, L_t\, f_{2b}(2\chi(t))\,. \tag{24}$$

The function f_s satisfies $df_s/d\chi = -f_{s+b}(\chi)$ and $f_s(0) = \sum_{k=1}^{N} k^{-s} \xrightarrow[N\to\infty]{} \zeta(s)$, with the large-argument asymptotic

$$f_s(\chi) \approx \frac{\Gamma((s-1)/b)}{b}\, \chi^{(1-s)/b}\,, \qquad \chi \to \infty\,. \tag{25}$$

Equation (24) is implicit in L_t. Two observations close it. First, under (A5) the excess loss decays towards zero while σ^2 stays of order one, so $L_t - \sigma^2 \ll \sigma^2$ at late times. Second, the integrand in Eq. (24) carries the factor $f_{2b}(2\chi(t))$, which is monotonically decreasing in χ and hence concentrates the integral on times near $t = T$ where $\chi(t)$ is small; this is exactly the regime in which $L_t \approx \sigma^2$ holds with the smallest error. Substituting $L_t \to \sigma^2$ inside the integral yields the form used in the rest of the paper:

$$L_T - \sigma^2 \approx f_a(2\chi_0) + \sigma^2 \int_0^T dt\, \frac{\eta(t)^2}{m(t)}\, f_{2b}(2\chi(t))\,. \tag{26}$$

B Analytical derivation of optimal learning-rate schedules

B.1 Power-law spectra

For constant batch size $m(t) = m$ and the closure of Eq. (26), the optimal-control problem reduces to finding a non-increasing trajectory $\chi(t)$ with $\chi(T) = 0$ that minimizes

$$L_T - \sigma^2 \approx f_a(2\chi_0) + \int_0^T dt\, \mathcal{L}(\dot\chi(t), \chi(t))\,, \qquad \mathcal{L}(\dot\chi, \chi) = \dot\chi^2\, A(\chi)\,, \qquad A(\chi) := \frac{\sigma^2}{m} f_{2b}(2\chi)\,, \tag{27}$$

where we used $\eta(t) = -\dot\chi(t)$.

Since $\mathcal{L}$ has no explicit t dependence, Noether's Theorem for time translations gives the conservation law $\frac{d}{dt}[\dot\chi\, \partial_{\dot\chi}\mathcal{L} - \mathcal{L}] = 0$, which yields

$$\dot\chi\, \partial_{\dot\chi}\mathcal{L} - \mathcal{L} = \dot\chi^2\, A(\chi) = c \tag{28}$$

for some positive constant c. Equivalently, since $\dot\chi < 0$,

$$\dot\chi(t) = -\sqrt{\frac{c}{A(\chi(t))}}\,. \tag{29}$$

Define

$$F(\chi) := \int_0^{\chi} d\chi'\, \sqrt{A(\chi')}\,. \tag{30}$$

Separating variables in Eq. (29) and integrating from t to T with $\chi(T) = 0$ gives $F(\chi(t)) = \sqrt{c}\,(T - t)$. At $t = 0$ this fixes $\sqrt{c} = F(\chi_0)/T$, hence

$$\frac{F(\chi(t))}{F(\chi_0)} = 1 - \frac{t}{T}\,. \tag{31}$$

Using the conservation law, the variance integral is

$$\int_0^T dt\, \dot\chi(t)^2\, A(\chi(t)) = c\,T = \frac{F(\chi_0)^2}{T}\,. \tag{32}$$

Therefore Eq. (27) along the optimal trajectory reduces to a function of χ_0 alone,

$$L_T - \sigma^2 \approx f_a(2\chi_0) + \frac{F(\chi_0)^2}{T}\,, \tag{33}$$

and the stationarity condition $\partial_{\chi_0}[L_T - \sigma^2] = 0$ gives

$$-2\, f_{a+b}(2\chi_0) + \frac{2\, F(\chi_0)\, F'(\chi_0)}{T} = 0\,, \tag{34}$$

where we used $df_a/d\chi = -f_{a+b}$ and $F'(\chi) = \sqrt{A(\chi)}$.

For large χ, the asymptotic in Eq. (25) gives

$$A(\chi) \approx \frac{\sigma^2}{m}\frac{\Gamma(2-1/b)}{b}\,(2\chi)^{(1-2b)/b}\,, \tag{35}$$

and substituting $u = 2\chi'$ in $F(\chi) = \int_0^\chi d\chi'\sqrt{A(\chi')}$, together with $\int_0^X u^{(1-2b)/(2b)}du = 2b\,X^{1/(2b)}$, yields

$$F(\chi) \approx \sqrt{\frac{b\,\sigma^2\Gamma(2-1/b)}{m}}\,(2\chi)^{1/(2b)}\,, \qquad F'(\chi_0) = \sqrt{A(\chi_0)} \approx \sqrt{\frac{\sigma^2\Gamma(2-1/b)}{b\,m}}\,(2\chi_0)^{(1-2b)/(2b)}\,. \tag{36}$$

The product $F(\chi_0)\,F'(\chi_0) \approx (\sigma^2/m)\,\Gamma(2-1/b)\,(2\chi_0)^{(1-b)/b}$, while $f_{a+b}(2\chi_0) \approx \Gamma((a+b-1)/b)(2\chi_0)^{(1-a-b)/b}/b$. Substituting into Eq. (34),

$$\frac{\Gamma((a+b-1)/b)}{b}\,(2\chi_0)^{(1-a-b)/b} = \frac{\sigma^2\,\Gamma(2-1/b)}{m\,T}\,(2\chi_0)^{(1-b)/b}\,. \tag{37}$$

Solving for χ_0 (the exponent of $2\chi_0$ on the left minus that on the right is $-a/b$),

$$\chi_0 \approx \frac{1}{2}\left[\frac{m}{b\,\sigma^2}\,\frac{\Gamma((a+b-1)/b)}{\Gamma(2-1/b)}\right]^{b/a} T^{b/a}\,. \tag{38}$$

For $b < a$ this gives $\chi_0 \sim T^{b/a}$ with $b/a < 1$, consistent with $\eta(0) \sim \chi_0/T \to 0$ as $T \to \infty$. For $b > a$, χ_0 grows superlinearly in T and $\eta(0)$ would diverge; the constraint $\eta(t) \le \eta_{\max}$ then becomes active and is treated below.

Easy phase. In the scaling limit $T \to \infty$ with t/T fixed, both $\chi(t)$ and $\chi_0 - \chi(t)$ are large and the asymptotic form of A above is valid. Substituting into Eq. (29),

$$\dot\chi(t) \propto -\chi(t)^{1-1/(2b)}\,. \tag{39}$$

Separating variables and imposing $\chi(T) = 0$,

$$\chi(t) \approx \chi_0\,(1-t/T)^{2b}\,, \qquad \eta(t) = -\dot\chi(t) \approx \frac{2b\,\chi_0}{T}\,(1-t/T)^{2b-1}\,, \tag{40}$$

which combined with Eq. (38) reproduces the closed-form schedule of the main text. Substituting Eq. (38) into $L_T - \sigma^2 \approx f_a(2\chi_0) + F(\chi_0)^2/T$ and using the asymptotic forms above, both terms scale as $T^{-(a-1)/a}$, yielding the easy-phase loss scaling.

Hard phase. In the hard phase, the constraint $\eta \le \eta_{\max}$ becomes active and we find the optimal schedule

$$\eta(t) = \begin{cases} \eta_{\max} & \text{for } t < t_s\,, \\ \eta_{\max}\left(\frac{1-t/T}{1-t_s/T}\right)^{2b-1} & \text{for } t > t_s\,, \end{cases} \tag{41}$$

where the switching time $0 < t_s < T$ can be determined by minimizing the generalization error. It is useful to define $\tilde\eta(\tau) = \eta(\tau T)$. Then, the loss can be written as

$$L_T - \sigma^2 \sim T^{-(a-1)/b}\left[\int_0^1 d\tau\;\tilde\eta(\tau)\right]^{-(a-1)/b} + \frac{\sigma^2}{m}T^{-(b-1)/b}\int_0^1 d\tau\;\tilde\eta(\tau)^2\left[\int_\tau^1 d\tau'\;\tilde\eta(\tau')\right]^{-\frac{2b-1}{b}}\,. \tag{42}$$

If $b > a$, the first term will dominate, implying $t_s \approx T$ to leading order. If $b < a$, the second term will dominate. As shown below, the second term is minimized by $t_s = 0$, therefore the constraint $\eta < \eta_{\max}$ is never saturated for $b < a$, in agreement with the easy-phase analysis above. We therefore focus on the case $b > a$ and compute the first-order corrections to $\tau_s := t_s/T \approx 1$.

Using Eq. (41), we find

$$L_T - \sigma^2 \sim T^{-(a-1)/b} f_{\text{Bias}}(\tau_s) + \frac{\sigma^2}{m} T^{-(b-1)/b} f_{\text{Var}}(\tau_s)\,, \tag{43}$$

where

$$f_{\text{Bias}}(\tau_s) = \left[\tau_s + \frac{1-\tau_s}{2b}\right]^{-(a-1)/b}\,, \tag{44}$$

and

$$f_{\text{Var}}(\tau_s) = \left[\frac{b}{b-1}\left(\left(\frac{1-\tau_s}{2b}\right)^{-(b-1)/b} - \left(\tau_s + \frac{1-\tau_s}{2b}\right)^{-(b-1)/b}\right) + (2b)^{(2b-1)/b}(1-\tau_s)^{-(b-1)/b}\right] \tag{45}$$

Note that the variance second term is an increasing function of τ_s, confirming that in the easy phase $b < a$, the constraint $\eta(t) < \eta_{\max}$ is never saturated. In the hard case $b > a$, we consider corrections $\tau_s \approx 1 - \epsilon$ and expand to leading order for small ϵ, yielding (up to constants)

$$L_T - \sigma^2 \sim T^{-(a-1)/b} + T^{-(a-1)/b}\epsilon + T^{-(b-1)/b}\epsilon^{-(b-1)/b}\,. \tag{46}$$

Minimizing this expression over ϵ, we identify the scaling

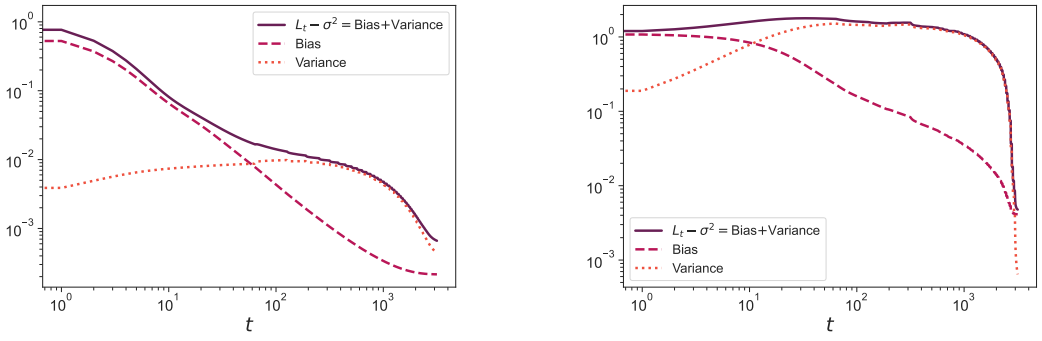
$$\epsilon \sim T^{-(b-a)/(2b-1)}\,. \tag{47}$$

Overall, this argument predicts that

$$t_s \approx T(1 - cT^{-(b-a)/(2b-1)}) \tag{48}$$

for some constant c. Figure 6 shows that this prediction agrees with the result obtained by optimizing the full dynamics with CasADi (in that case, for a schedule $\eta(t)$ we define t_s as the first time at which $\eta(t) = 0.15$). As a result, the loss scales as

$$L_T - \sigma^2 \sim T^{-(a-1)/b} + \sigma^2 T^{-(b-1)/b}\epsilon^{-(b-1)/b} \sim T^{-(a-1)/b} + \sigma^2 T^{-(b-1)(a+b-1)/(b(2b-1))}\,. \tag{49}$$



(a) Easy Phase ($b < a$)

(b) Hard Phase ($b > a$)

Figure 5: **Decomposition of the excess loss $L_t - \sigma^2$ into bias and variance components. (a)** In the easy phase ($b = 2$, $a = 3.5$), the optimal schedule minimizes bias and variance simultaneously throughout the training trajectory. **(b)** In the hard phase ($b = 5$, $a = 3.5$), the schedule minimizes the bias for the majority of the training time ($t < t_s$) where the learning rate is large, while the final annealing phase ($t > t_s$) is responsible for suppressing the variance. **Parameters:** $T = 3162$, $N = 1000$, $\sigma = 0.5$, $m = 5$.

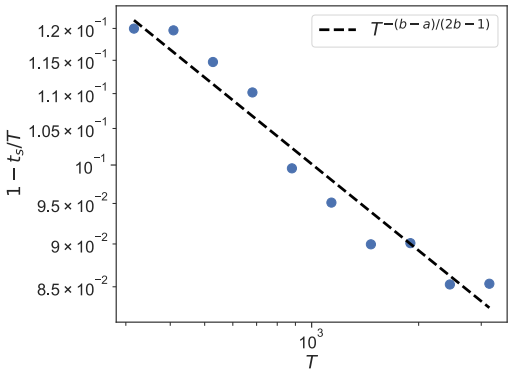
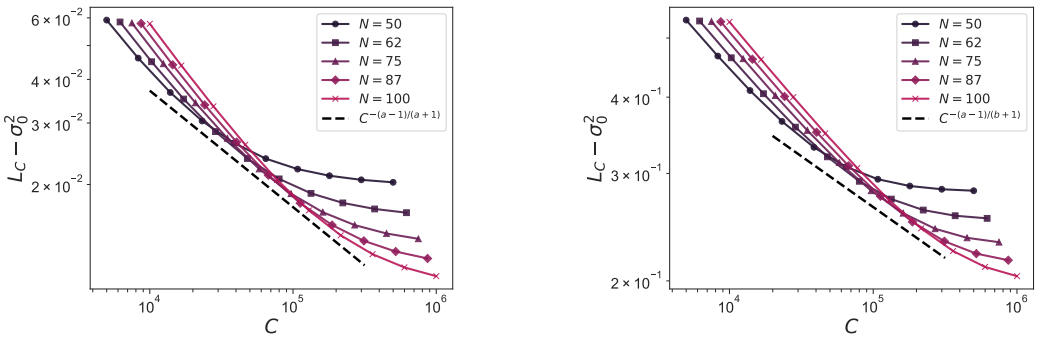


Figure 6: Fraction $1 - t_s/T$ of the total training time spent in the annealing regime. The switching time t_s is defined here as the time at which the schedule $\eta^*(t)$ crosses the level $\eta = 0.15$ for the first time. Same parameters as in Fig. 2a.



(a) Easy Phase ($b < a$)

(b) Hard Phase ($b > a$)

Figure 7: **Compute optimal scaling.** Residual loss $L_C - \sigma_0^2$ as a function of the compute $C = mNT$ for different values of the model size N. The dashed lines indicate the theoretical prediction. **Parameters:** $\sigma = 0.5$, $m = 5$. In the easy phase $b = 1.5$ and $a = 2$, in the hard phase $b = 2$ and $a = 1.5$.

B.2 Compute-optimal scaling

We next address the problem of jointly scaling the model size N and the data budget B_{tot} to minimize the generalization error L_C under a fixed compute budget $C = B_{\text{tot}}N$. Using Result 2, for $M \gg N \gg 1$, the effective noise variance scales as $\sigma^2 \sim \sigma_0^2 + N^{1-a}$. Consequently, in the easy phase, the excess loss scales as $L_T - \sigma_0^2 \sim B_{\text{tot}}^{-(a-1)/a} + N^{1-a}$. Balancing these terms yields the compute-optimal allocation $B_{\text{tot}} \sim C^{a/(a+1)}$ and $N \sim C^{1/(a+1)}$. This results in an optimal error scaling of $L_C - \sigma_0^2 \sim C^{-(a-1)/(a+1)}$. Similarly, in the hard phase, the loss scales as $L_T - \sigma_0^2 \sim B_{\text{tot}}^{-(a-1)/b} + N^{1-a}$. Optimizing this tradeoff leads to $B_{\text{tot}} \sim C^{b/(b+1)}$ and $N \sim C^{1/(b+1)}$, yielding a final error scaling of $L_C - \sigma_0^2 \sim C^{-(a-1)/(b+1)}$. These theoretical scaling laws are verified in Fig. 7.

C Comparison with benchmarks

C.1 Constant learning rate

For a constant learning rate $\eta_T(t) = \eta_T$ we have $\chi(t) = \eta_T(T-t)$, with $\chi_0 = \eta_T T$. Substituting into Eq. (26) (with $m(t) = m$),

$$L_T - \sigma^2 \approx f_a(2\eta_T T) + \frac{\sigma^2\,\eta_T^2}{m}\int_0^T dt\; f_{2b}(2\eta_T(T-t))\,. \tag{50}$$

Changing variables $s = T - t$ and then $\chi' = 2\eta_T s$,

$$\int_0^T dt\; f_{2b}(2\eta_T(T-t)) = \int_0^T ds\; f_{2b}(2\eta_T s) = \frac{1}{2\eta_T}\int_0^{2\eta_T T} d\chi'\; f_{2b}(\chi')\,. \tag{51}$$

The integral evaluates via the antiderivative $df_b/d\chi = -f_{2b}$ together with $f_b(0) = \zeta(b)$ (large N),

$$\int_0^{2\eta_T T} d\chi'\; f_{2b}(\chi') = \zeta(b) - f_b(2\eta_T T) \;\xrightarrow[2\eta_T T\to\infty]{}\; \zeta(b)\,, \tag{52}$$

since $f_b(2\eta_T T) \sim (2\eta_T T)^{(1-b)/b} \to 0$ for $b > 1$. The bias term satisfies $f_a(2\eta_T T) \approx \Gamma((a-1)/b)(2\eta_T T)^{(1-a)/b}/b$. Combining,

$$L_T - \sigma^2 \approx \frac{\Gamma((a-1)/b)}{b}\,(2\eta_T T)^{-(a-1)/b} + \frac{\zeta(b)\,\sigma^2\,\eta_T}{2\,m}\,. \tag{53}$$

Dropping T-independent prefactors,

$$L_T - \sigma^2 \sim (\eta_T T)^{-\frac{a-1}{b}} + \frac{\sigma^2\eta_T}{m}\,. \tag{54}$$

Optimizing over η_T, we find $\eta_T \sim T^{-(a-1)/(a+b-1)}$ and $L_T - \sigma^2 \sim T^{-(a-1)/(a+b-1)}$.

C.2 Power-law schedules

In this section, we investigate the optimal power-law schedule $\eta(t) \sim t^{-\delta}$, with $0 < \delta < 1$. We start from the expression

$$L_T - \sigma^2 \sim \left[\int_0^T ds\;\eta(s)\right]^{-\alpha} + \frac{\sigma^2}{m}\int_0^T dt\;\eta(t)^2 f_{2b}(\chi(t)) = \text{Bias}(T) + \text{Variance}(T)\,, \tag{55}$$

where $\alpha = (a-1)/b$ and

$$\text{Bias}(T) = \left[\int_0^T ds\;\eta(s)\right]^{-(a-1)/b} \sim T^{-\alpha(1-\delta)}\,. \tag{56}$$

The variance integral can be decomposed in two contributions: one from the "bulk" of the integral and one from the region close to the upper limit T.

$$\text{Variance}(T) = \int_0^T dt\,\eta(t)^2 f_{2b}(\chi(t)) \approx \int_0^{T-\tau} dt\,\eta(t)^2 f_{2b}(\chi(t)) + \int_{T-\tau}^T dt\,\eta(t)^2 f_{2b}(\chi(t))\,, \tag{57}$$

where we choose τ such that $\chi(T-\tau) \sim \mathcal{O}(1)$, leading

$$\chi(T-\tau) \coloneqq \int_{T-\tau}^T \eta(t')dt' \sim T^{1-\delta} - (T-\tau)^{1-\delta} \approx \tau T^{-\delta}\,, \tag{58}$$

so that $\tau \sim T^\delta$. As a consequence, for $t < T - \tau$, we can approximate $f_{2b}(\chi(t)) \approx \chi(t)^{-\gamma} \sim T^{-\gamma(1-\delta)}$ with $\gamma = (2b-1)/b$, while for $t > T - \tau$, $f_{2b}(\chi(t))$ is approximately constant. Therefore,

$$\text{Variance}(T) \sim T^{-\gamma(1-\delta)}\int_0^{T-\tau} dt\; t^{-2\delta} + \int_{T-\tau}^T dt\; T^{-2\delta} = T^{-\gamma(1-\delta)}\frac{T^{1-2\delta}-1}{1-2\delta} + T^{-\delta}\,. \tag{59}$$

Therefore, δ must be chosen to minimize the combination of four terms

$$L_T - \sigma^2 \sim T^{-\alpha(1-\delta)} + T^{-\gamma(1-\delta)} + T^{-\gamma(1-\delta)+1-2\delta} + T^{-\delta}\,, \tag{60}$$

where we recall that $\alpha = (a-1)/b > 0$ and $1 < \gamma = 2 - 1/b < 2$. We find

$$\delta = \min\left[\frac{\alpha}{\alpha+1}, \frac{\gamma}{\gamma+1}\right] \tag{61}$$

with a crossover between the two regimes at $a = 2b$ and

$$L_T - \sigma^2 \sim T^{-\delta}\,. \tag{62}$$

Note that for the optimal schedule

$$L_T - \sigma^2 \sim T^{-\min((a-1)/a,(a-1)/b)} \tag{63}$$

so that there is an exponent gap between the optimal and the optimal power law schedules for all $a, b > 1$.

We next consider the scaling of power-law schedules with T-dependent prefactors of the form $\eta(t) \sim T^{-\nu}t^{-\delta}$. Following the same steps as before, we find

$$L_T - \sigma^2 \sim T^{\alpha\nu-\alpha(1-\delta)} + T^{-2\nu-\gamma(1-\delta-\nu)} + T^{-2\nu-\gamma(1-\delta-\nu)+1-2\delta} + T^{-\delta-\nu}\,. \tag{64}$$

Denoting $x = \delta + \nu$, we can rewrite

$$L_T - \sigma^2 \sim T^{-\alpha(1-x)} + T^{-(\gamma+(2-\gamma)x-2\delta)} + T^{-\gamma+1-(2-\gamma)x} + T^{-x}\,. \tag{65}$$

Therefore, we can choose $\delta = 0$, leading to the same scaling behavior as the constant-in-t optimal-in-T schedule.

C.3 General scaling form

In this section, we investigate the performance of scaling form schedules

$$\eta(t) = T^{-\xi} g(t/T)\,, \tag{66}$$

for $\xi \geq 0$. Note that in the easy phase $b < a$, the optimal schedule corresponds to $\xi = 1 - b/a$ and $g(z) \sim (1-z)^{2b-1}$. We find (for $m = 1$)

$$L_T - \sigma^2 \sim \left[\int_0^T dt\, \eta(t)\right]^{-\alpha} + \sigma^2 \int_0^T dt\, \eta^2(t) f_{2b}\left(\int_t^T dt'\, \eta(t')\right)\,, \tag{67}$$

where $\alpha = (a-1)/b$ and $f_{2b}(\chi) = \sum_{k=1}^N k^{-2b} e^{-k^{-b}\chi}$. Plugging the expression in Eq. (66) and changing variables $\tau = t/T$ and $\tau' = t'/T$, we find

$$L_T - \sigma^2 \approx \text{Bias}(T) + \text{Variance}(T)\,, \tag{68}$$

where

$$\text{Bias}(T) = T^{-\alpha(1-\xi)}\left[\int_0^1 d\tau\, g(\tau)\right]^{-\alpha} \tag{69}$$

and

$$\text{Variance}(T) = \sigma^2 T^{1-2\xi} \int_0^1 d\tau\, g^2(\tau)\, f_{2b}\left(T^{1-\xi}\int_\tau^1 d\tau'\, g(\tau')\right) \tag{70}$$

We recall the asymptotic behaviors $f_s(\chi) \approx \zeta(s) = \sum_{k\geq 1} k^{-s}$ for small χ (and large N) and $f_s(\chi) \approx \Gamma((s-1)/b)\chi^{(1-s)/b}/b$ for large χ. It is then instructive to split the variance integral into an edge component, where $\chi(t) = \int_t^T dt'\, \eta(t') \ll 1$ and a bulk component where $\chi(t) \gg 1$. To do so, we define t^* so that $\chi(t^*) = 1$, yielding the condition for $\tau^* := t^*/T$

$$\int_{\tau^*}^1 d\tau' g(\tau') = T^{\xi-1}\,. \tag{71}$$

Assuming $\xi < 1$ (since for $\xi > 1$ the bias increases with T), we get that τ^* is determined by the asymptotic behavior of $g(\tau)$ for $\tau \to 1$. Assuming $g(\tau) \sim (1-\tau)^\delta$ with $\delta > 0$, we get

$$(1-\tau^*) \sim T^{-(1-\xi)/(1+\delta)} \,. \tag{72}$$

We can then split the variance term into a "bulk" component for $\tau < \tau^*$ and an "edge" component for $\tau > \tau^*$, leading to

$$\text{Variance}(T) \approx \text{Bulk Variance}(T) + \text{Edge Variance}(T) \,. \tag{73}$$

The bulk contribution reads

$$\text{Bulk Variance}(T) \sim T^{1-2\xi} \int_0^{\tau^*} d\tau \; g^2(\tau) \; \left(T^{1-\xi} \int_\tau^1 d\tau' \; g(\tau') \right)^{-\gamma} \,. \tag{74}$$

For $\delta > (\gamma-1)/(2-\gamma) = b-1$ the integral over τ converges as $\tau^* \to 1$ and hence

$$\text{Bulk Variance}(T) \sim T^{1-2\xi-\gamma(1-\xi)} \,. \tag{75}$$

However, if $\delta < b-1$ there is an additional contribution from the divergence as $\tau^* \to 1$ for large T, leading to

$$\text{Bulk Variance}(T) \sim T^{-(\delta+\xi)/(\delta+1)} \,. \tag{76}$$

The edge contribution is

$$\text{Edge Variance}(T) \sim T^{1-2\xi} \int_{\tau^*}^1 d\tau \; g^2(\tau) \sim T^{1-2\xi} \int_{\tau^*}^1 d\tau \; (1-\tau)^{2\delta} \sim T^{1-2\xi} (1-\tau^*)^{2\delta+1} \tag{77}$$

$$\sim T^{1-2\xi-(1-\xi)(1+2\delta)/(1+\delta)} \sim T^{-(\delta+\xi)/(\delta+1)} \,. \tag{78}$$

Overall, for $\delta > b-1$, we get

$$L_T - \sigma^2 \sim T^{-\alpha(1-\xi)} + T^{1-2\xi-\gamma(1-\xi)} + T^{-(\delta+\xi)/(\delta+1)} \,, \tag{79}$$

while for $\delta < b-1$ we find

$$L_T - \sigma^2 \sim T^{-\alpha(1-\xi)} + T^{-(\delta+\xi)/(\delta+1)} \,, \tag{80}$$

Interestingly, for large enough δ, the edge variance terms is negligible and the exponent ξ is obtained by matching the bias and bulk variance terms. In particular, we find $\xi = (a-b)/a$ for $b < a$ (easy phase) and $\xi = 0$ for $b > a$ (hard phase). The corresponding scaling laws are $L_T - \sigma^2 \sim T^{-(a-1)/a}$ for $b < a$ and $L_T - \sigma^2 \sim T^{-(a-1)/b}$ for $b > a$. The scalings match those of the optimal schedule.

C.3.1 The edge term is subleading for the optimal schedule

In the easy phase $b < a$, the optimal schedule takes the form in Eq. (66) with $\xi = (a-b)/a$ and $\delta = 2b-1$. This value of δ is chosen to minimize the prefactor of the scaling law. For this value of δ, the edge variance term, which was not included in the derivation of the optimal schedule, is indeed subleading.

We next consider the edge variance term of the optimal schedule in the hard phase $b > a$. Recall that in this phase the optimal learning rate schedule is (taking $\eta_{\max} = 1$)

$$\eta(t) = \begin{cases} 1 & \text{for } t < t_s \,, \\ \left(\frac{1-t/T}{1-t_s/T} \right)^\delta & \text{for } t > t_s \,, \end{cases} \tag{81}$$

with $\delta = 2b-1$ and $T - t_s \sim T^{1-(b-a)/(2b-1)}$. The edge boundary t^* is determined by the condition $\chi(t^*) = 1$, yielding

$$\chi(t^*) = \int_{t^*}^T dt' \eta(t') = \int_{t^*}^T dt' \left(\frac{1-t'/T}{1-t_s/T} \right)^\delta = T\epsilon^{-\delta}(1-\tau^*)^{\delta+1} = 1 \,, \tag{82}$$

where we have assumed $t_s < t^*$ and we have defined $\epsilon \coloneqq 1 - t_s/T \sim T^{-(b-a)/(2b-1)}$ and $\tau^* = t^*/T$. Therefore, we get

$$1 - \tau^* = \left(\frac{\epsilon^\delta}{T} \right)^{1/(\delta+1)} \sim T^{-(b-a+1)/(2b)} \,. \tag{83}$$

Note that $1 - t_s/T > 1 - \tau^*$, confirming that $t_s < t^*$.

The edge variance term reads

$$\text{Edge Variance}(T) = \int_{t^*}^{T} dt\, \eta^2(t) = \int_{t^*}^{T} dt\, \eta^2(t) = \int_{t^*}^{T} dt\, \left(\frac{1-t/T}{1-t_s/T}\right)^{2\delta} = (\epsilon T)^{-\delta/(\delta+1)}\,. \tag{84}$$

The bulk variance term scales as $(\epsilon T)^{-1+1/b}$, confirming that the edge term is subleading.

D Joint optimization of learning rate and batch size

We want to find the joint optimal schedule of learning rate and batch size that minimizes the final loss. We consider a constraint of the total number of training examples $B_{\text{tot}} = \int_0^T dt\, m(t)$.

Using Eq. (26), the loss with the data-budget constraint reads

$$L_T - \sigma^2 \approx f_a(2\chi_0) + \sigma^2 \int_0^T dt\, \frac{\eta^2(t)}{m(t)}\, f_{2b}(2\chi(t)) + \mu \int_0^T dt\, m(t)\,, \tag{85}$$

where the Lagrange multiplier μ enforces the constraint on $m(t)$. Optimizing over $m(t)$ yields

$$m(t) = \eta(t)\,\sigma \sqrt{\frac{f_{2b}(2\chi(t))}{\mu}}\,, \tag{86}$$

and the constraint $B_{\text{tot}} = \int_0^T dt\, m(t)$ then fixes

$$\sqrt{\mu} = \frac{\sigma}{B_{\text{tot}}} \int_0^{\chi_0} d\chi\, \sqrt{f_{2b}(2\chi)}\,, \tag{87}$$

where we changed integration variable using $\eta(t)\, dt = -d\chi$. Plugging back into the loss,

$$L_T - \sigma^2 \approx f_a(2\chi_0) + \frac{\sigma^2}{B_{\text{tot}}} \left[\int_0^{\chi_0} d\chi\, \sqrt{f_{2b}(2\chi)}\right]^2\,. \tag{88}$$

Note that this expression depends on χ_0 only, so at fixed χ_0 all learning-rate schedules give the same performance provided $m(t)$ is chosen accordingly.

For large χ_0, we find

$$L_T - \sigma^2 \sim \chi_0^{-(a-1)/b} + \chi_0^{1/b}/B_{\text{tot}}\,, \tag{89}$$

implying that at optimality $\chi_0 \sim B_{\text{tot}}^{b/a}$. This scaling can be achieved in the easy phase ($b < a$), leading to $L_T - \sigma^2 \sim B_{\text{tot}}^{-(a-1)/a}$. Note however that χ_0 cannot scale superlinearly with B_{tot}. Indeed, since $\eta(t) \leq \eta_{\max}$ and $m(t) \geq 1$, we have the conditions $\chi_0 \leq \eta_{\max} T$ and $B_{\text{tot}} \geq T$, yielding $\chi_0 \leq \eta_{\max} B_{\text{tot}}$. Therefore, in the hard phase, the optimum is achieved for $\chi_0 \sim B_{\text{tot}} \sim T$, for which $L_T \sim B_{\text{tot}}^{-(a-1)/b}$.

In both easy and hard phases, the degeneracy in the optimization over η and m can be broken by requiring the solution to have the minimum wall-clock time T (see Fig. 8). This corresponds to taking $\eta(t) = \eta_{\max}$ and

$$m(t) = \frac{B_{\text{tot}}}{2bT}(1 - t/T)^{1/(2b)-1}\,. \tag{90}$$

In the hard phase, $B_{\text{tot}} \sim T$, so that $m(t)$ remains bounded. However, in the easy phase $T \sim B_{\text{tot}}^{b/a}$ so that the batch size actually grows with B_{tot} as $m(t) \sim B_{\text{tot}}^{1-b/a}$.

E Loss Dynamics for SGD + Momentum

E.1 SGD Derivation

In this section, we compute the average case loss dynamics over random draws of data. This follows the standard arguments in Bordelon and Pehlevan [2022]. Starting from the following dynamics for

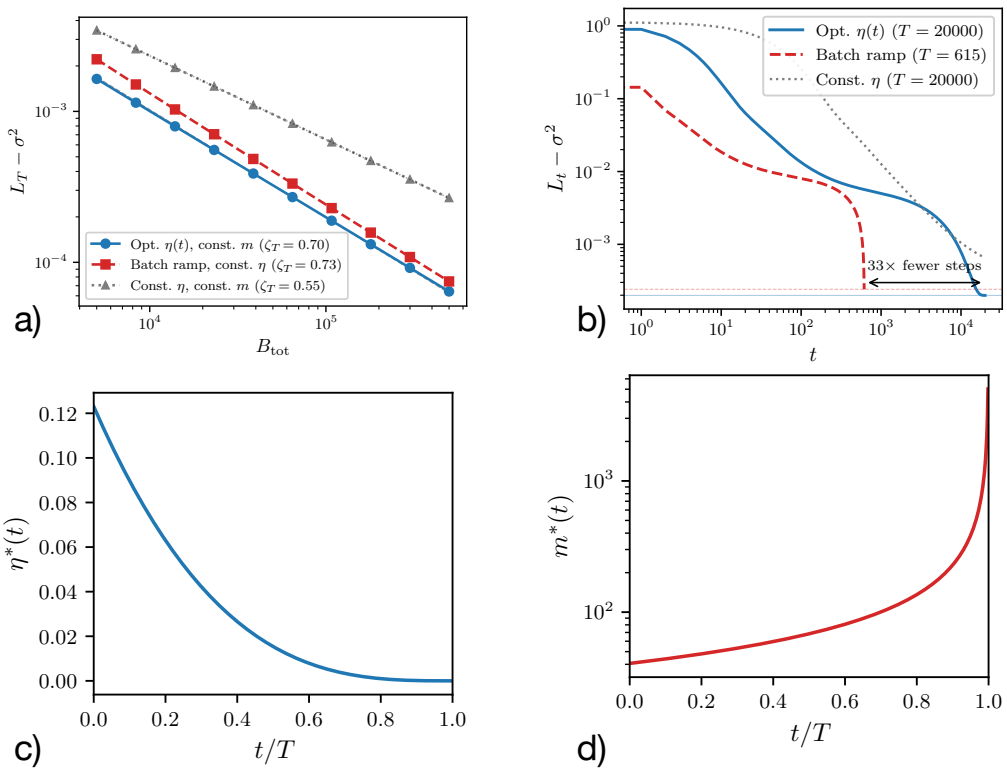


Figure 8: **Optimal batch schedule in the easy phase** ($b < a$)**.** We simulate the deterministic loss dynamics (Eq. (5)) of the powerlaw random feature model and compare three strategies at fixed total data budget $B_{\rm tot}$: the optimal learning rate schedule (panel c) with constant batch size m, the batch ramp (panel d) with constant $\eta = \eta_{\max}$, and a constant learning rate baseline (with η chosen optimally at each $B_{\rm tot}$). **a)** Final excess loss $L_T - \sigma^2$ as a function of $B_{\rm tot}$. Both the optimal schedule and the batch ramp achieve the same scaling exponent $\zeta_T \approx (a-1)/a \approx 0.71$, outperforming the constant baseline ($\zeta_T \approx (a-1)/(a+b-1) \approx 0.56$). **b)** Loss dynamics for $B_{\rm tot} = 10^5$. The batch ramp reaches the same loss scaling in $33\times$ fewer wall-clock steps, confirming that the joint optimization reduces wall-clock time while preserving the optimal scaling with respect to $B_{\rm tot}$. **Parameters:** $N = 1000$, $a = 3.5$, $b = 2$, $\sigma = 0.5$, $m = 5$, $\eta_{\max} = 1$.

$\boldsymbol{\Delta}_t = (\boldsymbol{w}^\star - \boldsymbol{w}_t)$. This variable has dynamics

$$\boldsymbol{\Delta}_{t+1} = \boldsymbol{\Delta}_t - \eta_t \boldsymbol{g}_t \ , \ \boldsymbol{g}_t = \frac{1}{m_t} \sum_{\mu=1}^{m_t} \tilde{\boldsymbol{\psi}}_{\mu,t} [\tilde{\boldsymbol{\psi}}_{\mu,t} \cdot \boldsymbol{\Delta}_t + \sigma \epsilon_{\mu,t}] \tag{91}$$

The correlation matrix of interest is $\boldsymbol{C}_t = \left\langle \boldsymbol{\Delta}_t \boldsymbol{\Delta}_t^\top \right\rangle_{\mathcal{D}_{t-1}}$. The loss is $L_t = {\rm Tr} \boldsymbol{\Lambda} \boldsymbol{C}_t + \sigma^2$. This matrix satisfies the exact recursion

$$\boldsymbol{C}_{t+1} = \boldsymbol{C}_t - \eta_t \boldsymbol{\Lambda} \boldsymbol{C}_t - \eta_t \boldsymbol{C}_t \boldsymbol{\Lambda} + \eta_t^2 \left\langle \boldsymbol{g}_t \boldsymbol{g}_t^\top \right\rangle . \tag{92}$$

For Gaussian features, the gradient correlations have the following form

$$\left\langle \boldsymbol{g}_t \boldsymbol{g}_t^\top \right\rangle = \frac{m_t + 1}{m_t} \boldsymbol{\Lambda} \boldsymbol{C}_t \boldsymbol{\Lambda} + \frac{1}{m_t} \boldsymbol{\Lambda} \left[{\rm Tr} \boldsymbol{\Lambda} \boldsymbol{C}_t + \sigma^2 \right] . \tag{93}$$

Since the loss only depends on the diagonal entries of $\boldsymbol{C}_t$ which we define as $c_{t,k} \coloneqq C_{t,kk}$.

$$c_{t+1,k} = \left(1 - 2\eta_t \lambda_k + \eta_t^2 \frac{m_t + 1}{m_t} \lambda_k^2 \right) c_{t,k} + \frac{\eta_t^2}{m_t} \lambda_k \sum_\ell \lambda_\ell c_{t,\ell} + \frac{\sigma^2 \eta_t^2}{m_t} \lambda_k . \tag{94}$$

Under upper and lower bounds on the fourth moments of the features, one can derive analogous upper and lower bounds for the loss dynamics and SGD noise Bordelon and Pehlevan [2022].

E.2 Momentum

To analyze momentum, we must also track auxiliary momentum variable $\boldsymbol{v}(t)$, we track the following updates

$$\boldsymbol{\Delta}_{t+1} = \boldsymbol{\Delta}_t - \eta_t \boldsymbol{v}_t \ , \ \boldsymbol{v}_t = (1-\beta_t)\boldsymbol{v}_{t-1} + \beta_t \boldsymbol{g}_t \quad , \quad \boldsymbol{g}_t = \frac{1}{m}\sum_{\mu=1}^{m} \tilde{\boldsymbol{\psi}}_{\mu,t}[\tilde{\boldsymbol{\psi}}_{\mu,t}\cdot\boldsymbol{\Delta}_t + \sigma\epsilon_{\mu,t}] \quad (95)$$

To compute the test loss, we need to track the following three correlation functions

$$C_{k,t} = \left\langle \Delta_{k,t}^2 \right\rangle \ , \ V_{k,t} = \left\langle v_{k,t}^2 \right\rangle \ , \ R_{k,t} = \langle \Delta_{k,t} v_{k,t} \rangle \quad (96)$$

where the average is computed over the random samples from each minibatch. These three variables evolve according to the following linear dynamics

$$\begin{aligned}
C_{k,t+1} &= C_{k,t} - 2\eta_t R_{k,t} + \eta_t^2 V_{k,t} \, , \\
V_{k,t+1} &= (1-\beta_t)^2 V_{k,t} + 2\beta_t(1-\beta_t)\lambda_k(R_{k,t} - \eta_t V_{k,t}) \\
&+ \beta_t^2 \left[\frac{m+1}{m}\lambda_k^2 C_{k,t+1} + \frac{1}{m}\lambda_k \sum_\ell \lambda_\ell C_{\ell,t+1} + \frac{\sigma^2}{m}\lambda_k \right] \\
R_{k,t+1} &= (1-\beta_t)(R_{k,t} - \eta_t V_{k,t}) + \beta_t \lambda_k C_{k,t+1}
\end{aligned} \quad (97)$$

We note that due to the time indexing that we adopted above, the right hand side for the equations defining the dynamics for $R_{k,t+1}$ and $V_{k,t+1}$ depend on $C_{k,t+1}$ rather than $C_{k,t}$ (though of course this can be substituted).[4] The loss function dynamics, as before, can be computed from $C_{k,t}$

$$L_t - \sigma^2 = \sum_k \lambda_k C_{k,t}. \quad (98)$$

We use the above dynamics for $\{C_{k,t}, R_{k,t}, V_{k,t}\}$.

F Proof of Theorem 2

We use the notation $q \coloneqq (m+1)/m$, $p_k \coloneqq \lambda_k (w_k^\star)^2 = k^{-a}$, $\alpha \coloneqq (a-1)/b$, $\gamma \coloneqq (2b-1)/b$, the discrete integrated learning rate $\chi_t \coloneqq \sum_{j=t}^{T-1} \eta_j$ (so that $\chi_T = 0$ and $\chi_t = \chi(t)$ at integer times), and the spectral sums

$$f_{s,N}(x) \coloneqq \sum_{k=1}^{N} k^{-s} e^{-x k^{-b}} \, . \quad (99)$$

The condition $\eta_{\max} < 2m/(m+1+\zeta(b))$ is equivalent to $\kappa \coloneqq \eta_{\max}\zeta(b)/\left(m(2-q\eta_{\max})\right) < 1$. Since $\sigma^2 = \sigma_0^2 + \sum_{k>N} k^{-a}$, we have $\sigma_0^2 \le \sigma^2 \le \sigma_0^2 + \zeta(a)$. All constants below depend only on $(a, b, m, \sigma_0, \eta_{\max})$.

Step 1: exact per-mode recursion. By Theorem 1, using $\sum_\ell \lambda_\ell c_{t,\ell} + \sigma^2 = L_t$,

$$c_{t+1,k} = r_t(k)\, c_{t,k} + \frac{\eta_t^2}{m}\lambda_k L_t \, , \qquad r_t(k) \coloneqq 1 - 2\eta_t\lambda_k + q\,\eta_t^2\lambda_k^2 \, . \quad (100)$$

Step 2: exact solution. Let $P_{s,t}(k) \coloneqq \prod_{j=s}^{t-1} r_j(k)$, with $P_{t,t}(k) = 1$. Iterating the recursion and using $\lambda_k c_{0,k} = p_k$, which follows from $\boldsymbol{w}_0 = \boldsymbol{0}$,

$$L_t - \sigma^2 = \sum_{k=1}^{N} p_k \, P_{0,t}(k) + \frac{1}{m}\sum_{s=0}^{t-1} \eta_s^2 \, L_s \sum_{k=1}^{N} \lambda_k^2 \, P_{s+1,t}(k) \, . \quad (101)$$

[4] We find this convention yields better stability than defining the dynamics as $\boldsymbol{\Delta}_{t+1} = \boldsymbol{\Delta}_t - \eta_t \boldsymbol{v}_t$ where $\boldsymbol{v}_t = (1-\beta_t)\boldsymbol{v}_{t-1} + \beta_t \boldsymbol{g}_{t-1}$.

Step 3: products versus exponentials. Let $r(x) \coloneqq 1 - 2x + qx^2$ for $0 \le x \le \eta_{\max}$, and define $c_- \coloneqq 2 - q\eta_{\max} > 0$ and $c_+ \coloneqq \sup_{0<x\le\eta_{\max}}[-\log r(x)]/x$. Since $r(x) - (1 - c_- x) = qx(x - \eta_{\max}) \le 0$, we have $r(x) \le 1 - c_- x \le e^{-c_- x}$. Since $r(x) \ge \min_y r(y) = 1/(m+1)$ and $1 - r(x) = x(2 - qx) \le 2x$, we have $-\log r(x) \le (1 - r(x))/r(x) \le 2(m+1)x$, so $c_+ \le 2(m+1)$ and $r(x) \ge e^{-c_+ x}$. Because $\lambda_k \le 1$, we can apply these bounds with $x = \eta_j \lambda_k$ and obtain

$$e^{-c_+ \lambda_k \sum_{j=s}^{t-1} \eta_j} \le P_{s,t}(k) \le e^{-c_- \lambda_k \sum_{j=s}^{t-1} \eta_j} \le 1\,. \tag{102}$$

Step 4: uniform control of the loss. We show $\sigma^2 \le L_t \le L_{\max} \coloneqq (\sigma_0^2 + 2\zeta(a))/(1 - \kappa)$ for all t. The lower bound holds because $c_{t,k} \ge 0$. For the upper bound, fix k and t, and let $Q_s \coloneqq \prod_{j=s}^{t-1}(1 - c_- \eta_j \lambda_k) \in [0, 1]$. Then $Q_{s+1} - Q_s = c_- \eta_s \lambda_k Q_{s+1}$ and, since $0 < r_j(k) \le 1 - c_- \eta_j \lambda_k$ for each j (Step 3), $P_{s+1,t}(k) \le Q_{s+1}$, so that

$$\sum_{s=0}^{t-1} \eta_s^2 P_{s+1,t}(k) \le \eta_{\max} \sum_{s=0}^{t-1} \eta_s Q_{s+1} = \frac{\eta_{\max}(1 - Q_0)}{c_- \lambda_k} \le \frac{\eta_{\max}}{c_- \lambda_k}\,. \tag{103}$$

If $L_s \le L_{\max}$ for all $s < t$, Eq. (101) together with $\sum_k p_k \le \zeta(a)$ and $\sum_k \lambda_k \le \zeta(b)$ gives $L_t \le \sigma_0^2 + 2\zeta(a) + \kappa L_{\max} = L_{\max}$, where we recall that $\kappa = \eta_{\max}\zeta(b)/(m\, c_-)$. Since $L_0 = \sigma_0^2 + \sum_{k \le M} k^{-a} \le \sigma_0^2 + \zeta(a) \le L_{\max}$, the claim follows by induction.

Step 5: sums over modes. Inserting the bounds of Eq. (102) into Eq. (101) at $t = T$, and noting that $\sum_k p_k e^{-c\lambda_k \chi} = f_{a,N}(c\chi)$ and $\sum_k \lambda_k^2 e^{-c\lambda_k \chi} = f_{2b,N}(c\chi)$, we obtain

$$f_{a,N}(c_+\chi_0) + \tfrac{1}{m} \sum_{t=0}^{T-1} \eta_t^2 L_t f_{2b,N}(c_+\chi_{t+1}) \le L_T - \sigma^2 \le f_{a,N}(c_-\chi_0) + \tfrac{1}{m} \sum_{t=0}^{T-1} \eta_t^2 L_t f_{2b,N}(c_-\chi_{t+1})\,. \tag{104}$$

Bounding L_t with Step 4, $\sigma^2 \le L_t \le L_{\max}$, then gives

$$f_{a,N}(c_+\chi_0) + \frac{\sigma^2}{m} \sum_{t=0}^{T-1} \eta_t^2 f_{2b,N}(c_+\chi_{t+1}) \le L_T - \sigma^2 \le f_{a,N}(c_-\chi_0) + \frac{L_{\max}}{m} \sum_{t=0}^{T-1} \eta_t^2 f_{2b,N}(c_-\chi_{t+1})\,. \tag{105}$$

We now show that, for $s > 1$, $y \ge 0$ and $c > 0$,

$$\begin{aligned} &f_{s,N}(cy) \le A_{s,c}\,(1 + y)^{-\frac{s-1}{b}} \quad \text{for every } N, \\ &f_{s,N}(cy) \ge e^{-c}\, 2^{1-2s}\,(1 + y)^{-\frac{s-1}{b}} \quad \text{if } N \ge 4(1 + y)^{1/b}, \end{aligned} \tag{106}$$

with $A_{s,c}$ independent of N and y.

Upper bound. Since $f_{s,N} \le f_{s,\infty}$, it suffices to bound the infinite sum. Let $x \coloneqq cy$. If $x < 1$, we simply use $f_{s,\infty}(x) \le \zeta(s)$. If $x \ge 1$, we split the sum at $K \coloneqq x^{1/b}$, the mode for which $xk^{-b} = 1$. For $k \le K$, the exponential is small, and $e^{-u} \le (p/e)^p u^{-p}$ with $u = xk^{-b}$ and $p = s/b$ gives $k^{-s} e^{-xk^{-b}} \le (s/(be))^{s/b} x^{-s/b}$ for each of the at most K terms. For $k > K$, we drop the exponential. Altogether,

$$f_{s,\infty}(x) \le K \left(\frac{s}{be}\right)^{s/b} x^{-s/b} + \sum_{k>K} k^{-s} \le \left[\left(\frac{s}{be}\right)^{s/b} + \frac{2^{s-1}}{s-1}\right] x^{-\frac{s-1}{b}}\,. \tag{107}$$

Since $x \ge (1+x)/2$ for $x \ge 1$, and $1 + x < 2$ for $x < 1$, both cases give $f_{s,\infty}(x) \le C\,(1+x)^{-(s-1)/b}$ with C independent of x. Finally, $1 + x = 1 + cy \ge \min(1, c)(1 + y)$ gives the upper bound in Eq. (106).

Lower bound. Since all terms are positive, we bound $f_{s,N}(cy)$ from below by keeping only the modes $K \le k \le 2K$, where $K \coloneqq \lceil (1 + y)^{1/b} \rceil$ is the smallest integer with $K^b \ge 1 + y$. These are the modes for which the exponential is of order one. They are retained by the cutoff because $K \le (1 + y)^{1/b} + 1 \le 2(1 + y)^{1/b}$, so that $2K \le 4(1 + y)^{1/b} \le N$. For each of them, $k^b \ge 1 + y$ gives $cyk^{-b} \le cy/(1 + y) \le c$, and $k \le 2K$ gives $k^{-s} \ge (2K)^{-s}$. Since there are $K + 1 \ge K$ such modes,

$$f_{s,N}(cy) \ge K\,(2K)^{-s} e^{-c} = 2^{-s} K^{1-s} e^{-c} \ge e^{-c}\, 2^{1-2s} (1 + y)^{-\frac{s-1}{b}}\,, \tag{108}$$

where the last step uses $K \le 2(1+y)^{1/b}$ and $s > 1$.

Conclusion. We apply Eq. (106) to both sides of Eq. (105), with $s = a$ and $y = \chi_0$ for the bias, and $s = 2b$ and $y = \chi_{t+1}$ for the variance. Since $\chi_{t+1} \le \chi_0$, the single condition $N \ge 4(1+\chi_0)^{1/b}$ covers all terms. Using also $L_{\max}/\sigma^2 \le L_{\max}/\sigma_0^2$, we obtain

$$L_T - \sigma^2 \asymp (1+\chi_0)^{-\alpha} + \frac{\sigma^2}{m} \sum_{t=0}^{T-1} \eta_t^2 (1+\chi_{t+1})^{-\gamma}\,, \tag{109}$$

where the upper bound holds for every N and the lower bound for $N \ge 4(1+\chi_0)^{1/b}$, and $\asymp$ denotes equality up to constants that depend only on $(a, b, m, \sigma_0, \eta_{\max})$.

Step 6: sums versus integrals. For $t \in [j, j+1)$, $\chi(t) = \chi_{j+1} + (j+1-t)\eta_j \in [\chi_{j+1}, \chi_{j+1} + \eta_{\max}]$, hence $1+\chi_{j+1} \le 1+\chi(t) \le (1+\eta_{\max})(1+\chi_{j+1})$. Integrating over each interval $[j, j+1)$ gives

$$(1+\eta_{\max})^{-\gamma} \sum_{t=0}^{T-1} \eta_t^2 (1+\chi_{t+1})^{-\gamma} \le \int_0^T dt\, \eta(t)^2 (1+\chi(t))^{-\gamma} \le \sum_{t=0}^{T-1} \eta_t^2 (1+\chi_{t+1})^{-\gamma}\,. \tag{110}$$

Together with $\chi_0 = \chi(0)$ and Eq. (109), this proves Theorem 2. □

G Optimal Control Implementation

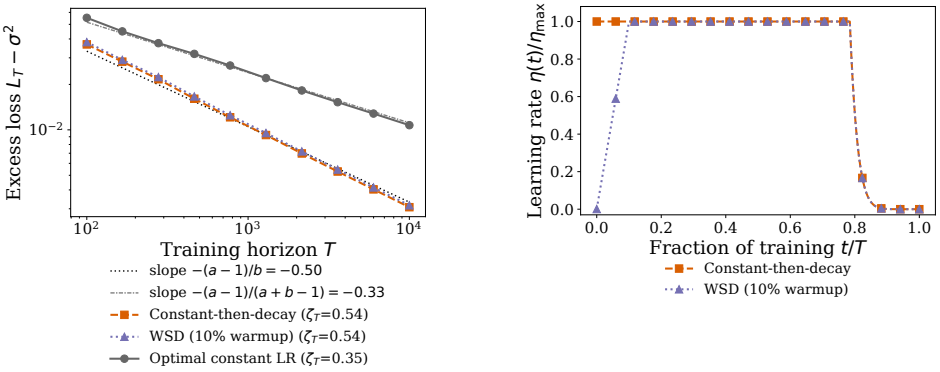
To determine the optimal learning rate schedules numerically, we formulate the problem as a discrete-time optimal control problem governed by the recursive evolution of $c_{t,k}$ (Eq. (5)). We employ a direct single-shooting method to transcribe this control problem into a non-linear programming (NLP) problem. In this formulation, the control inputs (the learning rate η_t and optionally the batch size m_t) are treated as the primary decision variables. The state trajectory is not included in the set of decision variables. The objective function, defined as the terminal generalization error L_T, is thus expressed as a direct function of the initial conditions and the sequence of control inputs.

We implement this optimization framework using CasADi Andersson et al. [2019], which performs automatic differentiation to efficiently compute the necessary gradients of the objective with respect to the control trajectory. The resulting NLP is solved using IPOPT (Interior Point OPTimizer), a primal-dual interior-point method. To ensure computational efficiency for large horizons, we configure the solver to use a limited-memory approximation of the Hessian and the MUMPS linear solver for the internal step computations. The implementation of our optimization framework is available in the supplementary material.

H Additional Experiments in the Random Feature Model

Warmup-stable-decay. Our model predicts a constant-then-decay schedule without warmup in the hard phase. To test whether a warmup changes the scaling, we iterate the exact recursion of Theorem 1 for (i) the constant-then-decay schedule of Result 2 with annealing fraction $1 - t_s/T = T^{-(b-a)/(2b-1)}$, (ii) the same schedule with a linear warmup over the first 10% of training, i.e., a full WSD schedule, and (iii) the optimal constant learning rate. Figure 9 shows that the warmup increases the final loss by about 3% but leaves the scaling exponent unchanged. Fits over $T \ge 10^3$ give $\zeta_T \approx 0.54$ for both (i) and (ii), and $\zeta_T \approx 0.35$ for (iii), close to the predictions $(a-1)/b = 0.5$ and $(a-1)/(a+b-1) = 1/3$ (the small offset is a finite-T effect). This is consistent with Theorem 2: a warmup over a fixed fraction of training reduces $\chi(0)$ by at most a constant factor, and its contribution to the variance is subleading.

Gaussian sketch. We replace the top-N projection by a Gaussian sketch $\boldsymbol{G}$ with i.i.d. entries drawn from $\mathcal{N}(0, 1/N)$, and simulate online SGD directly, averaging over 12 seeds and evaluating the population loss in closed form. We apply the same analytic schedules as for $\boldsymbol{G} = [\boldsymbol{I}_N, \boldsymbol{0}]$: the constant-then-decay schedule in the hard phase, and $\eta(t) = \eta_0(T)(1-t/T)^{2b-1}$ with $\eta_0(T) = (T/100)^{b/a-1}$ in the easy phase. In the hard phase, the fitted exponents are $\zeta_T \approx 0.53$ for the sketch and 0.55 for the top-N projection, compared with the prediction $(a-1)/b = 0.5$ (Figure 10b). In the easy phase,

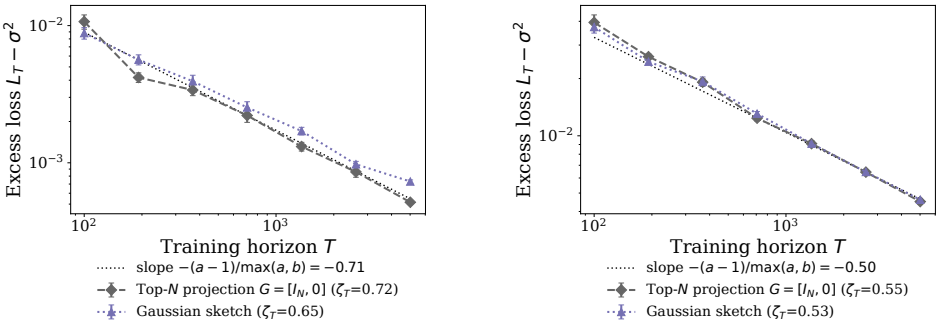


(a) Final loss

(b) Schedules at $T = 10^4$

Figure 9: **A warmup does not change the hard-phase scaling exponent.** (a) Final excess loss as a function of the horizon T for the constant-then-decay schedule, the same schedule with a 10% linear warmup (WSD), and the optimal constant learning rate, computed with the exact recursion of Theorem 1. Dotted lines indicate the predicted slopes. (b) The two schedules at $T = 10^4$. **Parameters:** $a = 3.5$, $b = 5$, $\sigma = 0.5$, $m = 5$, $N = 2000$, $\eta_{\max} = 1$.

we find $\zeta_T \approx 0.65$ for the sketch and 0.72 for the top-N projection, compared with $(a-1)/a \approx 0.71$ (Figure 10a). The sketch thus approximately preserves the exponents, with a deviation in the easy phase that we attribute to finite N.



(a) Easy phase ($b < a$)

(b) Hard phase ($b > a$)

Figure 10: **A Gaussian sketch approximately preserves the loss-scaling exponents.** Final excess loss of online SGD as a function of the horizon T under the analytic optimal schedules, for the top-N projection and for a Gaussian sketch $\boldsymbol{G}$. Error bars are standard errors over 12 seeds. Dotted lines indicate the predicted slope $(a-1)/\max(a,b)$. **Parameters:** $a = 3.5$, $b = 2$ (a) and $b = 5$ (b), $\sigma_0 = 0.5$, $m = 5$, $N = 500$, $M = 2000$, $\eta_{\max} = 1$.

I Experiment and Compute Details

ResNets on CIFAR-5M We train deep residual convolutional networks on CIFAR-5M with SGD and different schedules. We contrast different WSD like schedules predicted by our hard phase optimal annealing strategies in Figure 11. Experiments were conducted on a Nvidia A100 GPU and took about 10-15 minutes per run.

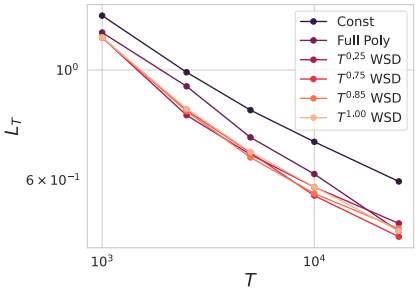


Figure 11: ResNet CIFAR-5M training contrasting annealing for T^δ steps instead of $O(T)$ steps. As predicted by our theory in the hard phase, these schedules outperform fully polynomial annealing.

Language Model The models are based on the OLMo repository models and training scripts OLMo et al. [2024]. The base model is $D_{\text{model}} = 768$ with $H = 12$ heads and $L = 12$ blocks attention and MLP blocks. The C4 data are tokenized with the GPT-2 tokenizer whose vocabulary is ≈ 50k. Weight decay is 0.1 and either a constant or cosine annealing schedule is adopted. We used 3x Nvidia A100 per run for varying token horizon T. The runtime ranges between 1-10 hours depending on T.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All claims are supported by theoretical derivations and empirical simulations.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have a limitations section in the discussion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Theorems 1 and 2 state their full assumptions in the main text and are proved in Appendices E and F. Results 1–3 are derived at leading order in large T under the assumptions listed in Appendix A, and are marked as such in Section 3.2.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide code to reproduce the experiments of the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Yes, we provide a .zip file with the code needed to reproduce the figures.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide details on all of our experiments on both the powerlaw random feature model and the ResNet computer vision and GPT-2 language model experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification: The loss that we optimize with optimal control is averaged analytically over the random draws of SGD iterates and is thus non-fluctuating.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, we specify the compute resources in the Appendix. For the optimal control of the analytical equations, we use a laptop or run a Google Colab notebook. The ResNet and GPT experiments are conducted with A100 GPUs (one per run for the ResNet and 4 per run for the GPT).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We conform.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have a broader impacts section in Discussion.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We are not releasing any large models or datasets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have citations to the main resources.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new assets introduced in this paper.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We don't crowdsource or use human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No participants.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: We used AI assistants for minor editing and to produce a first version of the proof of Theorem 2, which the authors checked. This is stated in the section "Use of AI Tools" after the Acknowledgements.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.