# A MULTI-POWER LAW FOR LOSS CURVE PREDICTION ACROSS LEARNING RATE SCHEDULES

**Kairong Luo**[1] **Haodong Wen**[2] **Shengding Hu**[1] **Zhenbo Sun**[1]
**Zhiyuan Liu**[1] **Maosong Sun**[1] † **Kaifeng Lyu**[3] † **Wenguang Chen**[1,4] †
[1]Department of Computer Science and Technology, Tsinghua University
[2]Qian Xuesen College, Xi'an Jiaotong University
[3]Simons Institute, University of California, Berkeley
[4]Peng Cheng Laboratory
{luokr24,sunzb20}@mails.tsinghua.edu.cn
{herrywenh,shengdinghu}@gmail.com
kaifenglyu@berkeley.edu
{liuzy,sms,cwg}@tsinghua.edu.cn

## ABSTRACT

Training large models is both resource-intensive and time-consuming, making it crucial to understand the quantitative relationship between model performance and hyperparameters. In this paper, we present an empirical law that describes how the pretraining loss of large language models evolves under different learning rate schedules, such as constant, cosine, and step decay schedules. Our proposed law takes a multi-power form, combining a power law based on the sum of learning rates and additional power laws to account for a loss reduction effect induced by learning rate decay. We extensively validate this law on various model sizes and architectures, and demonstrate that after fitting on a few learning rate schedules, the law accurately predicts the loss curves for unseen schedules of different shapes and horizons. Moreover, by minimizing the predicted final pretraining loss across learning rate schedules, we are able to find a schedule that outperforms the widely used cosine learning rate schedule. Interestingly, this automatically discovered schedule bears some resemblance to the recently proposed Warmup-Stable-Decay (WSD) schedule (Hu et al., 2024) but achieves a slightly lower final loss. We believe these results could offer valuable insights for understanding the dynamics of pretraining and designing learning rate schedules to improve efficiency.

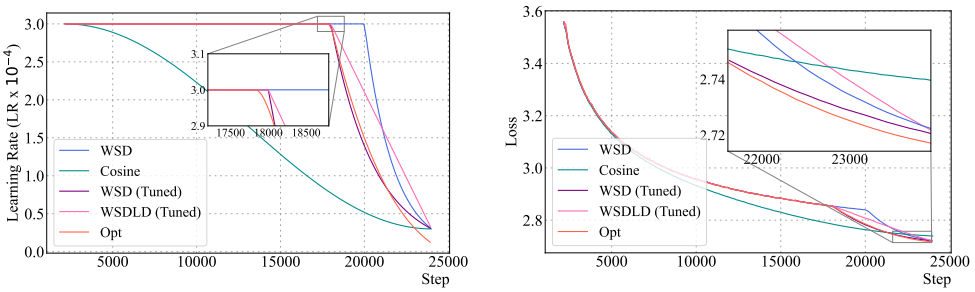
## 1 INTRODUCTION

Large Language Models (LLMs) can achieve strong performance if pretrained with an appropriate configuration of hyperparameters, such as model width, depth, number of training steps, and learning rate. However, tuning these hyperparameters at scale is extremely costly since one pretraining run can take weeks or even months.

To reduce the cost of hyperparameter tuning, various scaling laws have been proposed to predict pretraining loss or downstream performance by capturing empirical relationships between key hyperparameters and model performance. A notable example is the Chinchilla scaling law (Hoffmann et al., 2022), which approximates the final pretraining loss as a simple function of the model size $N$ and total training steps $T$ (or total training tokens), $\mathcal{L}(N, T) = L_0 + A \cdot T^{-\alpha} + B \cdot N^{-\beta}$. By fitting parameters $L_0, A, B, \alpha, \beta$ from a few training runs with varying $N$ and $T$, one can use the formula to infer the optimal choice of $N$ and $T$ given a fixed compute budget $C \propto NT$.

A key challenge that existing scaling laws have not addressed is how to set the **Learning Rate (LR)** optimally over time. LR is arguably the most critical hyperparameter in optimization, as it can significantly affect the training speed and stability. A large LR can quickly reduce the training loss, but in the long term, it may cause overshooting and oscillation along sharp directions on the loss

†Corresponding authors.




(a) Comparison of Optimized LR Schedules (b) Loss Curves for Different LR Schedules

Figure 1: Optimizing the LR schedule induces a schedule (Opt) better than cosine and WSD schedules. We conduct evaluation experiments on a 400M Llama-2 (Touvron et al., 2023) model trained over 12B tokens. Zoom-in regions facilitate the readers who are interested in the local details. **(a)** Our optimized schedule comprises constant and decay stages post-warmup, aligning with WSD (Hu et al., 2024). **(b)** Loss curves demonstrate that our optimized schedule outperforms cosine schedules and two major variants of WSD with tuned hyperparameters (WSD with exponential decay and WSDLD with linear decay).

landscape. In contrast, a small LR ensures a more stable training process but also slows down the convergence. Practitioners often balance these trade-offs by starting training with a large LR and then gradually reducing it over time, following a *Learning Rate schedule* (LR schedule) (Bengio, 2012). These LR schedules sometimes include a warmup phase at the beginning, where the LR linearly increases from zero to a large value over a few thousand steps, and only after this warmup phase does the LR start to decay. The most commonly used LR schedule in LLM pretraining is the cosine schedule (Loshchilov & Hutter, 2017), which decays the LR following a cosine curve. Other schedules include the cyclic (Smith, 2017), Noam (Vaswani, 2017), and Warmup-Stable-Decay (WSD) schedules (Hu et al., 2024), but there is no consensus on the optimal choice.

Existing scaling laws sidestep the complexity of LR schedules by fitting parameters on a fixed family of LR schedules. For instance, Hoffmann et al. (2022) fitted the parameters in the Chinchilla scaling law for training runs that have gone through the entire cosine LR schedule. As a result, it does not generalize well to other LR schedules, or even to the same schedule with early stopping. Moreover, existing scaling laws lack a term to account for LR schedules, limiting their ability to provide practical guidance on setting the LRs. This issue can become even more pronounced when scaling up training to trillions of tokens (Dubey et al., 2024; DeepSeek-AI et al., 2024), where the extreme cost of training makes it impractical to experiment with multiple LR schedules.

In this paper, we aim to quantify how LR schedules influence the evolution of training loss in LLM pretraining through empirical analysis. More specifically, we study the following problem, which we call the *schedule-aware loss curve prediction* problem: *Can we use a simple formula to accurately predict the training loss curve* $\mathcal{L}(t)$ $(1 \le t \le T)$ *given a LR schedule* $E := \{\eta_1, \eta_2, \ldots, \eta_T\}$ *for* $T$ *steps of training?* To align with standard practices in LLM pretraining and to enable a more precise analysis tailored to this setting, we impose the following reasonable restrictions on the problem. First, we take fresh samples from a data stream at each training step, so there is no generalization gap between the training and test loss. Second, we focus on LR schedules that decay the LR over time, i.e., $\eta_1 \ge \eta_2 \ge \eta_3 \ge \cdots$. Finally, as most LR schedules used in practice start with a warmup phase before the LR decays, we make a minor modification to the problem and include a fixed warmup phase before the decay phase we are interested in. We assume that the shape and the peak LR $\eta_{\max}$ of the warmup phase have been carefully picked, potentially through a series of short training runs, and we are only interested in understanding how different LR decay schedules after warmup affect the training loss curve. For convenience, we shift the time index so that $t = 1$ corresponds to the first step after the warmup phase.

In contrast to most existing scaling laws that rely on only two or three hyperparameters (Kaplan et al., 2020; Hoffmann et al., 2022; Muennighoff et al., 2023; Goyal et al., 2024), solving the above problem poses unique challenges, as it requires predicting the loss curve based on the entire LR schedule, which is inherently high-dimensional. This complexity necessitates a more sophisticated approach to understand and quantify the relationship between the LR schedule and the loss curve.

**Our Contribution: Multi-Power Law.** In this paper, we propose the following empirical law (1) for schedule-aware loss curve prediction:

$$\mathcal{L}(t) = L_0 + A \cdot (S_1(t) + S_W)^{-\alpha} - \mathrm{LD}(t), \quad \text{where} \quad S_1(t) := \sum_{\tau=1}^{t} \eta_\tau. \tag{1}$$

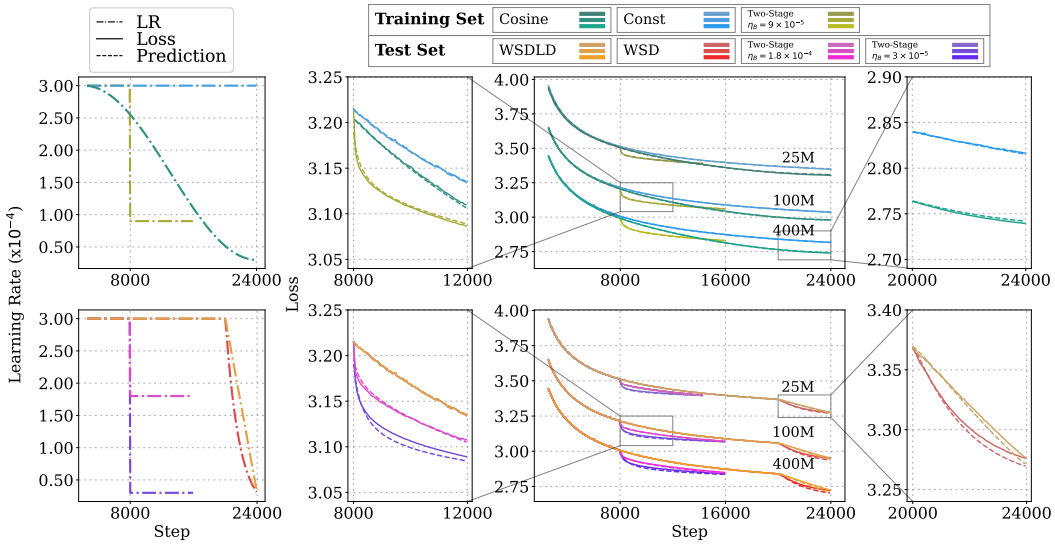



Figure 2: The Multi-Power Law (MPL) with parameters fitted on cosine, constant, and two-stage schedules can accurately predict the loss curves of unseen schedules, including WSDLD, WSD, and two-stage schedules with a different LR in the second stage. See Table 1 for evaluation metrics.

Here, $S_W$ denotes the sum of learning rates used in the warmup phase. The expression $L_0 + A \cdot (S_1(t)+S_W)^{-\alpha}$ can be viewed as an extension of the Chinchilla scaling law by replacing the number of steps $T$ with the cumulative sum of learning rates up to step $t$, while neglecting the dependence on the model size. While this alone provides a crude approximation of the loss curve by linearizing the contribution of the LR at each step (see Section 3.1 for further discussion), it does not account for the specific shape of the LR decay. The additional term $\mathrm{LD}(t)$ serves as a correction term, which captures the effect of LR decay in further reducing the loss:

$$\mathrm{LD}(t) := B \sum_{k=1}^{t} (\eta_{k-1} - \eta_k) \cdot G(\eta_k^{-\gamma} S_k(t)), \quad S_k(t) := \sum_{\tau=k}^{t} \eta_\tau, \quad G(x) := 1 - (Cx+1)^{-\beta}. \tag{2}$$

More specifically, $\mathrm{LD}(t)$ is linear with a cumulative sum of the LR reductions $\eta_{k-1} - \eta_k$ over time, scaled by a nonlinear factor $G(\eta_k^{-\gamma} S_k(t))$. This factor gradually saturates to a constant as the training progresses, which follows a power law in a scaled sum of learning rates $\eta_k^{-\gamma} S_k(t)$.

We call this law of $\mathcal{L}(t)$ the *Multi-Power Scaling Law (MPL)* as it consists of multiple power-law forms. $L_0, A, B, C, \alpha, \beta, \gamma$ are the parameters of the law and can be fitted by running very few pretraining experiments with different LR schedules. Our main contributions are as follows:

1. We propose the Multi-Power Law (1) for schedule-aware loss curve prediction, and empirically validate that after fitting the parameters of the law on at most 3 pretraining runs, it can predict the loss curve for unseen LR schedules with remarkable accuracy (see Figure 2). Unlike the Chinchilla scaling law, which relies solely on the final loss of each training run to fit its parameters, our approach utilizes the entire loss curve of each training run to fit the parameters, thus significantly reducing the number of training runs and compute resources needed for accurate predictions (Figure 5). Extensive experiments are presented for various model architectures, sizes, and training horizons (Section 4).
2. Our Multi-Power Law is accurate enough to be used to search for better LR schedules. We show that by minimizing the predicted final loss according to the law, we can obtain an optimized LR schedule that outperforms the standard cosine schedule. Interestingly, the optimized schedule has a similar shape as the recently proposed WSD schedule (Hu et al., 2024), but its shape is optimized so well that it outperforms WSD with grid-searched hyperparameters (Section 5).
3. We use a novel "bottom-up" approach to empirically derive the Multi-Power Law. Starting from two-stage schedules, we conduct a series of ablation studies on LR schedules with increasing complexity, which has helped us to gain strong insights into the empirical relationship between the LR schedule and the loss curve (Section 3).
4. We present a theoretical analysis for quadratic loss functions and show that the Multi-Power Law can arise when the Hessian and noise covariance matrices have a power-law decay in their eigenvalues (Appendix B).

## 2 PRELIMINARY

**Learning Rate Schedule.** A learning rate (LR) schedule is a sequence $E := \{\eta_1, \ldots, \eta_T\}$ that specifies the LR at each step of the training process. For language model pretraining, the cosine LR schedule (Loshchilov & Hutter, 2017) is the most popular schedule, which can be expressed as $\eta_t = \frac{1+\alpha}{2}\eta_{\max} + \frac{1-\alpha}{2}\eta_{\max}\cos(\frac{\pi t}{T})$. Here, $\eta_{\max}$ is the peak LR and $\alpha$ is usually set to 0.1. The Warmup-Stable-Decay (WSD) schedule (Hu et al., 2024) is a recently proposed LR schedule. This schedule first goes through a warmup phase, then maintains at a stable LR $\eta_{\max}$ with $T_{\text{stable}}$ steps, and finally decays in the form of $f(t-T_{\text{stable}})\eta_{\max}$ for $T_{\text{stable}} \leq t \leq T_{\text{total}}$. Here $f(x) \in (0,1)$ can be chosen as linear or exponential decay functions. We visualize these two LR schedules in Figure 1(a).

**Warmup Phase.** Many LR schedules, such as WSD, include a warmup phase in which the LR gradually increases from 0 to the peak LR $\eta_{\max}$ over a few thousand steps. We denote the number of warmup steps as $W$. By default, the LR increases linearly, so the total LR sum during warmup is given by $S_W = \frac{1}{2}\eta_{\max}W$. Our analysis focuses on the training process after the warmup, where the LR is decaying in almost all LR schedules. We count training steps starting from the end of warmup and set $t = 1$ as the first step after warmup. Accordingly, $\{\eta_1, \ldots, \eta_T\}$ represents the post-warmup schedule, and the LR at the last warmup step $\eta_0 = \eta_{\max}$ is the peak LR of the entire schedule.

**Power Law of Data Scaling** Prior studies (Hoffmann et al., 2022; Kaplan et al., 2020) demonstrate that, for a fixed model size, the final loss follows a power law of the data size or, equivalently, the total training step number $T$ in a constant-batch-size setting. This relationship is expressed as:

$$\mathcal{L}(T) \approx \hat{\mathcal{L}}(T) := L_0 + \tilde{A} \cdot T^{-\alpha}, \tag{3}$$

where $L_0, \tilde{A}, \alpha$ are parameters to fit. This law is typically fitted over the final losses of a set of training curves generated from a specific LR schedule family, such as a cosine schedule with a given peak LR ($\eta_{\max}$), ending LR ($\alpha\eta_{\max}$) and warmup steps ($W$). However, applying (3) directly to intermediate steps ($t < T$) introduces bias, as the LR schedule up to $t$ bears insufficient decay compared to the full schedule over $T$, resulting in different loss trajectories. This discrepancy is confirmed in Figure 5(b). We refer to (3) as the Chinchilla Data Scaling Law (abbreviated as CDSL) throughout the paper since it is simplified from the Chinchilla scaling law (Hoffmann et al., 2022) to highlight the data dimension.

## 3 EMPIRICAL DERIVATION OF THE MULTI-POWER LAW

In this section, we present the empirical derivation of the Multi-Power Law (MPL) for schedule-aware loss curve prediction. Our key insights are summarized as follows:

1. If two training runs share the same sum of learning rates, $\sum_{t=1}^{T}\eta_t$, then their final losses tend to be similar, though a non-negligible discrepancy remains (Section 3.1).
2. In particular, for a training run with a given LR schedule, the final loss $\mathcal{L}(T)$ is similar to that of another training run using a constant learning rate schedule with the same total LR sum. This motivates us to decompose $\mathcal{L}(T)$ into two components: (1) the final loss of the corresponding constant LR run; and (2) a residual term that captures the effect of LR decay, defined as the difference between the final loss of the target run and the constant LR run. (Section 3.1)
3. Empirically, we observe that training runs with constant learning rates exhibit a Chinchilla-like power-law behavior in the loss curve and can thus be well approximated by a simple power law. (Section 3.2.1)
4. To approximate the residual term, instead of analyzing it directly, we imagine a sequence of training runs with schedules that gradually transition from a constant LR to the target schedule, all while maintaining the same total LR sum. Using a novel "bottom-up" approach, we derive an approximation formula for the loss difference introduced by each incremental change in the LR schedule, first by analyzing simple two-stage schedules and then extending the results to more complex schedules. (Sections 3.2.2 and 3.3)

Finally, we sum up all the approximation terms above, leading to our MPL. Below, we elaborate on our approach in detail.

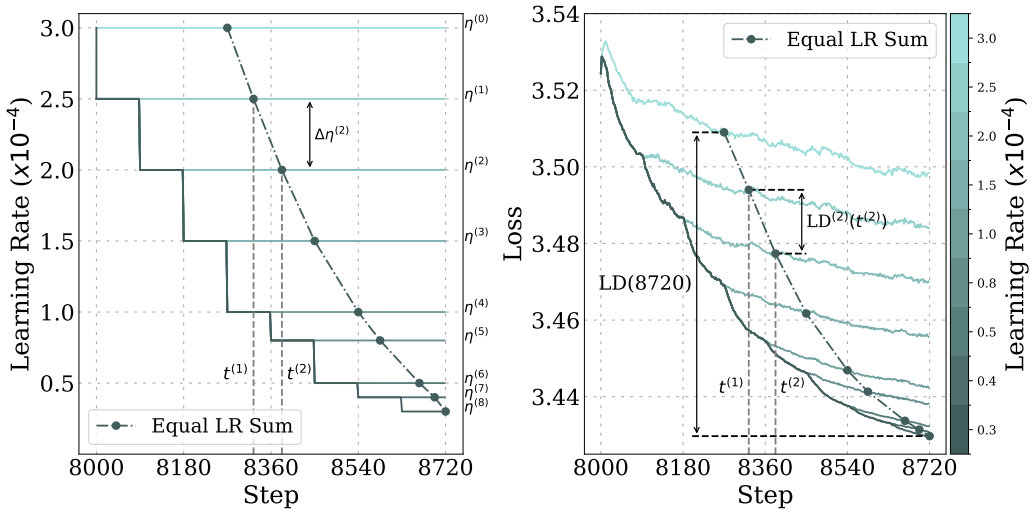



Figure 3: A multi-stage schedule (Appendix A.2) example to illustrate the learning rate (LR) sum matching (Section 3.1) and fine-grained loss reduction decomposition (Section 3.2.2). The step points with the equal LR sum as the final step $T_9 = 8720$ are marked and linked with the dash-point line. Each stage spans 90 steps. $T_1 = 8000$, $T_2 = 8090$, $t^{(1)} = Z_{T_2}(T_9)$, $t^{(2)} = Z_{T_3}(T_9)$. See Appendix G.3 for experiment details. **Left:** The actual multi-stage schedule and schedules for auxiliary processes. LR gap between adjacent points denotes the LR reduction $\Delta\eta^{(i)} = \eta^{(i-1)} - \eta^{(i)}$. **Right:** Corresponding training curves for the multi-stage schedule and the auxiliary processes. The total loss reduction is $\mathrm{LD}(T_9)$ and can be decomposed as the intermediate loss reduction sum. The loss gap between adjacent points denotes the stage-wise loss reduction $\mathrm{LD}^{(i)}(t^{(i)})$.

### 3.1 Our Approach: Learning Rate Sum Matching

**Auxiliary Training Process.** As introduced above, we construct a series of auxiliary training runs with LR schedules gradually changing from a constant LR schedule to the target schedule $E := \{\eta_1, \ldots, \eta_T\}$. Our construction is detailed as follows. We define the *$l$-th auxiliary process* shares the first $l$ steps of learning rates, $\{\eta_1, \ldots, \eta_l\}$, with the actual training process with LR schedule $E$, and continues with the constant LR $\eta_l$ afterwards. The corresponding loss curve for the $l$-th auxiliary process is denoted as $\mathcal{L}_l(t)$. In particular, the 0-th auxiliary process shares only the warmup phase with the actual training process and uses a constant LR $\eta_0 = \eta_{\max}$ after warmup. We especially call it the *constant process* and use $\mathcal{L}_{\mathrm{const}}(t)$ to represent its loss curve. The $T$-th auxiliary process coincides with the actual training run with the target LR schedule, so $\mathcal{L}_T(t) = \mathcal{L}(t)$.

**Learning Rate Sum Matching Decomposition** The Multi-Power Law (MPL) approximates the loss curve $\mathcal{L}(t)$ of the actual training process through the following decomposition. We define $Z(t)$ as the equivalent step in a constant LR process that shares the same cumulative LR sum as the actual process up to step $t$, where $Z(t) = \frac{S(t)}{\eta_0}$ and $S(t) = \sum_{\tau=1}^{t} \eta_\tau$ represents the sum of post-warmup LRs. The loss at step $t$ is then decomposed as:

$$\mathcal{L}(t) = \mathcal{L}_{\mathrm{const}}(Z(t)) - \underbrace{(\mathcal{L}_{\mathrm{const}}(Z(t)) - \mathcal{L}(t))}_{=:\ \mathrm{LD}(t)}, \tag{4}$$

where $\mathcal{L}_{\mathrm{const}}(Z(t))$ interpolates the loss for non-integer $Z(t)$ in the constant LR process. We first approximate $\mathcal{L}(t)$ using the training loss $\mathcal{L}_{\mathrm{const}}(Z(t))$ at step $Z(t)$, and then write the residual term $\mathrm{LD}(t)$ representing the approximation error. We call $\mathrm{LD}(t)$ the ***Loss reDuction term***, as it quantifies the loss reduction due to LR decay. We will approximate these two terms by parts in Section 3.2, with $\mathcal{L}_{\mathrm{const}}(Z(t))$ detailed in Section 3.2.1 and $\mathrm{LD}(t)$ in Section 3.2.2.

**Motivation: Continuous Approximations of the Training Dynamics.** The rationale behind this approach is that two training runs with the same LR sum should result in similar training losses, thus making it natural to decompose the loss curve into a major term corresponding to the loss of a run with the same LR sum and a small residual term. To see this, we use SGD as an example. If the learning rates $\eta_1, \ldots, \eta_T$ are small, then SGD can be seen as a first-order approximation of its continuous counterpart, gradient flow, under mild conditions (Li et al., 2017; Cheng et al., 2020; Elkabetz & Cohen, 2021). Here gradient flow describes a continuous-time process in which the parameters $\boldsymbol{\theta}(\tau)$ evolve according to the differential equation $\frac{\mathrm{d}\boldsymbol{\theta}(\tau)}{\mathrm{d}\tau} = -\nabla\mathcal{L}(\boldsymbol{\theta}(\tau))$, where $\nabla\mathcal{L}(\boldsymbol{\theta})$

is the gradient at $\boldsymbol{\theta}$, and $\tau$ denotes the continuous time. In this approximation, the $t$-th step of SGD corresponds to evolving $\boldsymbol{\theta}(\tau)$ over a small time interval of length $\eta_t$. When the learning rates are sufficiently small, the parameters after $t$ steps of SGD are close to $\boldsymbol{\theta}(\tau)$ at time $\tau = \sum_{k=1}^{t} \eta_k$. This connection naturally motivates us to compare the losses of two training runs with the same LR sum. While we use SGD for illustration, other optimization methods such as Adam can be similarly approximated by their continuous counterparts (Ma et al., 2022).

### 3.2 Approximation by Parts

#### 3.2.1 Constant Process Loss Approximation

Motivated by the continuous approximation of the training dynamics, we hypothesize that losses of constant LR processes with identical LR sums are closely aligned. This insight inspires us to represent $\mathcal{L}_{\text{const}}(Z(t))$ as a function of $S(t) + S_W$, where $S(t) + S_W$ represents the cumulative LR sum up to step $t$, including the warmup phase part $S_W$. Analogous to (3), we propose that $\mathcal{L}_{\text{const}}(Z(t))$ follows a power law over the LR sum:

$$\hat{\mathcal{L}}_{\text{const}}(Z(t)) = L_0 + A \cdot (S(t) + S_W)^{-\alpha}, \tag{5}$$

where $A$ is a parameter counterpart of $\tilde{A}$. We perform extensive empirical validation and ablation studies across different model sizes, training horizons, and learning rates to confirm the robustness of (5), as detailed in Appendix G.1 and illustrated in Figure 11.

#### 3.2.2 Loss Reduction Approximation

Now we turn to the loss reduction term $\text{LD}(t)$. We start by proposing a simple yet effective linear approximation as a warmup, then we further break down the term with a finer-grained LR sum matching approach.

**A Crude Linear Approximation.** We first generate training loss curves across various LR schedule types, including cosine and WSD schedules, alongside the loss curves of their corresponding constant processes. Then we can compute the loss reduction $\text{LD}(t)$ for different LR schedules and analyze their dependency. As demonstrated in Figure 10, $\text{LD}(t)$ is approximately proportional to the LR reduction, $\Delta\eta_k = \eta_0 - \eta_k$ across different schedules. This leads to the following approximation:

$$\text{LD}(t) \approx B(\eta_0 - \eta_k), \tag{6}$$

where $B$ is a constant. This finding highlights a strong correlation between the loss gap and the LR gap at equivalent LR sum points on the loss landscape. However, while the linear approximation offers insights into the shape of $\text{LD}(t)$, deviations from the actual loss reduction remain. Notably, when the LR decreases abruptly (e.g., in step-wise schedules), it predicts an instant loss drop at the stage switch, whereas the true loss decline remains smoother during the training process. See Appendix C for further discussion.

**Fine-Grained LR Sum Matching Decomposition.** In practice, the loss reduction term $\text{LD}(t)$ can have a more complex dependency on the LR schedule. To provide a more accurate approximation than the linear approximation above, we employ LR sum matching between adjacent auxiliary processes and decompose the loss reduction $\text{LD}(t)$ into the sum of *intermediate loss reductions* between adjacent auxiliary processes. We define the intermediate loss reduction between adjacent auxiliary processes as:

$$\text{LD}_k(t_{k+1}) := \mathcal{L}_k(A_k(t_{k+1})) - \mathcal{L}_{k+1}(t_{k+1}). \tag{7}$$

$\mathcal{L}_k$ and $\mathcal{L}_{k+1}$ are the loss curves for $k$-th and $(k+1)$-th auxiliary processes respectively. $A_k(t_{k+1})$ denotes the step in $k$-th auxiliary process, which has the same LR sum as step $t_{k+1}$ in the $(k+1)$-th auxiliary process. Then we consider the steps in all the auxiliary processes sharing the LR sum with step $t$ in the actual training. Thus, analogous to $Z(t)$, the equal-LR-sum step in $k$-th auxiliary process can be computed through

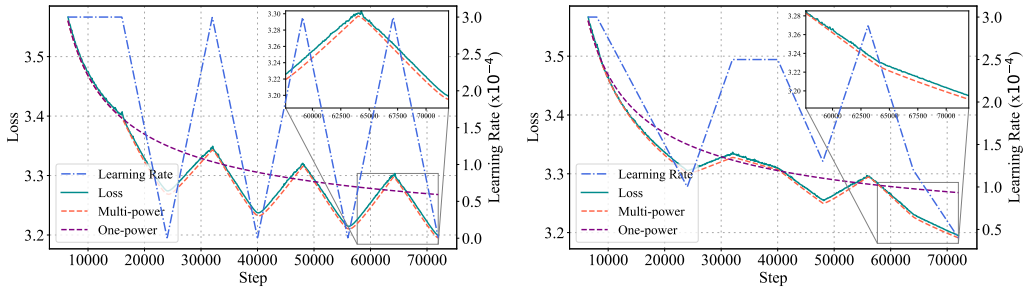
$$Z_k(t) = k - 1 + \frac{1}{\eta_k} S_k(t), \tag{8}$$

where $S_k(t) = \sum_{\tau=k}^{t} \eta_\tau$, representing the cumulative LR sum. Clearly, $Z_k(t)$ and $Z_{k+1}(t)$ have the same LR sum in the adjacent auxiliary processes, so we obtain

$$A_k(Z_{k+1}(t)) = Z_k(t). \tag{9}$$

Consequently, we can derive from (7) and (9):

$$\text{LD}_k(Z_{k+1}(t)) = \mathcal{L}_k(Z_k(t)) - \mathcal{L}_{k+1}(Z_{k+1}(t)). \tag{10}$$




(a) Cyclic Schedule (b) Random-Polyline Schedule

Figure 4: The examples of long-horizon non-monotonic schedules. The one-power line represents the constant process prediction. **(a)** The cyclic schedule with 72,000 steps, where each half-cycle spans 8,000 steps, and the first decay begins after 16,000 steps. **(b)** The random-polyline schedule, consisting of piecewise linear interpolation between randomly selected intermediate learning rates in the range of $3 \times 10^{-5}$ to $3 \times 10^{-4}$, with LR milestones occurring at intervals of 8,000 steps.

Finally, the total loss reduction can be decomposed as the sum of intermediate loss reductions:

$$\mathrm{LD}(t) = \mathcal{L}_{\text{const}}(Z(t)) - \mathcal{L}(t) = \mathcal{L}_0(Z_0(t)) - \mathcal{L}_t(Z_t(t)) = \sum_{k=0}^{t-1} \mathrm{LD}_k(Z_{k+1}(t)). \tag{11}$$

Here, $Z_0(t) = Z(t)$, ensuring that $\mathcal{L}_0(Z_0(t)) = \mathcal{L}_{\text{const}}(Z(t))$.

By leveraging this fine-grained decomposition, a refined estimation of $\mathrm{LD}_k(t_{k+1})$ enables a more precise approximation of $\mathrm{LD}(t)$. Where the context is clear, we simplify notation by omitting subscripts and denoting intermediate loss reduction as $\mathrm{LD}_k(t)$.

### 3.3 Bottom-Up Derivation: Two-Stage, Multi-Stage, and General Schedules

The challenges in approximating the intermediate loss reduction $\mathrm{LD}_k(t)$ are twofold. First, for commonly used schedules, the learning rate (LR) reduction at intermediate steps is often too small to induce a measurable loss reduction. Second, $\mathrm{LD}_k(t)$ may depend intricately on all previous learning rates $\{\eta_1, \ldots, \eta_k\}$, which we refer to as the *LR prefix* in this section. To address these issues, we derive the form of $\mathrm{LD}_k(t)$ using a "bottom-up" approach regarding schedule structures. Initially, we propose its form through schedules comprising two constant LR stages, leveraging significant LR reductions. Next, we examine its dependency on the LR prefix using schedules of multiple stages. Finally, we generalize the form to encompass all schedules and conclude with a multi-power law. The discussion of two-stage and multi-stage schedule is detailed in Appendices A.1 and A.2.

For general LR schedules, we extrapolate our findings from the two-stage and multi-stage cases in Appendices A.1 and A.2, and propose to approximate the intermediate loss reduction at step $k$ as the following power form:

$$\mathrm{LD}_k(t) \approx \widehat{\mathrm{LD}}_k(t) := B(\eta_k - \eta_{k+1}) \left(1 - \left(C\eta_{k+1}^{1-\gamma}(t-k) + 1\right)^{-\beta}\right), \tag{12}$$

with LR-prefix independent constants $B$, $C$, $\gamma$ and $\beta$.

Thus, the loss reduction between the constant process and the actual process can be approximated as

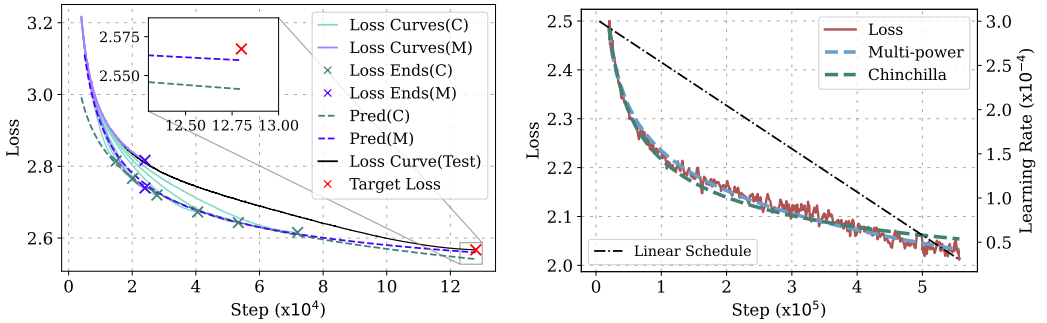
$$\widehat{\mathrm{LD}}(t) := \sum_{k=0}^{t-1} \widehat{\mathrm{LD}}_k(Z_{k+1}(t)) = \sum_{k=0}^{t-1} B(\eta_k - \eta_{k+1}) \left(1 - (C\eta_{k+1}^{1-\gamma}(Z_{k+1}(t) - k) + 1)^{-\beta}\right).$$

By the definition of $Z_k(t)$, we have $Z_{k+1}(t) - k = \frac{S_{k+1}(t)}{\eta_{k+1}}$. Therefore, we can conclude

$$\mathrm{LD}(t) \approx \widehat{\mathrm{LD}}(t) = \sum_{k=1}^{t} B\left(\eta_{k-1} - \eta_k\right)(1 - (C\eta_k^{-\gamma} S_k(t) + 1)^{-\beta}), \tag{13}$$

where we also change the subscript indices from $k+1$ to $k$. Combining the above ansatz for the loss reduction term with the power-law ansatz for the auxiliary loss in Equation (5) leads to our multi-power law:

$$\mathcal{L}(t) \approx L_0 + A \cdot (S_1(t) + S_W)^{-\alpha} - \sum_{k=1}^{t} B(\eta_{k-1} - \eta_k)(1 - (C\eta_k^{-\gamma} S_k(t) + 1)^{-\beta}). \tag{14}$$




(a) Fitting Sample Efficiency Comparison

(b) Whole Curve Fitting Comparison

Figure 5: **(a)** Target loss predictions at 128,000-step for a cosine schedule using MPL and CDSL fitting, with a 400M model. CDSL fitting requires six cosine losses (Loss Curve(C)) from 14,960 steps to 72,000 steps but relies solely on their final losses (Loss Ends(C)). In contrast, MPL leverages the entire 24,000-step constant and cosine loss curves (Loss Curves(M)). Final loss predictions are denoted as Pred(C) for CDSL and Pred(M) for MPL respectively. **(b)** Comparison of MPL and CDSL fittings on the whole loss curve of the open-source 7B OLMo model, trained with a linear schedule.

See Appendix C for the discussion about the simplification of the multi-power law.

## 4 Empirical Validation of the Multi-Power Law

The Multi-Power Law (MPL) comes from our speculations based on our experiments with special types of LR schedules. Now we present extensive experiments to validate the law for common LR schedules used in practice. Our experiments demonstrate that MPL requires only two or three LR schedules and their corresponding loss curves in the training set to fit the law. The fitted MPL can then predict loss curves for test schedules with different shapes and extended horizons.

### 4.1 Results

**Generalization to Unseen LR Schedules.** MPL can accurately predict loss curves for LR schedules outside the training set. As illustrated in Figure 2, despite the absence of WSD schedules in the training set and the variety of decay functions, MPL successfully predicts their loss curves with high accuracy. Furthermore, MPL can generalize to two-stage schedules with different $\eta_{\mathrm{B}}$ values from the training set, effectively extrapolating curves for both continuous and discontinuous cases.

**Generalization to Longer Horizons.** MPL demonstrates the ability to extrapolate loss curves for horizons exceeding three times the training set length. In our runs, the training set contains approximately 22,000 post-warmup steps, while the test set includes curves with up to 70,000 post-warmup steps. These results validate MPL's capability to generalize to longer horizons. Notably, the data-to-model ratio for a 25M-parameter model trained over 72,000 steps (36B tokens) is comparable to Llama2 pretraining (70B model, 2T tokens), consistent with trends favoring higher data volumes for fixed model sizes (Dubey et al., 2024).

**Generalization to Non-monotonic Schedules.** MPL extends effectively to complex non-monotonic schedules, although derived for monotonic decay schedules. We test the fitted MPL over challenging cases such as cyclic schedules and the *random-polyline schedule*, where LR values are randomly selected at every 8,000 steps and connected by a polyline. These experiments, conducted on a 25M-parameter model over 72,000 steps, also represent a demanding long-horizon scenario. As shown in Figure 4, MPL accurately predicts these long-horizon non-monotonic schedules, demonstrating its robustness and adaptability.

### 4.2 Comparison with Baselines

**Comparison with Chinchilla Law.** While Chinchilla-style data scaling laws, which we abbreviate as CDSLs, are widely adopted (Muennighoff et al., 2023; Hoffmann et al., 2022), MPL offers several distinct advantages: (1) MPL incorporates LR dependency, unlike CDSLs, and (2) MPL predicts the entire loss curve, whereas CDSLs are limited to estimating only the final loss. These advantages enable MPL to achieve higher sample efficiency than CDSLs. Notably, we demonstrate that a single constant and cosine schedule curve suffices to fit MPL with strong generalization. As illustrated in Figure 5(a), MPL reduces final loss prediction to less than $1/3$ that of CDSLs while requiring about $1/5$ compute budget. Furthermore, MPL excels in fitting the open-source 7B OLMo (Groeneveld

Table 1: Model performance comparison. $R^2$, MAE, RMSE, PredE, and WorstE are the coefficient of determination, Mean Absolute Error, Root Mean Square Error, Prediction Error, and Worst-case Error, respectively.

| Model Size | Method | $R^2 \uparrow$ | MAE $\downarrow$ | RMSE $\downarrow$ | PredE $\downarrow$ | WorstE $\downarrow$ |
|---|---|---|---|---|---|---|
| **25M** | Momentum Law | 0.9904 | 0.0047 | 0.0060 | 0.0014 | 0.0047 |
| | Multi-Power Law (Ours) | **0.9975** | **0.0039** | **0.0046** | **0.0012** | **0.0040** |
| **100M** | Momentum Law | 0.9959 | 0.0068 | 0.0095 | 0.0022 | 0.0094 |
| | Multi-Power Law (Ours) | **0.9982** | **0.0038** | **0.0051** | **0.0013** | **0.0058** |
| **400M** | Momentum Law | 0.9962 | 0.0071 | 0.0094 | 0.0025 | 0.0100 |
| | Multi-Power Law (Ours) | **0.9971** | **0.0053** | **0.0070** | **0.0019** | **0.0070** |

Table 2: Downstream performance comparison for the cosine and our optimized schedules. Percentage changes (↑ or ↓) indicate relative improvements or regressions compared to the cosine schedule.

| Schedule | LAMBADA | HellaSwag | PIQA | ARC-E | $C^3$ | RTE |
|---|---|---|---|---|---|---|
| Cosine | 46.54 | 37.12 | **65.13** | 43.56 | 48.44 | 52.71 |
| **Optimized** | **48.71** (↑ 2.17%) | **37.74** (↑ 0.62%) | 65.07 (↓ 0.06%) | **44.09** (↑ 0.53%) | **50.30** (↑ 1.86%) | **53.79** (↑ 1.08%) |

et al., 2024), as shown in Figure 5(b). Additional details of the comparison with Chinchilla Law are provided in Appendix H.2.

**Comparison with Momentum Law.** The MPL outperforms the recently proposed Momentum Law(MTL) (Tissue et al., 2024)[1] in both accuracy and applicability to discontinuous learning rate schedules. While MTL incorporates LR annealing effects by modeling loss reduction through the momentum of LR decay, it indicates an exponential loss reduction for two-stage LR schedules, inconsistent with our observations (see Appendix A.1). Across the diverse schedules in the test set, MPL consistently outperforms MTL in both average and worst-case prediction accuracy, as summarized in Table 1. Additionally, for WSD schedules with linear LR decay, MPL more accurately captures the loss reduction trend during the decay stage, as highlighted in Figure 14(b), compared to MTL. Further details on MTL and its relationship to MPL can be found in Appendix C, with fitting specifcs provided in Appendix H.2.

## 5 The Multi-Power Law Induces Better LR Schedules

Due to the high cost of each pretraining run and the curse of dimensionality for LR schedules, it is generally impractical to tune the LR for every training step. To address this, we propose leveraging the Multi-Power Law (MPL) to predict the final loss as a surrogate function to optimize the entire LR schedule, achieving a lower final loss and outperforming the cosine schedule and WSD variants.

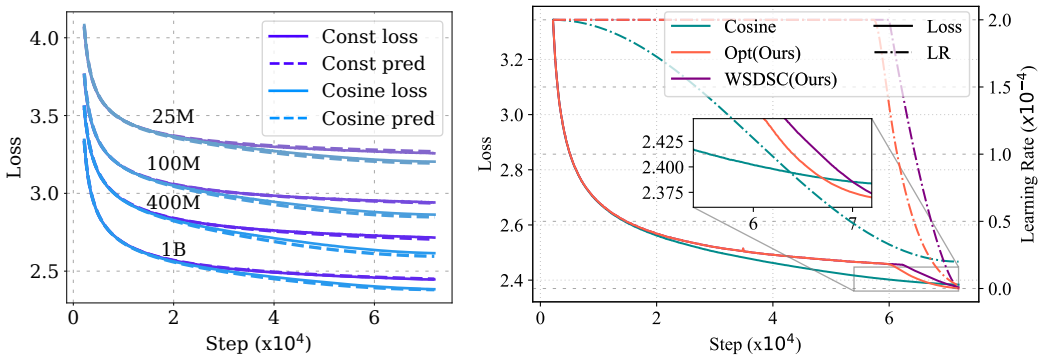
### 5.1 Method

The Multi-Power Law (MPL) provides an accurate loss estimation, enabling its final loss prediction to serve as a surrogate for evaluating schedules. We represent the learning rate (LR) schedule as a $T$-dimensional vector $E = (\eta_1, \ldots, \eta_T)$, with the final loss denoted as $\mathcal{L}(E)$ under given hyperparameters. Our goal is to find the optimal LR schedule $E^* = \arg\min_E \mathcal{L}(E)$. Using MPL, we parameterize the predicted final loss as $\mathcal{L}_\Theta(E)$ with parameters $\Theta = \{L_0, A, B, C, \alpha, \beta, \gamma\}$, estimated as outlined in Section 4. We approximate $E^*$ by optimizing the surrogate loss $\mathcal{L}_{\hat{\Theta}}(E)$ subject to monotonicity constraints:

$$\hat{E} = \min_E \mathcal{L}_{\hat{\Theta}}(E) \quad \text{s.t.} \quad 0 \leq \eta_t \leq \eta_{t-1}, \forall\, 1 \leq t \leq T. \tag{15}$$

This optimization induces an "optimal" schedule $\hat{E}$ derived from MPL with parameter $\hat{\Theta}$. We set the peak LR $\eta_0 = 3 \times 10^{-4}$ and assume $\eta_t$ is monotonically non-increasing, reflecting established training practices. We view $E$ as a high-dimensional vector and optimize it using the Adam optimizer. Further details are provided in Appendix I. Results for a 400M model are shown in Figure 1, with additional experiments for 25M and 100M models in Figure 18.

[1]Concurrent work. Early versions of our work are available at https://openreview.net/pdf?id=KnoS9XxIlK(October 2024).




(a) Long-Horizon Prediction of MPL

(b) Loss Curves Comparison for 1B Models

Figure 6: **(a)** Long-horizon loss predictions using MPL for cosine and constant schedules, with model sizes ranging from 25M to 1B (top to bottom). **(b)** Loss curve comparison for 1B models across the optimized schedule (Opt), cosine schedule (Cosine), and simplified optimized schedule (WSDSC, see Section 5.2), featuring a WSD schedule with sqrt-cube decay.

### 5.2 Results

**Optimized LR Schedule Exhibits Stable-Decay Pattern.** The optimized LR schedule follows a Warmup-Stable-Decay (WSD) structure, comprising two main post-warmup phases: a stable phase with a constant peak LR, and a decay phase ending with a lower LR, as illustrated in Figures 1 and 18. By contrast, the momentum law (Tissue et al., 2024) theoretically yields a collapsed learning rate schedule, as proved in Appendix J. However, unlike traditional WSD schedules (Hu et al., 2024), which decays linearly or exponentially to 1/10 of the peak LR, our optimized schedule reaches lower ending learning rates, typically below $1/20$ of the peak, even close to zero. Using normalized steps $\tilde{t}$ and normalized learning rates $\tilde{\eta}_{\text{avg}}$, We find that the decay function of the optimized schedule roughly follows $\tilde{\eta}_{\text{avg}} = (1 - \tilde{t})^{1.5}$, capturing the near-zero ending LR ($\tilde{t} = 1$, $\tilde{\eta}_{\text{avg}} = 0$).

**Optimized LR Schedule Outperforms Cosine Schedules.** Across comparison experiments of different model sizes and training steps, our optimized schedules consistently outperform the cosine schedules, achieving a margin exceeding 0.02. Notably, no WSD-like schedule is present in the training set, highlighting MPL's extrapolation capability. Figure 19 extends this comparison to longer training horizons and Figure 6(b) validates the superiority for 1B model. we further validate the effectiveness of our optimized schedules by evaluating the downstream task performance. As shown in Table 2, our optimized schedule leads to overall improvements in downstream tasks against the cosine schedules, showing practical gains from loss improvements. Ablation details for longer horizons and larger models are in Appendix I.

**Optimized LR Schedule Outperforms Tuned WSD Variants.** For a 400M model, the decay step of a 24000-step optimized schedule (Figure 1) is close to the optimally tuned step ( 6,000) for WSD and WSDLD schedules, determined via grid search over {3,000, 4,000, 5,000, 6,000, 7,000}. However, it surpasses these decay-ratio-tuned variants, suggesting that tuning the decay ratio alone is insufficient. Adjusting the ending LR to near-zero (see Appendix I) or altering the decay function also falls short. We propose a WSD variant with sqrt-cube decay (WSDSC), whose decay function is $\tilde{\eta}_{\text{avg}} = (1 - \tilde{t})^{1.5}$. WSDSC is effective across various model sizes and architectures, as evidenced in Figures 6(b) and 15(a), offering an alternative decay function for WSD schedules. Yet, it still falls short of the optimized schedule (Figure 6(b)), possibly due to untuned decay ratios. See Appendix I for more details.

## 6 Conclusions

This paper proposes the Multi-Power Law (MPL) to capture the relationship between loss and LR schedule, derived bottom-up from stage-wise schedules using LR sum matching decomposition and. The fitted MPL can accurately predict the entire loss curve while reducing the computational cost of fitting compared to traditional scaling laws. Through a theoretical analysis of a quadratic loss function, We discuss the possible underlying mechanism for MPL. Furthermore, we get optimized schedules via minimizing the predicted final loss of MPL, and extensively validate their superiority over commonly used schedules, thereby improving training efficiency.

## REFERENCES


Armen Aghajanyan, Lili Yu, Alexis Conneau, Wei-Ning Hsu, Karen Hambardzumyan, Susan Zhang, Stephen Roller, Naman Goyal, Omer Levy, and Luke Zettlemoyer. Scaling laws for generative mixed-modal language models. In International Conference on Machine Learning, pp. 265–279. PMLR, 2023.

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

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.

Yoshua Bengio. Gradient-based optimization of hyperparameters. Neural computation, 12(8):1889–1900, 2000.

Yoshua Bengio. Practical Recommendations for Gradient-Based Training of Deep Architectures, pp. 437–478. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-35289-8. doi: 10.1007/978-3-642-35289-8_26. URL https://doi.org/10.1007/978-3-642-35289-8_26.

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

James Bergstra, Dan Yamins, David D Cox, et al. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. SciPy, 13:20, 2013.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In Proceedings of the AAAI conference on artificial intelligence, volume 34, pp. 7432–7439, 2020.

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

David Brandfonbrener, Nikhil Anand, Nikhil Vyas, Eran Malach, and Sham Kakade. Loss-to-loss prediction: Scaling laws for all datasets. arXiv preprint arXiv:2411.12925, 2024.

Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. Nature communications, 12(1):2914, 2021.

Xiang Cheng, Dong Yin, Peter Bartlett, and Michael Jordan. Stochastic gradient and langevin processes. In International Conference on Machine Learning, pp. 1810–1819. PMLR, 2020.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. arXiv preprint arXiv:1803.05457, 2018.

Hugo Cui, Bruno Loureiro, Florent Krzakala, and Lenka Zdeborová. Generalization error rates in kernel regression: The crossover from the noiseless to noisy regime. Advances in Neural Information Processing Systems, 34:10131–10143, 2021.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J.L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R.J.

Chen, R.L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S.S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W.L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X.Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y.K. Li, Y.Q. Wang, Y.X. Wei, Y.X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z.F. Wu, Z.Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report. arXiv preprint arXiv:2412.19437, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. arXiv preprint arXiv:2407.21783, 2024.

Omer Elkabetz and Nadav Cohen. Continuous vs. discrete optimization of deep neural networks. Advances in Neural Information Processing Systems, 34:4947–4960, 2021.

Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. In International conference on machine learning, pp. 1165–1173. PMLR, 2017.

Jonas Geiping and Tom Goldstein. Cramming: Training a language model on a single gpu in one day. In International Conference on Machine Learning, pp. 11117–11143. PMLR, 2023.

Sachin Goyal, Pratyush Maini, Zachary C. Lipton, Aditi Raghunathan, and J. Zico Kolter. Scaling laws for data filtering– data curation cannot be compute agnostic. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 22702–22711, June 2024.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. Olmo: Accelerating the science of language models. arXiv preprint arXiv:2402.00838, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752, 2023.

Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro Von Werra, and Martin Jaggi. Scaling laws and compute-optimal training beyond fixed training durations. arXiv preprint arXiv:2405.18392, 2024.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.

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.

Peter J Huber. Robust estimation of a location parameter. In Breakthroughs in statistics: Methodology and distribution, pp. 492–518. Springer, 1992.

Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. Selected Papers 5, pp. 507–523. Springer, 2011.

Marcus Hutter. Learning curve theory. arXiv preprint arXiv:2102.04074, 2021.

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

Ayush Jain, Andrea Montanari, and Eren Sasoglu. Scaling laws for learning with real and surrogate data. arXiv preprint arXiv:2402.04376, 2024.

Yuchen Jin, Tianyi Zhou, Liangyu Zhao, Yibo Zhu, Chuanxiong Guo, Marco Canini, and Arvind Krishnamurthy. Autolrs: Automatic learning-rate schedule by bayesian optimization on the fly. arXiv preprint arXiv:2105.10762, 2021.

Arlind Kadra, Maciej Janowski, Martin Wistuba, and Josif Grabocka. Scaling laws for hyperparameter optimization. In Thirty-seventh Conference on Neural Information Processing Systems, 2023. URL https://openreview.net/forum?id=ghzEUGfRMD.

Arlind Kadra, Maciej Janowski, Martin Wistuba, and Josif Grabocka. Scaling laws for hyperparameter optimization. Advances in Neural Information Processing Systems, 36, 2024.

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, 2020. URL https://arxiv.org/abs/2001.08361.

Aaron Klein, Stefan Falkner, Jost Tobias Springenberg, and Frank Hutter. Learning curve prediction with bayesian neural networks. In International conference on learning representations, 2022.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. Journal of Machine Learning Research, 18(185):1–52, 2018.

Qianxiao Li, Cheng Tai, and E Weinan. Stochastic modified equations and adaptive stochastic gradient algorithms. In International Conference on Machine Learning, pp. 2101–2110. PMLR, 2017.

Zhiyuan Li and Sanjeev Arora. An exponential learning rate schedule for deep learning. arXiv preprint arXiv:1910.07454, 2019.

Licong Lin, Jingfeng Wu, Sham M Kakade, Peter L Bartlett, and Jason D Lee. Scaling laws in linear regression: Compute, parameters, and data. arXiv preprint arXiv:2406.08466, 2024.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In International Conference on Learning Representations, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

Chao Ma, Lei Wu, and Weinan E. A qualitative study of the dynamic behavior for adaptive gradient algorithms. In Joan Bruna, Jan Hesthaven, and Lenka Zdeborova (eds.), Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference, volume 145 of Proceedings of Machine Learning Research, pp. 671–692. PMLR, 16–19 Aug 2022. URL https://proceedings.mlr.press/v145/ma22a.html.

Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In International conference on machine learning, pp. 2113–2122. PMLR, 2015.

Alexander Maloney, Daniel A Roberts, and James Sully. A solvable model of neural scaling laws. arXiv preprint arXiv:2210.16859, 2022.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. Scaling data-constrained language models. Advances in Neural Information Processing Systems, 36:50358–50376, 2023.

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

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

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. arXiv preprint arXiv:1606.06031, 2016.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9, 2019.

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

Utkarsh Sharma and Jared Kaplan. A neural scaling law from the dimension of the data manifold. arXiv preprint arXiv:2004.10802, 2020.

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

Leslie N Smith. Cyclical learning rates for training neural networks. In 2017 IEEE winter conference on applications of computer vision (WACV), pp. 464–472. IEEE, 2017.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. Advances in neural information processing systems, 25, 2012.

Stefano Spigler, Mario Geiger, and Matthieu Wyart. Asymptotic learning curves of kernel methods: empirical data versus teacher–student paradigm. Journal of Statistical Mechanics: Theory and Experiment, 2020(12):124001, 2020.

Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. Investigating prior knowledge for challenging chinese machine reading comprehension. Transactions of the Association for Computational Linguistics, 8:141–155, 2020.

Yunfei Teng, Jing Wang, and Anna Choromanska. Autodrop: Training deep learning models with automatic learning rate drop. arXiv preprint arXiv:2111.15317, 2021.

Howe Tissue, Venus Wang, and Lu Wang. Scaling law with learning rate annealing. arXiv preprint arXiv:2408.11029, 2024.

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

A Vaswani. Attention is all you need. Advances in Neural Information Processing Systems, 2017.

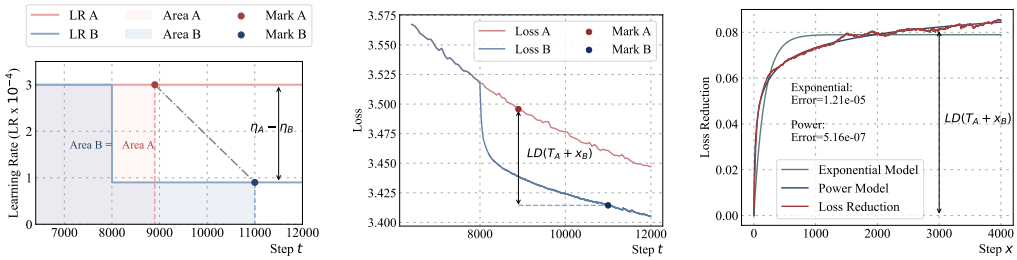
Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. Advances in neural information processing systems, 32, 2019.

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.

Zhen Xu, Andrew M Dai, Jonas Kemp, and Luke Metz. Learning an adaptive learning rate schedule. arXiv preprint arXiv:1909.09712, 2019.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? arXiv preprint arXiv:1905.07830, 2019.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 12104–12113, 2022.




(a) LR vs Step $t$ (b) Loss vs Step $t$ (c) Loss Reduction vs Step $x$

Figure 7: Loss reduction (LD) of two-stage schedule exhibits a power law. Example setting: $t_B = 11000$, $x_B = 3000$, $\eta_B = 9 \times 10^{-5}$, $\eta_A = 3 \times 10^{-4}$, $T_A = 8000$. **(a)** A and B have the equal LR sums: $x_A = 900$, $t_A = 8900$. **(b)** Loss reduction at $B$: $\mathrm{LD}(T_A + x_B) = \mathcal{L}_A(t_A) - \mathcal{L}_B(t_B)$. **(c)** Fitting loss reduction $\widehat{\mathrm{LD}}(T_A + x_B)$ with power form results in $0.13(1 - (1 + 0.21x)^{0.15})$; Fitting with exponential form results in $0.0790(1 - e^{-0.01x})$. The shape of loss reduction is closer to a power form than exponential.

# A Bottom-Up Derivation: Two-Stage, Multi-Stage (Section 3.3)

## A.1 Case 1: Two-Stage Learning Rate Schedule

The two-stage schedule keeps learning rates at $\eta_A$ for $T_A$ steps, directly drops to $\eta_B$, and continues for $T_B$ steps. Then the LR reduction $\eta_A - \eta_B$ could be significant enough to induce $\mathrm{LD}_{T_A}(t)$, which is also the loss reduction $\mathrm{LD}(t)$ for step $t$ on Stage 2. See Appendix G.2 for experiment details.

**Loss Reduction Term Follows a Power Law.** As shown in Figure 7, the number of steps $x := t - T_A$ in Stage 2 increases, $\mathrm{LD}(T_A + x)$ monotonically rises from 0 to around 0.09 and eventually saturates. This motivates us to approximate $\mathrm{LD}(T_A + x)$ in the form $\tilde{B} \cdot (1 - U(\eta_B x))$, where $\tilde{B}$ is a parameter and $U(s)$ is a function that decreases from 1 to 0 as $s = \eta_B x$ increases from 0 to infinity. The reason we choose $\eta_B x$ instead of $x$ as the argument of $U$ will be clear in the general case.

But at what rate should $U(s)$ decrease? After trying different forms of $U(s)$ to fit $\mathrm{LD}(T_A + x)$, we find that the power-law form $U(s) = (\tilde{C} \cdot s + 1)^{-\beta}$ for some $\tilde{C}, \beta > 0$ fits most properly as shown in Figure 7, which leads to the following power-law form for the loss reduction term:

$$\mathrm{LD}(T_A + x) \approx \widehat{\mathrm{LD}}(T_A + x) := \tilde{B}(1 - (\tilde{C} \cdot \eta_B x + 1)^{-\beta}). \tag{16}$$

Appendix A.1 shows that this power law aligns well with the actual loss reduction term $\mathrm{LD}(T_A + x)$. In contrast, the exponential form $U(s) = e^{-Bs}$ (so $\mathrm{LD}(T_A + x) \approx A(1 - e^{-B\eta_B x})$) struggles to match the slow and steadily increase of $\mathrm{LD}(T_A + x)$ when $x$ is large.

**Parameter Pattern of Power Law.** We further investigate how to estimate the parameters $\tilde{B}, \tilde{C}, \beta$ in the power law. Based on our preliminary experiments, we set $\beta = 0.4$, a constant that works well. Then we conduct experiments to understand how the best parameters $\tilde{B}, \tilde{C}$ to fit $\mathrm{LD}(t)$ depend on $\eta_A, \eta_B, T_A$, where we set default values $\eta_A = 3 \times 10^{-4}$, $\eta_B = 3 \times 10^{-5}$, $T_A = 8000$ and change one variable at a time. The details of ablation experiments can refer to Appendix G.2. The observations are summarized as follows.

(1) **$\tilde{B}$ is Linear to LR Reduction**. As shown in the first row of Figure 8, $\tilde{B}$ linearly decreases with $\eta_B$ and approximately increases linearly with $\eta_A$, especially when $\eta_A$ is not too large. Moreover, the slope of $\tilde{B}$ over $\eta_A$ and $\eta_B$ are approximately opposite to each other. This motivates us to hypothesize that $\tilde{B} \propto \eta_A - \eta_B$ and reparameterize $\tilde{B}$ as $\tilde{B} = B(\eta_A - \eta_B)$, where $B$ is a constant.

(2) **$\tilde{C}$ Follows a Power Law of $\eta_B$.** As shown in the second row of Figure 8, $\tilde{C}$ is very sensitive to $\eta_B$ but much less dependent on $\eta_A$. We hypothesize that $\tilde{C}$ follows a power law $\tilde{C} \propto \eta_B^{-\gamma}$, and reparameterize $\tilde{C}$ as $\tilde{C} = C\eta_B^{-\gamma}$, where $C > 0$ and $\gamma > 0$ are constants.

(3) **LR Reduction Term Depends Less on $T_A$.** We also find that $\tilde{B}$ and $\tilde{C}$ are less sensitive to $T_A$, relatively stable as $T_A$ varies, as shown in the last column in Figure 8. This suggests that the loss reduction has a weak dependency of loss reduction on LR prefix length.

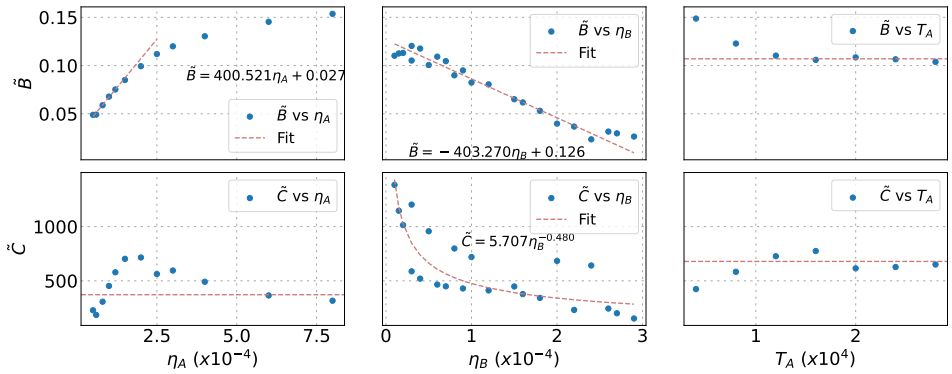



Figure 8: The dependency patterns of $\tilde{B}$, $\tilde{C}$ over $\eta_A$, $\eta_B$ and $T_A$ in the two-stage cases. $\tilde{B}$ is approximately proportional to $\eta_A - \eta_B$, and $\tilde{C}$ manifests power-law pattern over $\eta_B$. The dependency of $\eta_A$ over $\tilde{C}$ and the impacts of $T_A$ on $\tilde{B}$, $\tilde{C}$ are unpredictable or negligible, which are approximately ignored in our discussion.

**Approximation Form.** Putting all the above observations together, we have the final approximation form for the loss reduction term in the two-stage schedule:

$$\mathrm{LD}(T_A + x) \approx \widehat{\mathrm{LD}}(T_A + x) := B(\eta_\mathrm{A} - \eta_\mathrm{B}) \left(1 - (C\eta_\mathrm{B}^{1-\gamma} x + 1)^{-\beta}\right). \tag{17}$$

### A.2 Case 2: Multi-Stage Learning Rate Schedule

In the two-stage case, the LR prefix is constant at $\eta_\mathrm{A}$, leaving uncertainty about whether the intermediate loss reduction conforms to the power form when the LR prefixes vary. To investigate this, we analyze the multi-stage step decay schedule. Consider an $n$-stage LR schedule $E = \{\eta_1, \ldots, \eta_T\}$, where the $i$-th stage spans from step $T_i+1$ to $T_{i+1}$ and uses the LR $\eta^{(i)}$ ($0 \le T_1 < \cdots < T_{n+1} = T$, with $\eta_0 = \eta^{(0)} > \eta^{(1)} > \cdots > \eta^{(n)}$, $1 \le i \le n$). An example is illustrated in Figure 3.

**Stage-Wise Loss Reduction.** In the multi-stage schedule, given stage index $1 \le i \le n$, the stage-wise loss reduction is defined as $\mathrm{LD}^{(i)}(t) = \mathrm{LD}_{T_i}(t)$[2]. The LR reduction between stages, $\Delta\eta^{(i)} = \eta^{(i-1)} - \eta^{(i)}$, is also measurable. Using this, we estimate the shape of $\mathrm{LD}^{(i)}(t)$ for different stages. Regard $T_i$ as $T_\mathrm{A}$ in the two-stage case and define $x := t - T_i$. As shown in Figure 9(a), $\mathrm{LD}^{(i)}(T_i + x)$ approximately conforms to a similar power law as (16) for the two-stage case:

$$\mathrm{LD}^{(i)}(T_i + x) \approx \widehat{\mathrm{LD}}^{(i)}(T_i + x) := \tilde{B}^{(i)} \left(1 - \left(\tilde{C}^{(i)} \cdot \eta^{(i)} x + 1\right)^{-\beta}\right), \tag{18}$$

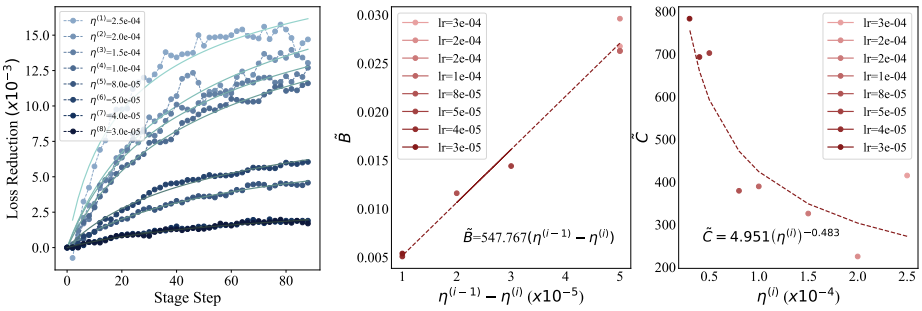
where $\tilde{B}^{(i)}$ and $\tilde{C}^{(i)}$ are constants dependent on the LR prefix $\{\eta_1, \ldots, \eta_{T_i}\}$ for stage $i$.

**Intermediate Loss Reduction Weakly Depends on the LR Prefix Shape.** For stage $i$, the LR prefix is $\{\eta_1, \ldots, \eta_{T_i}\}$, which varies in length and scale across stages. To evaluate the effect of the LR prefix on the intermediate loss reduction form, we examine its impact on $\tilde{B}^{(i)}$ and $\tilde{C}^{(i)}$. Interestingly, as shown in Figure 9(b), we observe that $\tilde{B}^{(i)} \approx B(\eta^{(i-1)} - \eta^{(i)})$ and $\tilde{C}^{(i)} \approx C(\eta^{(i)})^{-\gamma}$, which align closely with the two-stage results. Here, $B$, $C$, and $\gamma$ are constants largely independent of the stage index. This suggests that intermediate loss reductions are relatively insensitive to the LR prefix compared to the LR reductions $\Delta\eta^{(i)}$ and the stage LR $\eta^{(i)}$. Moreover, this weak dependence on the LR prefix may extend to general schedules, indicating a broader applicability of the power-law form for intermediate loss reduction.

## B How Might the Multi-Power Law Arise?

In this section, we present a preliminary theoretical analysis to understand how the multi-power law might arise. More specifically, we consider a simple setting where SGD optimizes a quadratic loss function with certain gradient noise, and show that the multi-power law naturally emerges when the Hessian and noise covariance matrices exhibit power-law decay in their eigenvalues. While this analysis does not capture the full complexity of deep learning, we believe it sheds light on how the multi-power law is related to certain power-law structures in the optimization landscape.

[2]Note that $\mathrm{LD}^{(i)}(t) = \mathrm{LD}_{t^{(i)}}(t)$ for each $T_i + 1 \le t^{(i)} \le T_{i+1}$, as these auxiliary processes for a specific stage coincide.




(a) Power Fitting of $\mathrm{LD}^{(i)}(t)$ (b) $\mathrm{LD}^{(i)}(t)$ Parameter Patterns

Figure 9: The intermediate loss reductions of a multi-stage schedule (Figure 3) and their shape patterns. **(a)** The loss reduction $\mathrm{LD}^{(i)}(t)$ between the adjacent stages of the multi-stage schedules still follows the power form. **(b)** $\tilde{B} \propto \eta^{(i-1)} - \eta^{(i)}$, $\tilde{C} \propto (\eta^{(i)})^{-\gamma}$. The parameter patterns in the two-stage setting hold in the multi-stage setting approximately. The shape of patterns is similar to the patterns in the two-stage experiments, as shown in Figure 8.

### B.1 Setup

We consider a quadratic loss function $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_*)^\top \boldsymbol{H}(\boldsymbol{\theta} - \boldsymbol{\theta}_*)$, where $\boldsymbol{\theta} \in \mathbb{R}^d$ represents the trainable parameters, $\boldsymbol{\theta}_*$ is the ground truth and $\boldsymbol{H} \in \mathbb{R}^{d\times d}$ is the Hessian matrix. Linear regression is a special case of this formulation. More generally, any loss function can be locally approximated by such a quadratic form near the optimum.

We use SGD with LR schedule $E = \{\eta_1, \ldots, \eta_T\}$ to optimize the loss function, where the $t$-th iteration is given by $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \eta_t \boldsymbol{g}_t$, with $\boldsymbol{g}_t$ being the stochastic gradient at step $t$. We assume that the stochastic gradient $\boldsymbol{g}_t$ equals the true gradient $\nabla\mathcal{L}(\boldsymbol{\theta}_t) = \boldsymbol{H}\boldsymbol{\theta}_{t-1}$ plus Gaussian noise $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ is the covariance matrix. We use $\Phi(\boldsymbol{\theta}_0, E)$ to denote the distribution of the $T$-th iteration $\boldsymbol{\theta}_T$ of SGD starting from $\boldsymbol{\theta}_0$.

**From spectra to scaling law for the loss.** We now aim to analyze the scaling behavior of the loss for the quadratic loss function defined above during training. This behavior is typically determined by the eigenvalue spectrum of the Hessian and the spectrum of the diagonal elements of the noise covariance matrix $\boldsymbol{\Sigma}$ in the gradient noise. Specifically, if we make certain assumptions about the Hessian matrix $\boldsymbol{H}$ and the noise covariance matrix $\boldsymbol{\Sigma}$, similar to the previous works (Canatar et al., 2021; Spigler et al., 2020; Maloney et al., 2022; Cui et al., 2021; Brandfonbrener et al., 2024), we can show that the loss follows a multi-power law.

**Assumption 1.** *Let $\lambda_i$ be the $i$th eigenvalue of $\boldsymbol{H}$, and $\Sigma_{ii}$ be the element of $\boldsymbol{\Sigma}$ in the $i$th column and $i$th row. $\lambda_i \overset{i.i.d.}{\sim} \boldsymbol{p}(\lambda) \propto \lambda^\alpha$, $\Sigma_{ii} \overset{i.i.d.}{\sim} \boldsymbol{q}(\Sigma)$ for all $i \in \{1, 2, \ldots, d\}$, where $\alpha > -1$ and $\lambda \in [0, D]$. Also, given some $\rho \in \mathbb{R}$ and $\mu \in \mathbb{R}^+$, we have that*

$$\mathbb{E}_{\boldsymbol{q}}[\Sigma|\lambda] \propto \lambda^\rho \exp(-G\lambda), \qquad \mathbb{E}_{\boldsymbol{q}}[\Sigma] = \mu,$$

*where $D, G$ are positive constants independent of LR schedule $E$.*

### B.2 Loss Formula

The following theorem provides an accurate estimate of the expected loss at step $t$.

**Theorem 1.** *Under Assumption 1, given $\boldsymbol{\theta}_T \sim \Phi(\boldsymbol{\theta}_0, E)$, and $S_1(t) > \frac{1}{\eta_{\max}}$, we have the following estimate of $\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_t)]$ for any $0 \le t \le T$:*

$$\tilde{M}_t(\boldsymbol{\theta}_0, E) := L_0 + A \cdot S_1(t)^{-\alpha-2} - B\sum_{k=2}^{T}(\eta_{k-1} - \eta_k)\left(1 - (C\, S_k(t) + 1)^{-\alpha-\rho-1}\right),$$

*where $L_0 = \frac{d}{4}\eta_{\max}\mu$, and $A = \frac{\|\boldsymbol{\theta}_0\|_2^2 \mu\gamma(\alpha+2, D)}{2^{\alpha+3} Z_\lambda}, B = \frac{d\mu}{4}, C = \frac{2}{G} \ge 0$ are constants independent of LR schedule $E$, $\gamma(\cdot,\cdot)$ denotes the lower incomplete gamma function such that $\gamma(s, x) := \int_0^x t^{s-1}e^{-t}\mathrm{d}t$and $S_i(t) := \sum_{k=i}^{t}\eta_k$, and $Z_\lambda$ is the partition function for probability measure. The estimation error is bounded as*

$$|\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_t)] - \tilde{M}_t(\boldsymbol{\theta}_0, E)| = O(S_1(t)^{-\alpha-1} + \eta_{\max}^2).$$

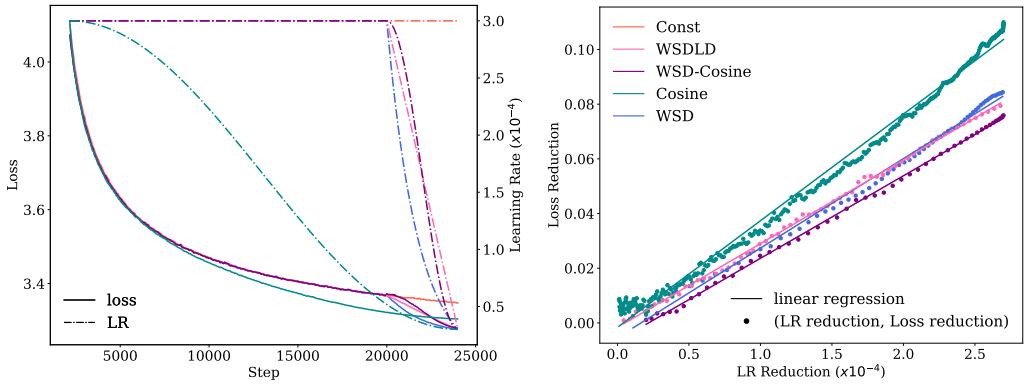



Figure 10: Linear regression of loss reduction versus LR reduction across different schedules for a 25M model over 24,000 steps. Decay steps are set at 4,000 for WSD and its variants, among which WSD-Cosine specifically denotes the WSD schedule with cosine decay function. **Left:** Visualization of learning rate schedules and their corresponding loss curves. **Right:** Scatter plot of loss reductions against LR reductions, accompanied by a linear regression fit (mean $R^2 = 0.9980$), demonstrating a strong linear relationship between the two variables.

$A$ characterizes the overall size of the item $\mathcal{L}_{\text{const}}$, which is proportional to the distance of the initial parameter from the optimal parameter $\|\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*\|_2^2$, and also proportional to the expectation of noise $\mu$ during training. The term $B$ represents the influence of loss drop induced by learning rate reduction, which is proportional to the data dimension $d$ and noise expectation $\mu$. Compared to the original Multi-Power Law in Equation (14), we notice that the term $C\eta_k^{-\gamma}$ is replaced by a simpler constant $C$ in this theoretical equation, which is a little misalignment between theory and practice. This theorem shows that there exists a theoretical setting where Multi-Power arises in the learning curve. The detailed proof of Theorem 1 can be found in Appendix K.

Beyond this quadratic case, to get more systematic theoretical results, which are also more realistic, we should take inspiration from data and the loss landscape side. Recent work proposes a river-valley loss landscape perspective based on sharpness analysis, to understand the advantage of the WSD schedules (Wen et al., 2024).

## C Formula Component Ablation

Table 3: Summary of fitting results for simplified versions and variants of the MPL. Metrics include $R^2$, MAE, RMSE, PredE, and WorstE, where higher $R^2$ values and lower values of other metrics indicate better fitting performance. See Table 1 for metric definitions.

| Formula | Features | $R^2 \uparrow$ | MAE↓ | RMSE↓ | PredE↓ | WorstE↓ |
|---|---|---|---|---|---|---|
| OPL | $LD(t) = 0$ $(B = 0)$ | 0.8309 | 0.0378 | 0.0412 | 0.0111 | 0.0241 |
| LLDL | $G(x) = 1$ | 0.9797 | 0.0077 | 0.0101 | 0.0023 | 0.0108 |
| No-$\gamma$ | $\gamma = 0$ | 0.9961 | 0.0046 | 0.0053 | 0.0014 | 0.0041 |
| SPL | $x = t - k$ | 0.9921 | 0.0066 | 0.0075 | 0.0020 | 0.0069 |
| MEL | $G(x) = 1 - e^{-Cx}$, $\gamma = 0$ | 0.9934 | 0.0044 | 0.0057 | 0.0013 | 0.0047 |
| MTL | $G(x) = 1 - e^{-Cx}$, $x = t - k$ | 0.9904 | 0.0047 | 0.0060 | 0.0014 | 0.0047 |
| **MPL (Ours)** | $G(x) = 1 - (Cx + 1)^{-\beta}$, $x = \eta_k^{-\gamma} \sum_{\tau=k}^{t} \eta_\tau$ | **0.9975** | **0.0039** | **0.0046** | **0.0012** | **0.0040** |

To understand and evaluate the role of each component in our Multi-Power Law (MPL; See (1)), we systematically simplify the MPL formula at various levels and explore alternative formulations. Table 3 summarizes the fitting performance of these simplified versions and variants of the MPL. The fitting experiments are conducted on 25M models, using the same experimental setup described in Appendix F.

**No Loss Reduction.** The necessity of the loss reduction term $LD(t)$ can be assessed by fitting a One-Power Law (OPL), a simplified MPL where $LD(t) = 0$ or equivalently $B = 0$:

$$\mathcal{L}_{\text{OPL}}(t) = L_0 + A \cdot (S_1(t) + S_W)^{-\alpha}, \quad S_1(t) := \sum_{\tau=1}^{t} \eta_\tau. \tag{19}$$

This formulation approximates the loss curve by matching the LR sum without correction term, as discussed in Section 3.1. The fitted results (first row of Table 3) exhibit significant degradation compared to the full MPL, demonstrating the critical role of $LD(t)$.

**Linear Approximation of Loss Reduction.** Based on the observation in Section 3.2.2, the loss reduction term $LD(t)$ (defined in Equation (2)) can be simplified by treating the scaling function $G(x)$ as a constant:

$$\text{LD}(t) \approx \sum_{k=1}^{t} B(\eta_{k-1} - \eta_k) = B(\eta_0 - \eta_t). \tag{20}$$

Despite its simplicity, we observe a near-linear relationship between $LD(t)$ and the LR reduction $(\eta_0 - \eta_t)$, regardless of the LR schedule type, as shown in Figure 10. This motivates the Linear Loss reDuction Law (LLDL):

$$\mathcal{L}_{\text{LLDL}}(t) = L_0 + A \cdot (S_1(t) + S_W)^{-\alpha} + B(\eta_0 - \eta_t). \tag{21}$$

As shown in Table 3, LLDL achieves significantly better accuracy than OPL, although it underperforms the full MPL. However, this formulation is unsuitable for optimizing schedules, as its results collapse to a trivial solution: $\eta_k = \eta_0$ when $k \leq T - 1$ and $\eta_k = 0$ when $k = T$.

**Loss Reduction Without $\gamma$.** Next, we simplify $G(x)$ by setting $\gamma = 0$, yielding the No-$\gamma$ Law:

$$\mathcal{L}_{\text{No}-\gamma} = L_0 + A \cdot (S_1(t) + S_W)^{-\alpha} + B \sum_{k=1}^{t} (\eta_{k-1} - \eta_k) \cdot G(S_k(t)). \tag{22}$$

Results (third row of Table 3) show a slight performance drop, confirming that $\gamma$ enhances fitting accuracy with minimal additional computational cost. Thus, we retain $\gamma$ in the final MPL.

**Step-Based Approximation.** An alternative is to replace $G(\eta_k^{-\gamma} S_k(t))$ with a step-based formulation, $G(t - k + 1)$. This yields the Step Power Law (SPL):

$$\mathcal{L}_{\text{SPL}} = L_0 + A \cdot (S_1(t) + S_W)^{-\alpha} + B \sum_{k=1}^{t} (\eta_{k-1} - \eta_k) \cdot G(t - k + 1). \tag{23}$$

While simpler, this approximation reduces prediction accuracy and contradicts empirical results, because it implies loss reduction continues to increase even when LR reaches zero.

**Exponential Approximation.** Substituting $G(x)$ with an exponential function $G(x) = 1 - e^{-Cx}$ gives the Multi-Exponential Law (MEL):

$$\mathcal{L}_{\text{MEL}} = L_0 + A \cdot (S_1(t) + S_W)^{-\alpha} + B \sum_{k=1}^{t} (\eta_{k-1} - \eta_k) \cdot G(S_k(t)). \tag{24}$$

Results (fifth row of Table 3) show a performance drop compared to the power-based MPL, consistent with observations in Appendix A.1 that $\tilde{U}(t, \eta_k)$ takes a power form rather than an exponential form.

**Relation to Momentum Law.** The concurrently proposed MomenTum Law (MTL) is in the form of

$$\mathcal{L}_{\text{MTL}}(t) = L_0 + A \cdot (S_1 + S_W)^{-\alpha} + B \cdot S_2, \text{ where } S_1 = \sum_{i=1}^{t} \eta_i, \; S_2 = \sum_{i=1}^{t} \sum_{k=1}^{i} (\eta_{k-1} - \eta_k) \lambda^{i-k},$$

where $\lambda$ is a hyper-parameter of MTL and $\lambda < 1$. It is indeed a variant of MPL since

$$S_2 = \sum_{i=1}^{t} \sum_{k=1}^{i} (\eta_{k-1} - \eta_k) \lambda^{i-k} = \sum_{k=1}^{t} (\eta_{k-1} - \eta_k) \sum_{i=k}^{t} \lambda^{i-k} = \sum_{k=1}^{t} (\eta_{k-1} - \eta_k) \left( \frac{1 - \lambda^{t-k+1}}{1 - \lambda} \right).$$

Thus, MTL is a variant of MPL with an exponential step-based approximation:

$$\mathcal{L}_{\text{MTL}}(t) = L_0 + A \cdot (S_1(t) + S_W)^{-\alpha} + B' \cdot G(t - k + 1), \quad G(x) = 1 - e^{-C'x}.$$

Here, $B' = \frac{B}{1-\lambda}$, $C' = -\log \lambda$. MTL incorporates step-based decay and its performance (last second row of Table 3) even lags behind MEL, highlighting the limitations of step-based approximations.

## D Limitation and Future Direction

Although our Multi-Power Law (MPL) exhibits excellent prediction accuracy and enables practical schedule optimization, certain limitations remain. The reliance on a pre-determined peak LR may limit the broader applicability of our Multi-Power Law. Furthermore, we observe slight deviations between our predictions and the actual training curves, which may arise from several simplifications in our derivation of the Multi-Power Law. These include assumptions that the coefficient $\beta$ remains constant across different LR scales, that intermediate loss reduction are independent of prior LR sequences, and that LR variations during the warm-up phase are not accounted for.

Our future work will pursue three objectives: (1) explore the theoretical understanding of the optimization landscape and scaling laws to uncover their underlying mechanisms, (2) investigate empirical laws with unfixed maximum learning rates and other hyperparameters, and (3) refine our Multi-Power Law to further enhance prediction accuracy and generalizability. We believe that enhancing the robustness and precision of this scaling law promises to further boost training efficiency, particularly for large-scale models.

## E Related Work

**Optimal Learning Rate Schedule.** Designing effective learning rate (LR) schedules is a prominent research focus in deep learning. Early work by Smith (2017) proposed a cyclical learning rate schedule. Loshchilov & Hutter (2017), inspired by warm restarts, introduced the cosine learning rate schedule, demonstrating its superiority across multiple experimental settings. From a theoretical perspective, Li & Arora (2019) introduced an exponential decay learning rate schedule based on the equivalence of weight decay. Xu et al. (2019); Teng et al. (2021) utilized reinforcement learning algorithms or Bayesian optimization for adaptive LR tuning. Pan et al. (2021) proposed an eigenvalue-dependent step schedule by incorporating the eigenvalue distribution of the objective function's Hessian matrix into the design of the learning rate schedule. Recently, Hu et al. (2024) introduced the Warmup-Stable-Decay (WSD) schedule, which starts with a warmup phase, continues a main stable phase, and ends with a rapid decay phase, showing good performance in LLM pretraining and enabling efficient continual training. Concurrent studies (Wen et al., 2024; Schaipp et al., 2025) analyze WSD schedules via the loss landscape structures. Geiping & Goldstein (2023); Zhai et al. (2022); Ibrahim et al. (2024); Hägele et al. (2024); Shen et al. (2024) also advocate slow-decay or stable phase followed by a rapid decay. Recent open-source models (DeepSeek-AI et al., 2024; OLMo et al., 2024) also adopt these kinds of stage-wise decay schedules. Additionally, a linear-to-zero decay is proposed to be optimal in recent work (Bergsma et al., 2025).

However, some of these papers rely on heuristic designs on LR schedules lacking a comprehensible, principled approach, while others try to optimize schedules within function subspaces, or draw conclusions based on some assumptions over the real-world data distribution to facilitate the loss landscape analysis. Consequently, the generality of these findings is often limited. Our paper seeks to open the door to a principled and comprehensible path of the optimal learning rate schedule design.

**Scaling Laws.** Scaling laws have arguably been the driving force behind the development of large language models. Initially proposed by Kaplan et al. (2020) and further developed by Hoffmann et al. (2022), Kadra et al. (2023), Aghajanyan et al. (2023) and Muennighoff et al. (2023), among others, most scaling laws adopt a power law form. However, due to the lack of dependence on the learning rate, these laws typically predict only the final loss of a training process, lacking guidance for the full training curve. This is because only the final loss bears a full LR decay while the LR decays at the intermediate steps are not sufficient. Typically, they need more than 10 training curves to obtain the scaling law of the final losses for one particular schedule type, the cosine schedule practically (Hoffmann et al., 2022; Muennighoff et al., 2023). As a comparison, we could fit the LR-dependent Multi-Power Law applicable across different LR schedule types within only 2-3 loss curves.

Several explanations for the power law form of scaling laws have been proposed, ranging from the perspective of data manifolds (Sharma & Kaplan, 2020) to the power law distribution of eigenvalues in the loss landscape (Lin et al., 2024). In our paper, we do not delve into the discussion about the model dimension scaling, we discuss the scaling along the data dimension and the LR dimension. We believe it offers new perspectives and a novel starting point for theoretical investigations.

**Hyperparameters Optimization.** Hyperparameter optimization has long been a focal point of research within the machine learning community, with early efforts like Bengio (2000) exploring gradient-based approaches (Bengio, 2000; Franceschi et al., 2017; Maclaurin et al., 2015) to improve the hyperparameter optimization. For learning rate schedules (LR schedule), early works primarily employed Bayesian optimization-based approaches (Hutter et al., 2011; Snoek et al., 2012; Bergstra et al., 2013) or bandit-based solutions (Li et al., 2018) to tune hyperparameters. These methods often model LR schedules as learnable constants or parametric function families, prioritizing theoretical and experimental simplicity over comprehensive exploration of the whole LR schedule space. More recent approaches, such as Teng et al. (2021); Jin et al. (2021), adjust LR schedule during training automatically, but these approaches cannot identify the optimal LR schedule before training, and they fail to fully generalize across different datasets, limiting their applicability to LLM training. In contrast, Klein et al. (2022) selects hyperparameters based on differences in learning curves for various hyperparameters, and Kadra et al. (2024) recognizes the power law phenomenon and develops HP methods based on power law. Our contribution advances this field by proposing a more robust scaling law than the power law specifically for the LR schedule dimension and present a comprehensive framework for optimizing LR schedule.

**Theory in Scaling Law.** Although there are numerous experimental studies on scaling law, the theoretical explanation of scaling law remains very limited. Sharma & Kaplan (2020) demonstrated that the exponent of the power law is related to the intrinsic dimension of the data in a specific regression task. Hutter (2021) examined a binary classification toy problem, deriving a scaling law with respect to data dimensionality for this problem. Jain et al. (2024) investigated scaling laws in the context of data selection. Bahri et al. (2024) assumed a power-law spectrum on the covariates, obtaining a scaling law with respect to data and model dimensions in the setting of least squares loss. Bordelon et al. (2024) considered scaling laws in regression problems under gradient flow. Atanasov et al. (2024) and Lin et al. (2024) discussed the formation of scaling laws in high-dimensional linear regression problems. Notably, our theoretical analysis provides a loss prediction throughout the training process from the perspective of the learning rate schedule, formally resembling the Multi-Power Law observed in our experiments.

## F Experiment Setting

Unless otherwise specified, the model training in the Section 3, 4 and 5 follows the following settings.

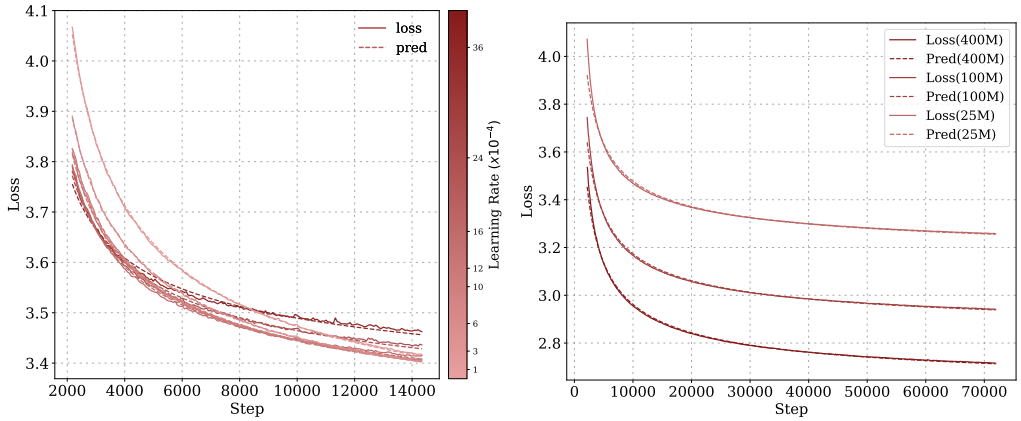
| Codename | Embedding Dimension | #Heads | #Layers | #Non-embeddings | #Params |
|---|---|---|---|---|---|
| 25M | 640 | 5 | 5 | 25 | 89 |
| 100M | 1024 | 8 | 8 | 101 | 205 |
| 400M | 1536 | 12 | 12 | 340 | 493 |
| 1B | 2048 | 32 | 16 | 822 | 1026 |
| GPT-2 | 768 | 12 | 12 | 85 | 162 |

Table 4: The model series run in all the experiments. Hoffmann et al. (2022) utilizes the number of non-embedding parameters (#Non-embeddings) to count model sizes, while Kaplan et al. (2020) counts the total number of parameters (#Params). The unit of the Parameter is M in this table.

Our validation contains two steps: (1) fitting schedule-curve pairs from the training set and (2) predicting the loss curves for schedules in the test set. The training set contains only a single 24,000-step constant and cosine schedule pair, alongside a 16,000-step two-stage schedule of $\eta_{\mathrm{B}} = 0.3\eta_{\mathrm{A}}$. The test set has one 72,000-step constant and cosine schedule, 24,000-step unseen WSD and WSDLD schedules, and 16,000-step two-stage schedules with $\eta_{\mathrm{B}} = 0.1\eta_{\mathrm{A}}$ and $\eta_{\mathrm{B}} = 0.6\eta_{\mathrm{A}}$. The details are provided in Table 6. We train Llama2 (Touvron et al., 2023) models of 25M, 100M, and 400M, and

| Default Hyperparameter | Value |
|---|---|
| Sequence Batch Size | 128 |
| Sequence Length | 4096 |
| Optimizer Type | AdamW |
| $\beta_1$ | 0.9 |
| $\beta_2$ | 0.95 |
| $\epsilon$ | $1 \times 10^{-8}$ |
| Weight Decay | 0.1 |
| Gradient Clipping | 1.0 |
| Peak Learning Rate | $3 \times 10^{-4}$ |
| Final Learning Rate | $3 \times 10^{-5}$ |
| Warmup Steps | 2160 |

Table 5: Hyperparameters related to model training.




(a) Peak LR Ablation for Constant Schedules (b) Model Size Ablation for Long-Horizon Training

Figure 11: Loss curves for constant LR schedules. *pred* denotes the fitted law prediction and *loss* represents the ground-truth loss curve. See Appendix G.1 for details.

collect their loss curves, with model parameter details in Table 4. Training employs the AdamW optimizer, with a weight decay of 0.1, gradient clipping at 1.0, $\beta_1 = 0.90$, and $\beta_2 = 0.95$, consistent with the Llama2 training setup. Default hyperparameters include a peak LR of $3 \times 10^{-4}$, a warmup period of 2160 steps, and a batch size of 0.5M. Additional hyperparameters are detailed in Table 5. In ablation studies, we simplify the experiment to fit short constant and cosine schedules and predict the loss for a long-horizon cosine schedule. The MPL fitting employs Huber loss (Huber, 1992) as the objection function, aligning with prior work (Hoffmann et al., 2022; Muennighoff et al., 2023), and uses the Adam optimizer for optimization. Unless otherwise specified, we report validation loss. For fitting approaches and additional details see Appendix H.

# G Discussions of Multi-Power Law Derivation (Section 3)

## G.1 Constant Process Loss Approximation (Section 3.1)

The constant process employs a constant LR schedule with the same warmup phase and peak LR as the actual process schedule. We validate (5), the LR sum power law of the loss curves for constant schedules, through two series of experiments. First, we conduct ablation over the peak LR, ranging from $3.0 \times 10^{-4}$ to $3.6 \times 10^{-3}$ over 14,400 steps, achieving an MSE of $1.55 \times 10^{-5}$ and $R^2$ of 0.9976 (Figure 11(a)). Second, we validate the power form over long-horizon curves (72,000 steps) for model sizes of 25M, 100M, and 400M, with a peak LR of $3.0 \times 10^{-4}$, yielding an average MSE of $8.04 \times 10^{-5}$ and $R^2$ of 0.9947 (Figure 11(b)).

### G.2 TWO-STAGE EXPERIMENTS (APPENDIX A.1)

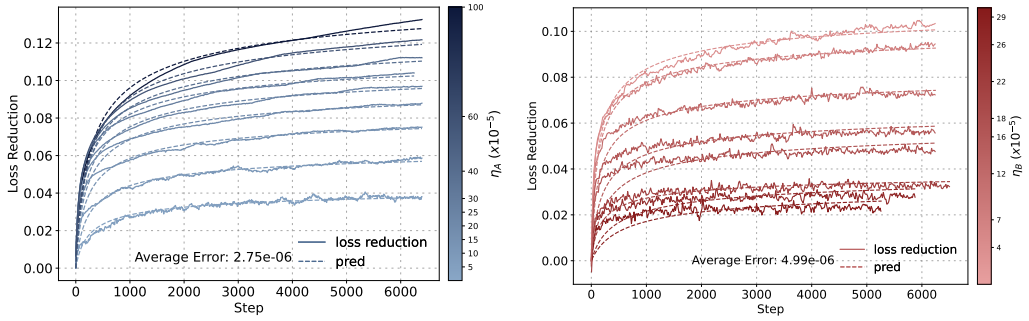
In this section, we provide details on the investigation of the variation of coefficients in the power law for two-stage LR schedules.

**Experiment Setting and Law Fitting.** The experiment setting aligns with Appendix F. Default configuration uses $\eta_{\mathrm{A}} = 3 \times 10^{-4}$, $\eta_{\mathrm{B}} = 3 \times 10^{-5}$, $T_{\mathrm{A}} = 8000$. In the ablation experiments, $\eta_{\mathrm{A}}$ ranges from $5 \times 10^{-5}$ to $1 \times 10^{-3}$, $\eta_{\mathrm{B}}$ ranges from $4 \times 10^{-5}$ to $2.9 \times 10^{-4}$, and $T_{\mathrm{A}}$ ranges from 4000 to 28000. The second stage lengths spanning 1,000 to over 6,000 steps. Validation loss is sampled every 2 steps due to the rapid loss changes after the stage switch. Following Hoffmann et al. (2022), we fit the law utilizing Huber loss as the objection function (Huber, 1992),

$$\min_{\Theta} \sum_{x} \mathrm{Huber}_{\delta}(\log \widehat{\mathrm{LD}}_{\Theta}(T_A + x) - \log \mathrm{LD}(T_A + x)), \tag{25}$$

where $\Theta = \{\tilde{B}, \tilde{C}, \beta\}$, and we set $\delta = 1 \times 10^{-2}$. For each experiment, we use the Adam optimizer with a learning rate at $1 \times 10^{-4}$ and total steps of 20000. Here we do not conform to the L-BFGS algorithm like Hoffmann et al. (2022) due to its sensitivity to the initialization. In our fitting, the parameters are initialized based on the loss reduction curve shape: $\tilde{B}$ corresponds to the estimation of asymptotic values of loss reduction and $\tilde{C}$ can be estimated according to the slope at $x = 0$ step (Equation (16)).

**Fixed $\beta$ Experiments for Parameter Patterns.** We fit the power-law form in Equation (16) across ablation experiments to identify the loss curve shape and power-law parameter patterns. For the sake of further derivation, we fix the exponent $\beta$ as LR-independent parameter 0.4 based on the warmup experiments. Then we re-fit the loss curves fixing $\beta = 0.4$ to confirm the validity of the power form. Figure 12 includes the re-fitted curves and ground truths for the ablation experiments over $\eta_{\mathrm{A}}$ and $\eta_{\mathrm{B}}$, showing feasible error margins for further derivation despite fixed $\beta$. We further investigate the dependency of different parameters on the $\eta_{\mathrm{A}}$, $\eta_{\mathrm{B}}$, and $T_{\mathrm{A}}$, with pair-wise relations presented in Figure 8 and summarized in Appendix A.1.




(a) $\eta_{\mathrm{A}}$ Ablation (b) $\eta_{\mathrm{B}}$ Ablation

Figure 12: Power-law fitting of loss reductions versus steps $x$ for two-stage LR schedules.

### G.3 MULTI-STAGE EXPERIMENTS (APPENDIX A.2)

We analyze intermediate loss reduction dependency on the LR prefix, through experiments of a multi-stage schedule and its auxiliary (intermediate) processes. As shown in Figure 3, our multi-stage schedule consists of 9 stages, with a first stage of 8,000 steps at $3 \times 10^{-4}$, followed by eight 90-step stages. The validation interval is also set to 2 steps. For adjacent stages $i-1$ and $i$ ($x \leq 90$), We compute $\mathrm{LD}^{(i)}(T_i + x)$, as defined in Appendix A.2 and fit it going through the same law fitting process as Equation (25). The fitting of loss reductions for different stages is presented in the left panel of Figure 9. Moreover, parameter trends, analogous to the two-stage findings, reveal $\tilde{B}^{(i)}$ changing with $\eta^{(i-1)} - \eta^{(i)}$ and $\tilde{C}^{(i)}$ changing with $\eta^{(i)}$, shown in the right two sub-figures of Figure 9.

# H Details of Validation Experiments (Section 4)

## H.1 Training Set and Test Set

| Set | Schedule Type | Total Lengths | $\eta_B/\eta_A$ |
|---|---|---|---|
| Training | Constant | 24,000 | |
| | Cosine | 24,000 | |
| | Two-stage | 16,000 | 0.3 |
| Test | WSD | 24,000 | |
| | WSDLD | 24,000 | |
| | Two-stage | 16,000 | 0.1 |
| | Two-stage | 16,000 | 0.6 |
| | Constant | 72,000 | |
| | Cosine | 72,000 | |

Table 6: Summary of training and test sets.

Our validation frames the Multi-Power Law (MPL) fitting as a machine learning task, training on schedule-loss curve pairs from the training set and predicting loss curves for the test set. The training set contains a 24,000-step constant and cosine schedule pair, and a 16,000-step two-stage schedule with $\eta_{\mathrm{B}} = 0.3\eta_{\mathrm{A}}$. The test set includes a 72,000-step constant and cosine schedule, a 24,000-step unseen WSD and WSDLD schedule, and 16,000-step two-stage schedules with $\eta_{\mathrm{B}} = 0.1\eta_{\mathrm{A}}$ and $\eta_{\mathrm{B}} = 0.6\eta_{\mathrm{A}}$. The peak learning rate is $3 \times 10^{-4}$, and the ending learning rate is $3 \times 10^{-5}$ for the cosine, WSD, and WSDLD schedules. For all two-stage schedules, $T_{\mathrm{A}} = 8000$. All schedules include a warmup phase of 2,160 steps. Detailed descriptions of the training and test sets are summarized in Table 6.

## H.2 Law Fitting

Similar to the two-stage fitting, we fit the parametric law using the Huber loss as the objective (Huber, 1992):

$$\min_{\theta} \sum_{t} \text{Huber}_{\delta}(\log \mathcal{L}_{\theta}(X_t) - \log \mathcal{L}_{\text{gt}}(X_t)), \tag{26}$$

where $\mathcal{L}_{\text{gt}}(X_t)$ denotes the ground truth of validation loss, and $\mathcal{L}_{\theta}(X_t)$ is the predicted loss, and $\delta$ is a hyperparameter for the Huber loss. The total fitting loss sums up the Huber loss over the validation steps. In practice, we compute the area under the linearly interpolated polyline of the learning rate at validation steps as a surrogate for the LR sum. This approach reduces the computational cost since requiring only step numbers, learning rates, and losses at validation steps.

**Multi-Power Law.** For the Multi-Power Law (MPL), $\theta = \{A, B, C, \alpha, \beta, \gamma, L_0\}$, and $X_t = \{\eta_1, \ldots, \eta_t\}$. We use the Adam optimizer to fit the MPL due to its flexibility, with a learning rate of $5 \times 10^{-3}$ for the index parameters ($\alpha$, $\beta$, and $\gamma$) and $5 \times 10^{-2}$ for the coefficient or constant parameters ($A$, $B$, $C$, and $L_0$). We also perform a second optimization with a learning rate of $1 \times 10^{-5}$ and $1 \times 10^{-6}$, initialized with parameters from the first optimization. Each optimization runs for $5 \times 10^4$ steps, selecting the lowest training loss result. Fitted parameters are listed in Table 7. In the discussion of Appendix C, we also fit simplified MPL or MPL variants in this manner, except for the momentum law (Appendix H.2). In Figure 13, we present the fitting and prediction results for a subset of experiments, with a zoom-in window highlighting predictions near the end of training. In long-horizon experiments, the zoomed-in view reveals slight discrepancies between the MPL predictions and the actual training curves, targeted for future refinement.

**Momentum Law.** For the momentum law (MTL; Appendix C), $\theta = \{A, B, \alpha, L_0\}$, with $\lambda$ as a tunable hyperparameter. The input $X_t$ for MTL is the same as MPL's input. Following Tissue et al. (2024), we use L-BFGS to minimize Equation (26), grid-searching $\lambda \in \{0.95, 0.99, 0.995, 0.999, 0.9995\}$ and selecting the best fit based on training accuracy. Predictions are evaluated across the test set (Table 6), with comparisons to MPL in Table 1 and Figure 14. In Figure 14, we compare them specifically over the WSDLD schedule. In the decay stage, MPL not

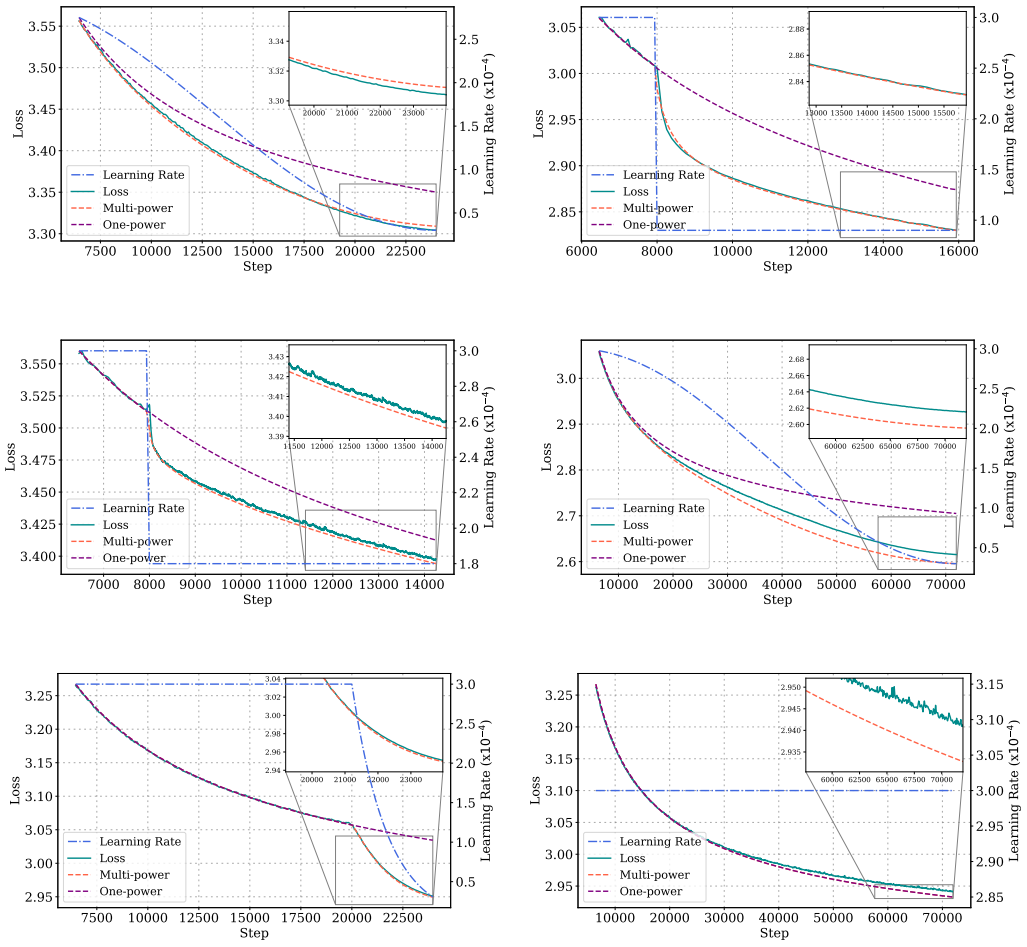
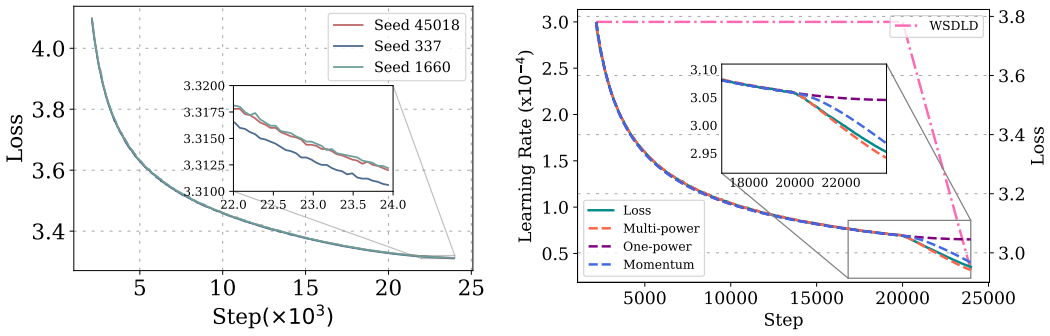



Figure 13: **Fitting and Prediction Details.** Subfigures depict loss curve fitting (training set) and prediction (test set) across various configurations, labeled as $(X, Y)$ for row $X$, column $Y$. The columns in the accompanying table indicate: **F/P** for Fitting (F) or Prediction (P), **Model Size**, **Step Length**, and **Learning Rate Schedule**. Subfigure details follow:

| $(X, Y)$ | F/P | Model Size | Step Length | LR Schedule |
|---|---|---|---|---|
| (1, 1) | F | 25M | 24,000 | Cosine |
| (1, 2) | F | 400M | 16,000 | 2-stage ($3 \times 10^{-4} \rightarrow 9 \times 10^{-5}$) |
| (2, 1) | P | 25M | 16,000 | 2-stage ($3 \times 10^{-4} \rightarrow 1.8 \times 10^{-4}$) |
| (2, 2) | P | 400M | 72,000 | Cosine |
| (3, 1) | P | 100M | 24,000 | WSD |
| (3, 2) | P | 100M | 72,000 | Constant |

Table 7: Parameter values for optimized schedules across different model sizes, rounded to two decimal places.

| **Model Size** | $A$ | $B$ | $C$ | $\alpha$ | $\beta$ | $\gamma$ | $L_0$ |
|---|---|---|---|---|---|---|---|
| 400M | 0.66 | 614.30 | 0.16 | 0.42 | 0.88 | 0.56 | 2.52 |
| 100M | 0.59 | 521.40 | 0.24 | 0.46 | 0.60 | 0.65 | 2.79 |
| 25M | 0.51 | 446.40 | 2.07 | 0.53 | 0.41 | 0.52 | 3.17 |




(a) Random Seed Ablation

(b) Comparison with Momentum Law

Figure 14: **(a)** Experiments with a 25M model over 24,000 steps across different seeds, showing a final loss standard deviation of 0.0007 and a maximum gap of 0.0014. **(b)** Comparison between Multi-Power Law (MPL) and Momentum Law (MTL). In the decay stage, MPL achieves higher fitting accuracy and matches the curvature of the loss curve, whereas MTL fits the stable stage but predicts a counterfactual concave curve during the decay stage.

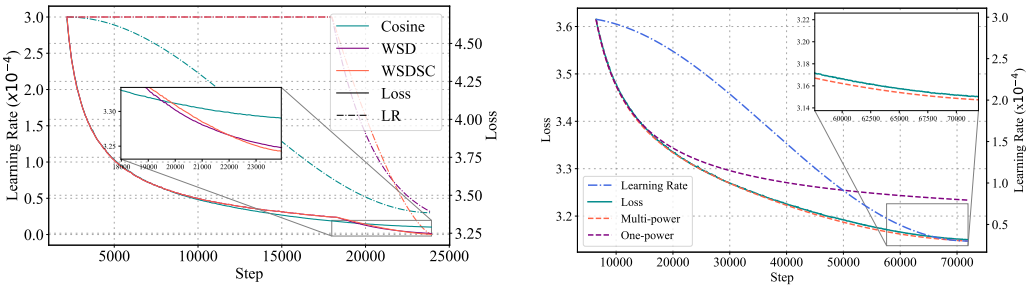
only achieves higher fitting accuracy but also aligns with the curvature of the loss curve. In contrast, MTL fits the stable stage well but predicts a counterfactual concave curve during the decay stage.

**Chinchilla Data Scaling Law.** The Chinchilla Data Scaling Law (CDSL) is similar to the one-power law mentioned in Appendix C, but uses the power of steps instead of the LR sum, with $\theta = \{A, \alpha, L_0\}$, and $X_t = t$ (final steps only) for Equation (26). The fitting of CDSL follows Hoffmann et al. (2022) and uses the L-BFGS algorithm to minimize the Huber loss. With regard to sample efficiency (Figure 5(a)), CDSL uses cosine curves at 14,960, 20,080, 27,760, 40,560, 53,360, and 72,000 steps, requiring 4.8 times more compute than MPL (two 24,000-step curves), with prediction errors of 0.007 (MPL) versus 0.024 (CDSL). MPL achieves less than one-third the prediction error of CDSL. In Figure 5(b), CDSL fits all intermediate steps, ignoring the effect of LR schedule and loss reductions for the comparison with MPL.

**Discussion on the Optimization Method.** We also explored the use of the L-BFGS algorithm for fitting MPL but found it highly sensitive to parameter initialization. For instance, under certain initializations, the fitted parameters may include a high $\beta$ value and a near-zero $C$. Note that $1 - (1 + Cx)^{-\beta} = 1 - \exp(-\beta \log(1 + Cx)) \approx 1 - \exp(-\beta Cx)$ in this case, making MPL resemble a multi-exponential form. In practice, this issue can be mitigated by constraining parameters such as $\beta$ and $\gamma$ to the interval $(0, 1)$. Additionally, we can initialize $C$, $\beta$, and $\gamma$ through grid search to obtain more feasible results. However, using the Adam optimizer is not without limitations, as it lacks theoretical convergence guarantees. Future work will focus on enhancing the fitting process to achieve greater robustness and stability.

### H.3 Ablation Experiments

We perform ablation studies over key hyper-parameters to assess the applicability and robustness of the Multi-Power Law (MPL). These hyperparameters include the model architectures, model sizes, peak learning rates, batch sizes, and random seeds, incorporating both self-conducted and open-source experimental results.




(a) Loss Curve Comparison for GPT-2 (b) Long-Horizon Prediction for GPT-2

Figure 15: Loss curves of GPT-2 models with Multi-Power Law fitted over 24,000-step constant and cosine schedule losses. **(a)** Comparison between the cosine, WSD, and WSDSC schedules (see Section 5.2); **(b)** Prediction for a 72,000-step cosine schedule loss curve.

**Model Architectures.** We validate MPL's generalizability across GPT-2 (Radford et al., 2019) and OLMo (Groeneveld et al., 2024) models to evaluate the generalizability of the MPL across model architectures. For GPT-2, we go through the simplified experiments, fitting the MPL on cosine and constant schedules of 24,000 steps and predicting the 72,000-step loss of the cosine schedule. The prediction result is illustrated in Figure 15 and model parameters align to GPT-2 small Radford et al. (2019), as detailed in Table 4. For the 7B OLMo model, we fit the MPL on the open-source training curve, which employs a linear decay schedule, as shown in Figure 5(b). Our results show that the MPL presents a high prediction accuracy across different model types for both self-run and open-source experiments.

**Model Size** To test MPL at scale, we train a 1B model on 144B tokens with a batch size of 2M-token and peak LR of $2 \times 10^{-4}$ for training stability. The model architecture matches Llama-3.2-1B (Dubey et al., 2024) and is detailed in Table 4. Simplified experiments involve fitting MPL to 24,000-step constant and cosine schedules and predicting the 72,000-step loss for both, as shown in Figure 6(a). Results demonstrate MPL's consistent performance across model sizes, as well as the robustness under a different peak LR and batch size.

**Peak Learning Rate** We further investigate MPL's robustness across varying peak learning rates. While prior experiments fixed the peak learning rate at $3 \times 10^{-4}$, empirical observations of two-stage schedules reveal deviations at higher rates, as illustrated in Figure 8. Therefore, we run full experiments with peak learning rates of $4 \times 10^{-4}$ and $6 \times 10^{-4}$ on 25M models, yielding average $R^2$ values of 0.9965 and 0.9940 respectively, underscoring MPL's consistently decent accuracy while accuracy degrading as peak LR goes up. The training and test set conform to Table 6 and the fitting results are shown in Figure 17.

**Batch Size** We extend experiments to batch sizes of 64 and 256 sequences on 25M models, complementing the prior 128-sequence (0.5M) results, with sequence length of 4096. MPL maintains high accuracy, with $R^2$ values exceeding 0.9970 across all cases, as illustrated in Figure 16. These experiments indicate that, while the coefficients of MPL are batch-size dependent, the functional form of MPL remains robust across varying batch size configurations

**Random Seeds** To assess the influence of random seed variability, we train a 25M model over 24,000 steps using cosine schedules with three distinct seeds. As shown in Figure 14(a), the loss values exhibit a standard deviation of approximately 0.001, establishing a lower bound for prediction error and highlighting MPL's precision.

## I Details of Optimized LR Schedule (Section 5)

**Optimizing the Surrogate Objective.** To enhance optimization stability, we redefine the learning rate schedule $E = \{\eta_0, \eta_1, \ldots, \eta_T\}$ using $\boldsymbol{d\eta} = \{d\eta_1, d\eta_2, \ldots, d\eta_T\}$, where $d\eta_i = \eta_{i-1} - \eta_i$. Thus, $\eta_i = \eta_0 - \sum_{k=1}^{i} d\eta_k$, establishing a one-to-one mapping between $E$ and $\boldsymbol{d\eta}$. We transform

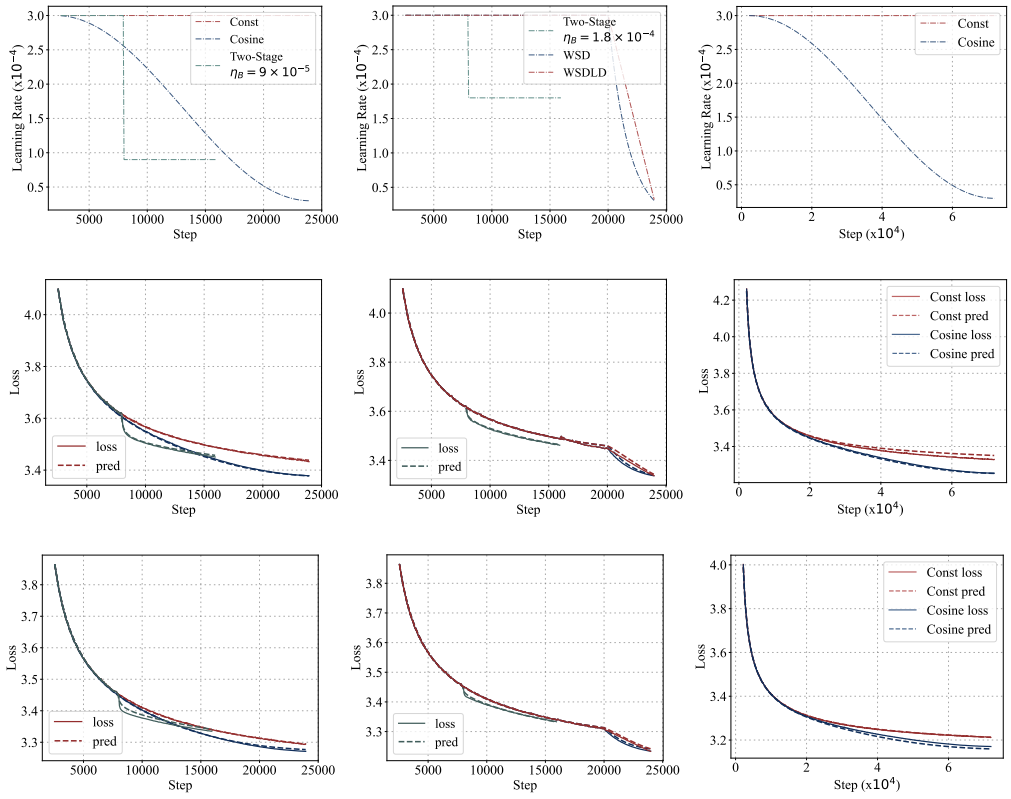



Figure 16: Ablation study on batch sizes, with $R^2$ values of 0.9977 (batch size 64) and 0.9973 (batch size 256). The subfigure layout is as follows. **Rows:** (1) Learning rate schedules, (2) Loss curves for batch size 64, (3) Loss curves for batch size 256. **Columns:** (1) Training set results, (2) Test set results (same horizon as training), (3) Test set results (extended horizon).

the objective $\mathcal{L}_{\hat{\Theta}}(E)$ in Equation (15) into $\tilde{\mathcal{L}}_{\hat{\Theta}}(\boldsymbol{d\eta})$, optimizing:

$$\begin{aligned} \min_{\boldsymbol{d\eta}} \quad & \tilde{\mathcal{L}}_{\hat{\Theta}}(\boldsymbol{d\eta}) \\ \text{s.t.} \quad & \sum_{i=1}^{T} d\eta_i \leq \eta_0, \\ & 0 \leq d\eta_i, \quad \forall i = 1, \ldots, T. \end{aligned}$$

In practice, we relax this to:

$$\begin{aligned} \min_{\boldsymbol{d\eta}} \quad & \tilde{\mathcal{L}}_{\hat{\Theta}}(\boldsymbol{d\eta}) \\ \text{s.t.} \quad & 0 \leq d\eta_i \leq \eta_0, \end{aligned}$$

enforcing constraints via clipping. This reformulation, applied to the MPL fitted from Appendix H.1, empirically stabilizes optimization by aligning learning rate reductions with zero initialization. For optimization, we use the Adam optimizer with a constant learning rate, grid searched from $2 \times 10^{-8}$ to $1 \times 10^{-9}$, over 50,000 to 200,000 for better convergence. The resulting $\boldsymbol{d\eta}$ satisfies the original constraint.

**Optimized Schedule of Longer Horizons and Different Model Sizes.** Beyond Figure 1 and Figure 18, we validate the optimized schedules for extended horizons and different model sizes. For models ranging from 25M to 400M, we optimize LR schedules for 72,000-step training based on the MPL fit over the training set. As shown in Figure 19, the resulting schedules exhibit a WSD-like shape, consisting of a stable phase and a decay phase, outperforming cosine schedules across sizes. For the 1B model, we derive a 72,000-step schedule based on the MPL fitted from 24,000-step constant and cosine schedule curves, with results in Figure 6(b) confirming superiority over cosine

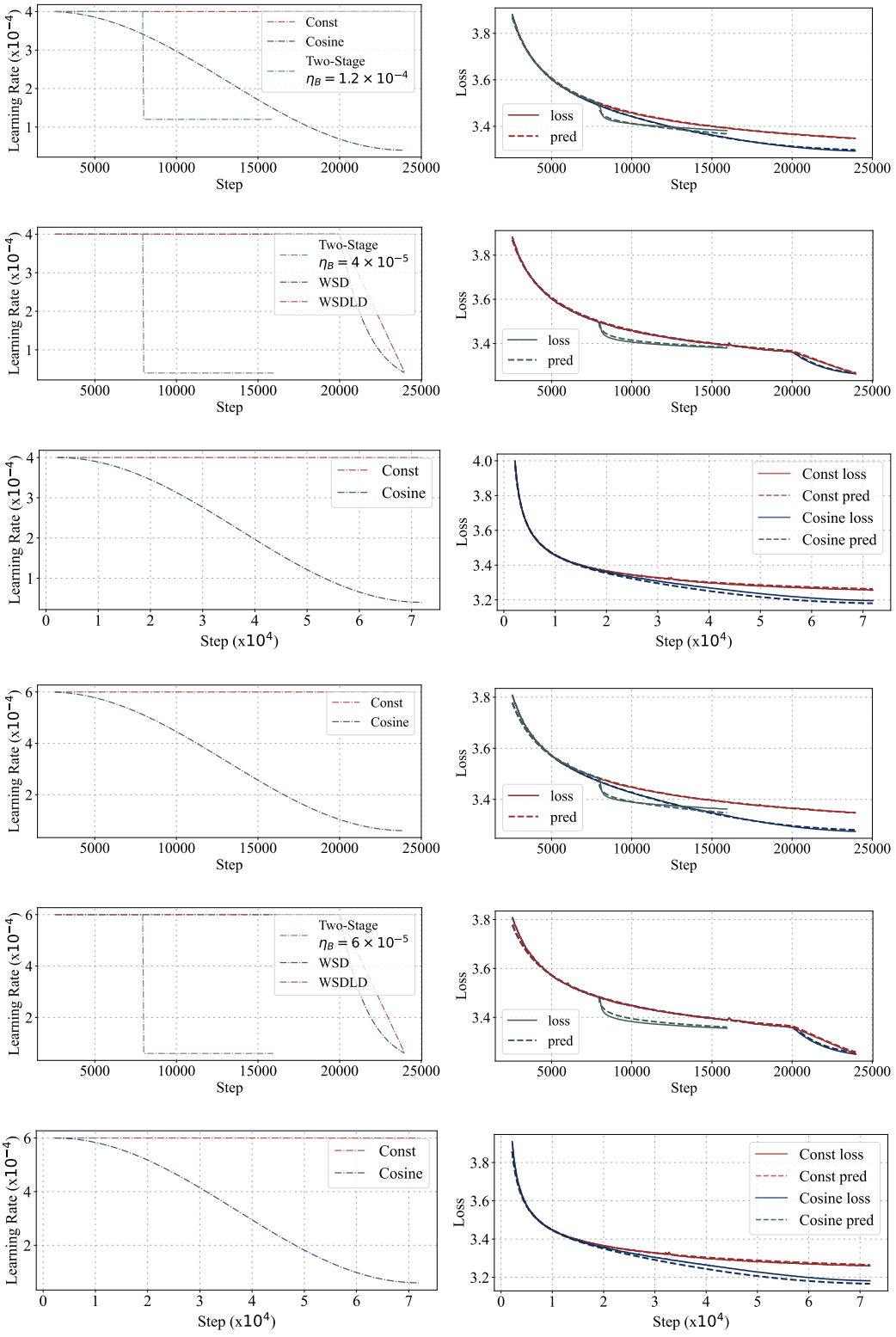



Figure 17: Ablation study on peak learning rates. **Left:** Learning rate schedules; **Right:** Corresponding loss curves. **Layout:** The first three rows show the results for a peak LR of $4 \times 10^{-4}$ while the last three rows are for the peak LR of $6 \times 10^{-4}$. Within each set of the three rows, the first row shows the fitting on the training set, the second row displays the prediction over unseen schedules and the third row demonstrates the extrapolation capability on a long horizon loss curve.

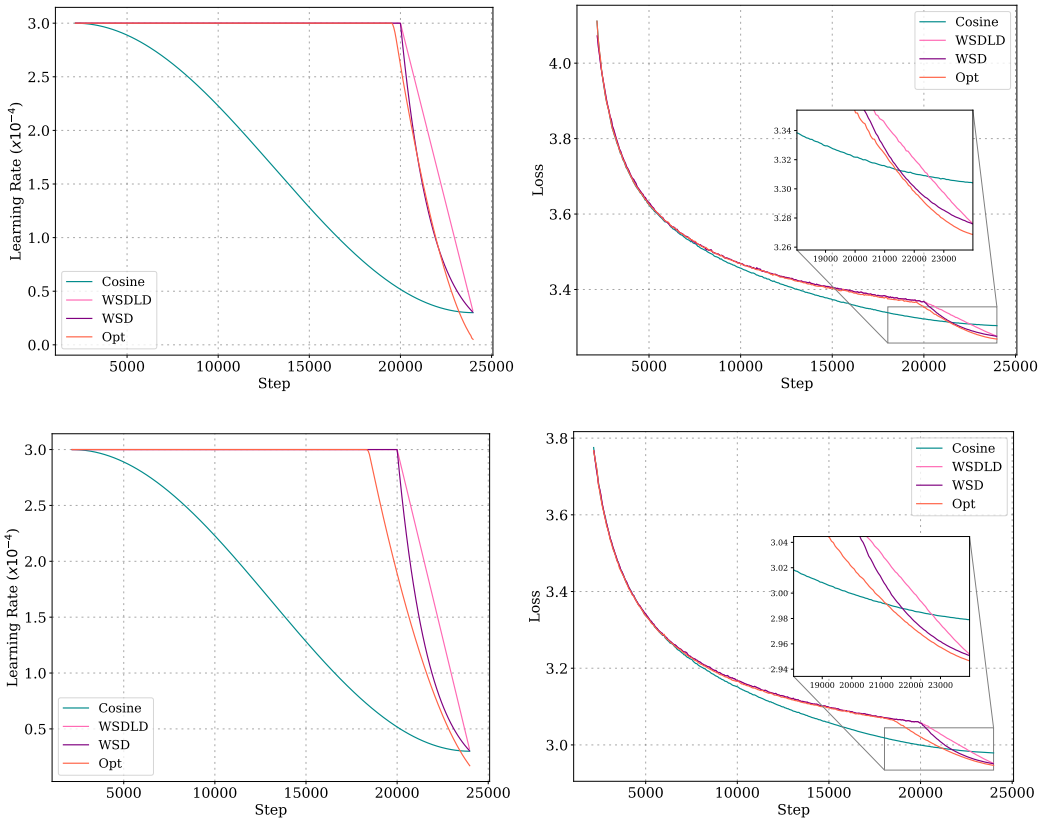



Figure 18: Comparison of our optimized LR schedules and their loss curves with cosine, WSD, and WSDLD schedules over 24,000 steps. The decay step for WSD and WSDLD is set to 4,000. **Upper:** 25M model; **Lower:** 100M model. **Left:** Learning rates over steps. **Right:** Losses over steps.

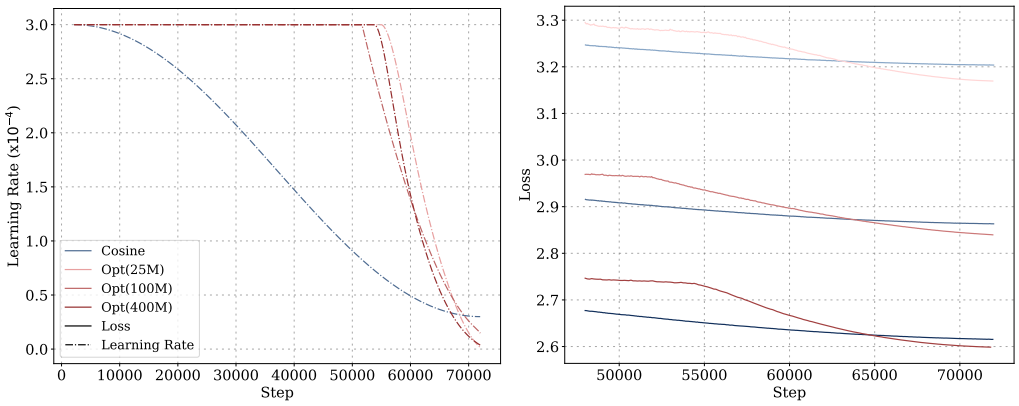



Figure 19: **Left:** Optimized and cosine LR schedules over 72,000 steps for models ranging from 25M to 400M. **Right:** Corresponding loss curves for optimized and cosine schedules.

schedule. Additionally, over 1B-model, we evaluate the downstream performance of the MPL-induced schedule against the cosine schedule on tasks including LAMBADA (Paperno et al., 2016), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), ARC-easy (Gu & Dao, 2023; Clark et al., 2018), $\mathbf{C}^3$ (Sun et al., 2020), and RTE (Wang et al., 2019). The MPL-induced schedule achieves an average score improvement of 1.03 compared to the cosine schedule, as shown in Table 2.

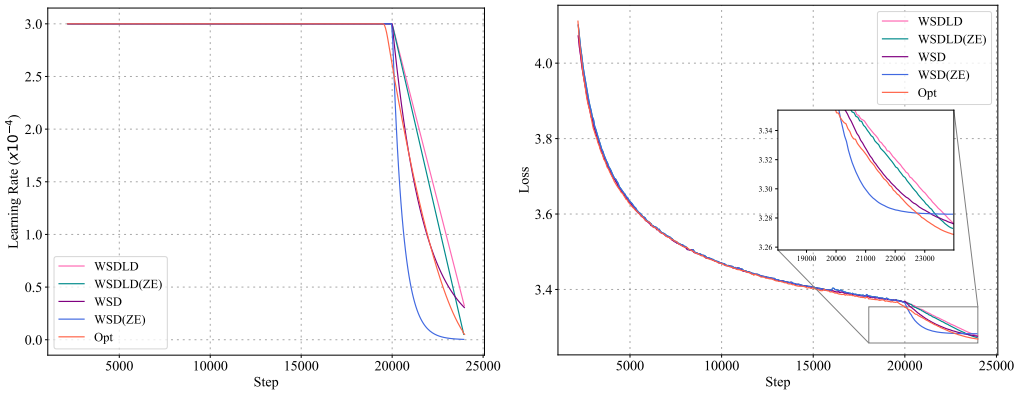



Figure 20: Comparison between the optimized schedules and WSD variants with a near-zero ending LR. WSD (ZE) and WSDLD (ZE) denote WSD and WSDLD schedules with an ending learning rate of $3 \times 10^{-7}$. **Left:** Learning rate comparison. **Right:** Loss comparison.

**Zero-Ending Learning Rate Experiments.** Optimized schedules consistently outperform WSD variants with near-zero learning rates ($3\times10^{-7}$, approximately 1/100 of the default setting). To test if a higher ending LR (e.g., $1/10$ peak LR) degrades baseline performance, we compare the optimized schedules against WSD(LD) variants with near-zero ending learning rates and the original ones. As shown in Figure 20, the optimized schedule still outperforms these WSD variant. In addition, lower ending learning rates do not consistently improve the final loss (e.g., zero-ending WSD exceeds baseline loss), suggesting a complex interaction between the ending learning rate and the decay function. This highlights the advantage of the optimized schedule in reducing the need for extensive hyperparameter tuning.

**WSD with Sqrt-Cube Decay (WSDSC).** We derive the decay function for optimized schedules by analyzing the decay phase across LLaMA2 models ranging from 25M to 400M. We compute normalized steps and learning rates (LRs) within the decay phase for schedules of varying step counts and model sizes. After averaging the normalized LRs, we perform symbolic regression against the normalized steps and approximate the decay function as $f(x) = (1-x)^{1.5}$. Validation experiments on 1B LLaMA2 and GPT models confirm its efficacy: Figure 6(b) shows that WSDSC outperforms the cosine schedule for the 1B model, though it falls short of the MPL-optimized schedule. Figure 15(a) demonstrates WSDSC's superiority over both the standard WSD and cosine schedules for GPT.

## J Optimal Learning Rate Schedule for Momentum Law

In this section, we derive the optimal learning rate schedules for the Momentum Law (Tissue et al., 2024):

$$L(T) = L_0 + A \cdot S_1^{-\alpha} - C \cdot S_2,$$

where $S_1 = \sum_{t=1}^{T} \eta_t$ and $S_2 = \sum_{t=1}^{T} \sum_{k=1}^{t} (\eta_{k-1} - \eta_k) \cdot \lambda^{t-k}$. $\lambda$ is a hyperparameter typically ranges from 0.99 to 0.999, and $L_0, A, C > 0$ are parameters.

Similar to Section 5, here we could also optimize this law to get a learning rate schedule achieving lowest final loss by solving

$$\min_{\eta_1,\eta_2,\ldots,\eta_T} \quad L_\Xi(\eta_1, \eta_2, \ldots, \eta_T) \tag{A}$$
$$s.t., \quad 0 \le \eta_t \le \eta_{t-1}, \ \forall 1 \le t \le T,$$

where $\Xi = \{L_0, A, C, \lambda\}$ represents the hyperparameters and parameters in $L(T)$. For simplicity of derivation, we introduce $\eta_0$ in front of $E$ as the maximal LR. Compared with Multi-Power Law (MPL), this optimization problem is obviously convex, so we could get its minimizer in theory. In our result, the Momentum Law would yield optimal schedules, which go through a stable phase at peak LR and then go to zero LR in no more than two steps. However, these kinds of schedules are

clearly far from the optimal schedules in practice. As a comparison, in Section 5, MPL can induce a WSD-like schedule, which is empirically effective. This shows the superiority of our MPL.

Next, we will formalize the above arguments mathematically.

**Theorem 2.** *For any LR schedule $E^* := \{\eta_0^*, \eta_1^*, \ldots, \eta_T^*\}$ that minimizes the optimization problem (A), there exists $k \in \{0, 1, \ldots, T\}$ such that the following holds for all $t \in \{1, \ldots, T\}$:*

1. *If $t \leq k$, then $\eta_t^* = \eta_0$;*
2. *If $t \geq k + 2$, then $\eta_t^* = 0$.*

For the convenience, we first prove the following lemma.

**Lemma 1.** *For a function $f(x) = x - M(1 - \lambda^x)$ with $M > 0$ and $0 < \lambda < 1$, we have the following properties:*

1. *$f(x)$ is strictly convex and has a unique minimizer over $x \in [0, \infty)$.*
2. *If $f(y) \geq 0$ for some $y \in [0, \infty)$, then $f(x) \geq f(y)$ for all $x \in [y, \infty)$.*

*Proof.* First, it is easy to check

$$f(0) = 0, \ \frac{df}{dx}(x) = 1 + M\lambda^x \log \lambda, \ \frac{d^2 f}{dx^2}(x) = M\lambda^x (\log \lambda)^2 > 0.$$

Then we can discuss the property of $f(x)$ over $x \in (0, \infty)$ by discussing $\frac{df}{dx}(0)$.

1. When $\frac{df}{dx}(0) \geq 0$, $\frac{df}{dx}(x) > \frac{df}{dx}(0) \geq 0$. Thus, $f(x)$ is monotonically increasing and $f(x) > f(0) = 0$.
2. When $\frac{df}{dx}(0) < 0$, then there exists $x^* \in (0, \infty)$ such that $\frac{df}{dx}(x^*) = 0$. Thus, $f(x)$ monotonically decreases over $(0, x^*)$ and monotonically increases over $(x^*, \infty)$. Hence $x^*$ is the only minimal over $(0, \infty)$. Moreover, $f(x^*) < f(0) = 0$ and $\lim_{x \to \infty} f(x) = \infty$, so there exists $\tilde{x} \in (x^*, \infty)$, such that $f(x) < 0$ over $(0, \tilde{x})$ and $f(x) > 0$ over $(\tilde{x}, \infty)$. Clearly, $f(x)$ monotonically increases over $x \in [\tilde{x}, \infty)$.

The above discussion completes the proof. □

Next, we prove Theorem 2.

*Proof for Theorem 2.* First, we reparameterize $\eta_t$ as $\eta_t = \eta_0 - \sum_{k=1}^t \Delta_k$, then the optimization problem (A) becomes

$$\begin{aligned} \min_{\Delta_1, \Delta_2, \ldots, \Delta_T} \quad & \hat{L}_\Xi(\Delta_1, \Delta_2, \ldots, \Delta_T) \\ \text{s.t.} \quad & \Delta_t \geq 0, \quad \forall 1 \leq t \leq T, \\ & \sum_{i=1}^T \Delta_i \leq \eta_0, \end{aligned}$$

where $\hat{L}_\Xi(\Delta_1, \Delta_2, \ldots, \Delta_T)$ is given by

$$\hat{L}_\Xi(\Delta_1, \Delta_2, \ldots, \Delta_T) = L_0 + A \cdot \left( T\eta_0 - \sum_{t=1}^T \sum_{k=1}^t \Delta_k \right)^{-\alpha} - C \cdot \sum_{t=1}^T \sum_{k=1}^t \Delta_k \lambda^{t-k}.$$

Define the Lagrangian by

$$\mathcal{L}(\Delta, \lambda, \mu) = \hat{L}_\Xi(\Delta_1, \ldots, \Delta_T) - \sum_{t=1}^T \lambda_t \Delta_t + \mu \left( \sum_{t=1}^T \Delta_t - \eta_0 \right),$$

where $\lambda_1, \ldots, \lambda_T$ and $\mu$ are the Lagrange multipliers associated with the constraints $\Delta_t \geq 0$ and $\sum_{i=1}^T \Delta_t \leq \eta_0$, respectively. By Karush-Kuhn-Tucker (KKT) conditions, there exist $\lambda_1, \ldots, \lambda_T \geq 0$ and $\mu \geq 0$ such that the following conditions hold:

- Complementary Slackness: $\lambda_t \Delta_t = 0$ for all $t = 1, \ldots, T$ and $\mu \left( \sum_{t=1}^{T} \Delta_t - \eta_0 \right) = 0$.
- Stationary: $\frac{\partial \hat{L}_\Xi}{\partial \Delta_t}(\Delta_1, \ldots, \Delta_T) - \lambda_t + \mu = 0$ for all $t = 1, \ldots, T$.

Here, we have

$$\begin{aligned}\frac{\partial \hat{L}_\Xi}{\partial \Delta_t} &= \alpha A \Phi^{-\alpha-1} \cdot (T - t + 1) - C \cdot (\lambda^0 + \lambda^1 + \cdots + \lambda^{T-t}) \\ &= \alpha A \Phi^{-\alpha-1} \cdot (T - t + 1) - C \cdot \frac{1 - \lambda^{T-t+1}}{1 - \lambda} \\ &= K f(T - t + 1),\end{aligned}$$

where $\Phi := T\eta_0 - \sum_{t=1}^{T} \sum_{k=1}^{t} \Delta_k$, $K := \alpha A \Phi^{-\alpha-1} > 0$, and $f(x) := x - M(1 - \lambda^x)$ with $M := \frac{C}{(1-\lambda)K} > 0$.

Note that $\Phi$ does not depend on $t$. We can rewrite the stationary condition as

$$\lambda_t = K f(T - t + 1) + \mu.$$

By Lemma 1, $f(x)$ is strictly convex and has a unique minimizer over $x \in [0, \infty)$. Let $x^*$ be this unique minimizer. Let $f_{\min} := \min_{t \in \{1,\ldots,T\}} \{f(T-t+1)\}$ be the minimum value of $f(T-t+1)$, and $S$ be the set of indices that minimize $f(T-t+1)$. Then $|S| \leq 2$ and $S \subseteq \{\lfloor T - x^* + 1 \rfloor, \lceil T - x^* + 1 \rceil\}$.

Now we discuss the following two cases by the value of $K f_{\min} + \mu$.

**Case 1.** If $K f_{\min} + \mu > 0$, then $\lambda_t > 0$ for all $t = 1, \ldots, T$. By the complementary slackness condition, $\Delta_t = 0$ for all $t = 1, \ldots, T$. This implies that $E^*$ is a constant schedule, $\eta_0 = \eta_1^* = \eta_2^* = \cdots = \eta_T^*$.

**Case 2.** If $K f_{\min} + \mu = 0$, then $\lambda_t = 0$ for all $t \in S$ and $\lambda_t > 0$ for all $t \notin S$. By the complementary slackness condition, the latter implies that $\Delta_t = 0$ for all $t \notin S$. Then $E^*$ falls into one of the two categories:

1. If $S = \{s\}$ for some $s$, then $\eta_0 = \eta_1^* = \eta_2^* = \cdots = \eta_{s-1}^*$ and $\eta_s^* = \eta_{s+1}^* = \cdots = \eta_T^*$;
2. If $S = \{s-1, s\}$ for some $s$, then $\eta_0 = \eta_1^* = \eta_2^* = \cdots = \eta_{s-2}^*$ and $\eta_s^* = \eta_{s+1}^* = \cdots = \eta_T^*$.

We claim that $\eta_T^* = 0$ if $s < T$. If not, then $\mu = 0$ must hold by the complementary slackness condition. Moreover, $s < T$ implies $T \notin S$, and then we have $0 < \lambda_T = K f(1) + \mu = K f(1)$. By Lemma 1, $f(x) \geq f(1) > 0$ for all $x \geq 1$, which implies that $\lambda_t = K f(T - t + 1) + \mu = K f(T - t + 1) > 0$ for all $t = 1, \ldots, T$, which contradicts the fact that $\lambda_t = 0$ for all $t \in S$.

Putting all these together, we conclude that $E^*$ must exhibit the pattern described in the theorem. □

## K PROOF OF THEOREM 1

The proof of Theorem 1 consists of two main parts. In the first part, we derive the last iterate loss. To prove Theorem 1, we first treat all $\lambda_i$ and $\Sigma_{ii}$ as fixed constants (not random variables as stated in Assumption 1), and we give a theorem in this scenario.

**Theorem 3.** *For $\boldsymbol{\theta}_T \sim \Phi(\boldsymbol{\theta}_0, E)$, we have the following estimate of $\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_T)]$:*

$$M(\boldsymbol{\theta}_0, E) := \frac{1}{2}\sum_{i=1}^{d}\left(\theta_{0,i}^2\lambda_i\exp(-2\lambda_i S_1) + \eta_1\Sigma_{ii}\cdot\frac{1-\exp(-2\lambda_i S_1)}{2}\right) - \frac{1}{2}\sum_{k=2}^{T}(\eta_{k-1}-\eta_k)\sum_{i=1}^{d}\frac{1-\exp(-2\lambda_i S_k)}{2}\Sigma_{ii},$$

*where $S_k := \sum_{\tau=k}^{T}\eta_\tau$, and the estimation error is bounded as*

$$|\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_T)] - M(\boldsymbol{\theta}_0, E)| \le 5\eta_{\max}\sum_{i=1}^{d}\lambda_i^3 S_1\exp(-2\lambda_i S_1)\theta_{0,i}^2 + 5\exp(2)\eta_{\max}^2\sum_{i=1}^{d}\Sigma_{ii}\lambda_i.$$

To prove the theorem, we first introduce some notations and auxiliary expectations. WLOG, we assume that $\boldsymbol{H} = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$, and set $\boldsymbol{\theta}_* = 0$. And we define that

$$U(\boldsymbol{\theta}, \eta, S) := \frac{1}{2}\sum_{i=1}^{d}\left(\theta_i^2\lambda_i\exp(-2\lambda_i S) + \eta\Sigma_{ii}\cdot\frac{1-\exp(-2\lambda_i S)}{2}\right).$$

We decompose the expected loss $\mathbb{E}_{\boldsymbol{\theta}_T\sim\Phi(\boldsymbol{\theta}_0,E)}[\mathcal{L}(\boldsymbol{\theta}_T)]$ into a telescoping sum of $T+1$ auxiliary expectations $A_0, A_1, \ldots, A_T$:

$$\mathbb{E}_{\boldsymbol{\theta}_T\sim\Phi(\boldsymbol{\theta}_0,E)}[\mathcal{L}(\boldsymbol{\theta}_T)] = A_0 + \sum_{k=1}^{T}(A_k - A_{k-1})$$
$$A_k := \mathbb{E}_{\boldsymbol{\theta}_k\sim\Phi(\boldsymbol{\theta}_0, E_{\le k})}[U(\boldsymbol{\theta}_k, \eta_k, S_{k+1})], \tag{B}$$

Here we define $\eta_0 = \eta_1$ for convenience. Also we define $S_{T+1} = 0$, so $A_T = \mathbb{E}_{\boldsymbol{\theta}_T\sim\Phi(\boldsymbol{\theta}_0,E)}[\mathcal{L}(\boldsymbol{\theta}_T)]$.

The above theorem needs the following two lemmas.

**Lemma 2.** *If $x \in [0,1]$, then*

$$\exists\xi_1 \in [0,10] \quad \textit{s.t.} \quad (1-x)^2 = \exp(-2x)(1+\xi_1 x^2),$$
$$\exists\xi_2 \in [0,10] \quad \textit{s.t.} \quad (1-2x) = \exp(-2x)(1+\xi_2 x^2).$$

*Proof.* The above inequalities hold for $x = 0$. For $x \in (0,1]$, we have

$$\frac{1-(1-2x)\exp(2x)}{x^2} \ge \frac{1-(1-x)^2\exp(2x)}{x^2} \ge \frac{1-\exp(-2x)\exp(2x)}{x^2} = 0,$$

where we use the fact that $1-2x \le (1-x)^2 \le \exp(-2x)$. Also note that $\frac{1-(1-2x)\exp(2x)}{x^2}$ is an increasing function of $x$. So we have $\frac{1-(1-2x)\exp(2x)}{x^2} \le \frac{1-(-1)\cdot\exp(2)}{1^2} \le 10$. □

**Lemma 3.** *If $\eta_{\max} \le \frac{1}{\lambda_{\max}}$, then for all $k \in [T]$,*

$$\sum_{t=1}^{k-1}\eta_t\exp(-2\lambda_i S_t) \le \frac{1}{2\lambda_i}\exp(-2\lambda_i S_k) \le \sum_{t=1}^{k-1}\eta_t\exp(-2\lambda_i S_{t+1}) \le \frac{\exp(2)}{2\lambda_i}\exp(-2\lambda_i S_k).$$

*Proof.* The first inequality follows the fact that lower Darboux sum is smaller than the Darboux integral

$$\sum_{t=1}^{k-1}\eta_t\exp(-2\lambda_i S_t) = \sum_{t=1}^{k-1}(S_t - S_{t+1})\exp(-2\lambda_i S_t) \le \int_{S_k}^{S_1}\exp(-2\lambda_i S)\mathrm{d}S$$
$$= \frac{1}{2\lambda_i}\left[\exp(-2\lambda_i S_k) - \exp(-2\lambda_i S_1)\right]$$
$$\le \frac{1}{2\lambda_i}\exp(-2\lambda_i S_k).$$

Similarly, the upper Darboux sum's property induces the second inequality. Also, we have

$$\sum_{t=1}^{k-1}\eta_t\exp(-2\lambda_i S_{t+1}) = \sum_{t=1}^{k-1}\eta_t\exp(-2\lambda_i S_t)\exp(2\lambda_i\eta_t) \le \sum_{t=1}^{k-1}\eta_t\exp(-2\lambda_i S_t)\exp(2)$$
$$\le \frac{\exp(2)}{2\lambda_i}\exp(-2\lambda_i S_k),$$

which completes the proof. □

The following lemma characterizes the difference between two consecutive auxiliary expectations $A_k$ and $A_{k-1}$.

**Lemma 4.** *If $\eta_{\max} \le \frac{1}{\lambda_{\max}}$, then for all $k \in [T]$,*

$$A_k - A_{k-1} = -\frac{1}{2}(\eta_{k-1}-\eta_k)\sum_{i=1}^{d}\frac{1-\exp(-2\lambda_i S_k)}{2}\Sigma_{ii} + \epsilon_k,$$

*where the error term $\epsilon_k$ is bounded by*

$$|\epsilon_k| \le 5\sum_{i=1}^{d}\eta_k^2\lambda_i^3\exp(-2\lambda_i S_k)\mathbb{E}_{\boldsymbol{\theta}_{k-1}\sim\Phi(\boldsymbol{\theta}_0,E_{\le k-1})}[\theta_{k-1,i}^2] + 5\sum_{i=1}^{d}\eta_k^3\Sigma_{ii}\lambda_i^2\exp(-2\lambda_i S_k).$$

*Proof.* By the definition of $A_k$ and $A_{k-1}$, we have

$$\begin{aligned}A_k - A_{k-1} &= \mathbb{E}_{\boldsymbol{\theta}_k\sim\Phi(\boldsymbol{\theta}_0,E_{\le k})}[U(\boldsymbol{\theta}_k,\eta_k,S_{k+1})] - \mathbb{E}_{\boldsymbol{\theta}_{k-1}\sim\Phi(\boldsymbol{\theta}_0,E_{\le k-1})}[U(\boldsymbol{\theta}_{k-1},\eta_{k-1},S_k)]\\ &= \mathbb{E}_{\boldsymbol{\theta}_{k-1}\sim\Phi(\boldsymbol{\theta}_0,E_{\le k-1})}\\ &\qquad\Bigg[\underbrace{\mathbb{E}_{\boldsymbol{g}_k\sim\mathcal{N}(\boldsymbol{H}\boldsymbol{\theta}_{k-1},\boldsymbol{\Sigma})}[U(\boldsymbol{\theta}_{k-1}-\eta_k\boldsymbol{g}_k,\eta_k,S_{k+1})\mid\boldsymbol{\theta}_{k-1}]}_{=:\,\bar{U}(\boldsymbol{\theta}_{k-1})} - U(\boldsymbol{\theta}_{k-1},\eta_{k-1},S_k)\Bigg].\end{aligned}$$

We expand $\bar{U}(\boldsymbol{\theta}_{k-1}) := \mathbb{E}_{\boldsymbol{g}_k\sim\mathcal{N}(\boldsymbol{H}\boldsymbol{\theta}_{k-1},\boldsymbol{\Sigma})}[U(\boldsymbol{\theta}_{k-1}-\eta_k\boldsymbol{g}_k,\eta_k,S_{k+1})\mid\boldsymbol{\theta}_{k-1}]$ based on the definition of $U$:

$$\bar{U}(\boldsymbol{\theta}_{k-1}) = \underbrace{\mathbb{E}_{\boldsymbol{g}_k\sim\mathcal{N}(\boldsymbol{H}\boldsymbol{\theta}_{k-1},\boldsymbol{\Sigma})}\left[\frac{1}{2}\sum_{i=1}^{d}(\theta_{k-1,i}-\eta_k g_{k,i})^2\lambda_i\exp(-2\lambda_i S_{k+1})\,\middle|\,\boldsymbol{\theta}_{k-1}\right]}_{=:\bar{U}_1(\boldsymbol{\theta}_{k-1})}$$
$$+\underbrace{\frac{1}{2}\sum_{i=1}^{d}\eta_k\Sigma_{ii}\cdot\frac{1-\exp(-2\lambda_i S_{k+1})}{2}}_{=:\bar{U}_2(\boldsymbol{\theta}_{k-1})}.$$

We can simplify $\bar{U}_1(\boldsymbol{\theta}_{k-1})$ as

$$\begin{aligned}\bar{U}_1(\boldsymbol{\theta}_{k-1}) &= \frac{1}{2}\sum_{i=1}^{d}\left(\lambda_i\exp(-2\lambda_i S_{k+1})\left((1-\eta_k\lambda_i)^2\theta_{k-1,i}^2+\eta_k^2\Sigma_{ii}\right)\right),\\ &= \underbrace{\frac{1}{2}\sum_{i=1}^{d}\lambda_i\exp(-2\lambda_i S_{k+1})(1-\eta_k\lambda_i)^2\theta_{k-1,i}^2}_{=:\bar{U}_{11}(\boldsymbol{\theta}_{k-1})} + \underbrace{\frac{1}{2}\sum_{i=1}^{d}\lambda_i\exp(-2\lambda_i S_{k+1})\eta_k^2\Sigma_{ii}}_{=:\bar{U}_{12}(\boldsymbol{\theta}_{k-1})}.\end{aligned}$$

Let $\bar{U}_3(\boldsymbol{\theta}_{k-1}) := \bar{U}_{12}(\boldsymbol{\theta}_{k-1}) + \bar{U}_2(\boldsymbol{\theta}_{k-1})$. Then $\bar{U}(\boldsymbol{\theta}_{k-1}) = \bar{U}_{11}(\boldsymbol{\theta}_{k-1}) + \bar{U}_3(\boldsymbol{\theta}_{k-1})$. We can rewrite $\bar{U}_3(\boldsymbol{\theta}_{k-1})$ as

$$\bar{U}_3(\boldsymbol{\theta}_{k-1}) = \frac{1}{2}\sum_{i=1}^{d}\left(\eta_k\Sigma_{ii}\cdot\frac{1-\exp(-2\lambda_i S_k)(1-2\eta_k\lambda_i)}{2}\right).$$

Since $\eta_k\lambda_i \in [0,1]$ for all $i$, by Lemma 2, we can find $\xi_{1,i},\xi_{2,i}\in[0,10]$ such that

$$(1-\eta_k\lambda_i)^2 = \exp(-2\eta_k\lambda_i)(1+\xi_{1,i}\eta_k^2\lambda_i^2),\quad (1-2\eta_k\lambda_i) = \exp(-2\eta_k\lambda_i)(1+\xi_{2,i}\eta_k^2\lambda_i^2).$$

Then we can rewrite $\bar{U}_{11}(\boldsymbol{\theta}_{k-1})$ as

$$\begin{aligned}\bar{U}_{11}(\boldsymbol{\theta}_{k-1}) &= \frac{1}{2}\sum_{i=1}^{d}(1+\xi_{1,i}\eta_k^2\lambda_i^2)\lambda_i \exp(-2\lambda_i S_k)\theta_{k-1,i}^2 \\ &= \frac{1}{2}\sum_{i=1}^{d}\lambda_i \exp(-2\lambda_i S_k)\theta_{k-1,i}^2 + \frac{1}{2}\sum_{i=1}^{d}\xi_{1,i}\eta_k^2\lambda_i^3 \exp(-2\lambda_i S_k)\theta_{k-1,i}^2.\end{aligned}$$

Similarly, we can rewrite $\bar{U}_3(\boldsymbol{\theta}_{k-1})$ as

$$\begin{aligned}\bar{U}_3(\boldsymbol{\theta}_{k-1}) &= \frac{1}{2}\sum_{i=1}^{d}\left(\eta_k\Sigma_{ii}\cdot\frac{1-(1+\xi_{2,i}\eta_k^2\lambda_i^2)\exp(-2\lambda_i S_k)}{2}\right) \\ &= \frac{1}{2}\sum_{i=1}^{d}\left(\eta_k\Sigma_{ii}\cdot\frac{1-\exp(-2\lambda_i S_k)}{2}\right) - \frac{1}{2}\sum_{i=1}^{d}\xi_{2,i}\eta_k^3\Sigma_{ii}\lambda_i^2\exp(-2\lambda_i S_k).\end{aligned}$$

Therefore, we can rewrite $\bar{U}(\boldsymbol{\theta}_{k-1})$ as

$$\begin{aligned}\bar{U}(\boldsymbol{\theta}_{k-1}) &= \frac{1}{2}\sum_{i=1}^{d}\left(\lambda_i\exp(-2\lambda_i S_k)\theta_{k-1,i}^2 + \eta_k\Sigma_{ii}\cdot\frac{1-\exp(-2\lambda_i S_k)}{2}\right) \\ &\quad + \frac{1}{2}\sum_{i=1}^{d}\xi_{1,i}\eta_k^2\lambda_i^3\exp(-2\lambda_i S_k)\theta_{k-1,i}^2 - \frac{1}{2}\sum_{i=1}^{d}\xi_{2,i}\eta_k^3\Sigma_{ii}\lambda_i^2\exp(-2\lambda_i S_k).\end{aligned}$$

Subtracting $U(\boldsymbol{\theta}_{k-1}, \eta_{k-1}, S_k)$ from the above expression, we have

$$\begin{aligned}\Delta\bar{U}(\boldsymbol{\theta}_{k-1}) &:= \bar{U}(\boldsymbol{\theta}_{k-1}) - U(\boldsymbol{\theta}_{k-1}, \eta_{k-1}, S_{k-1}) \\ &= -\frac{1}{2}\sum_{i=1}^{d}(\eta_{k-1}-\eta_k)\sum_{i=1}^{d}\frac{1-\exp(-2\lambda_i S_k)}{2}\Sigma_{ii} \\ &\quad + \frac{1}{2}\sum_{i=1}^{d}\xi_{1,i}\eta_k^2\lambda_i^3\exp(-2\lambda_i S_k)\theta_{k-1,i}^2 - \frac{1}{2}\sum_{i=1}^{d}\xi_{2,i}\eta_k^3\Sigma_{ii}\lambda_i^2\exp(-2\lambda_i S_k).\end{aligned}$$

Taking the expectation of $\Delta\bar{U}(\boldsymbol{\theta}_{k-1})$ over $\boldsymbol{\theta}_{k-1} \sim \Phi(\boldsymbol{\theta}_0, E_{\le k-1})$ proves the lemma. □

The following lemma gives an upper bound for $\mathbb{E}_{\boldsymbol{\theta}_{k-1}\sim\Phi(\boldsymbol{\theta}_0, E_{\le k-1})}[\theta_{k-1,i}^2]$.

**Lemma 5.** *If $\eta_{\max} \le \frac{1}{\lambda_{\max}}$, then for all $k \in [T]$ and $i \in [d]$,*

$$\mathbb{E}_{\boldsymbol{\theta}_{k-1}\sim\Phi(\boldsymbol{\theta}_0, E_{\le k-1})}[\theta_{k-1,i}^2] \le \theta_{0,i}^2\exp(-2\lambda_i(S_1 - S_k)) + \frac{\exp(2)}{\lambda_i}\eta_{\max}\Sigma_{ii}.$$

*Proof.* By the update rule, we have

$$\mathbb{E}[\theta_{t,i}^2] = (1-\eta_t\lambda_i)^2\mathbb{E}[\theta_{t-1,i}^2] + \eta_t^2\Sigma_{ii}.$$

Since $(1-\eta_t\lambda_i)^2 \le \exp(-2\eta_t\lambda_i)$ and $\eta_t \le \eta_{\max}$, we have the following bound:

$$\mathbb{E}[\theta_{t,i}^2] \le \exp(-2\eta_t\lambda_i)\mathbb{E}[\theta_{t-1,i}^2] + \eta_t\eta_{\max}\Sigma_{ii}.$$

Expanding the recursion, we have

$$\begin{aligned}\mathbb{E}[\theta_{k-1,i}^2] &\le \theta_{0,i}^2\exp(-2\lambda_i(S_1-S_k)) + \sum_{t=1}^{k-1}\eta_t\eta_{\max}\Sigma_{ii}\exp(-2\lambda_i(S_{t+1}-S_k)) \\ &= \theta_{0,i}^2\exp(-2\lambda_i(S_1-S_k)) + \exp(2\lambda_i S_k)\eta_{\max}\Sigma_{ii}\sum_{t=1}^{k-1}\eta_t\exp(-2\lambda_i S_{t+1}) \\ &\le \theta_{0,i}^2\exp(-2\lambda_i(S_1-S_k)) + \exp(2\lambda_i S_k)\eta_{\max}\Sigma_{ii}\cdot\frac{1}{\lambda_i}\exp(-2\lambda_i S_k) \\ &= \theta_{0,i}^2\exp(-2\lambda_i(S_1-S_k)) + \frac{\exp(2)}{\lambda_i}\eta_{\max}\Sigma_{ii},\end{aligned}$$

where the first inequality uses the fact that $\prod_{\tau=t+1}^{k-1}\exp(-2\eta_\tau\lambda_i) = \exp(-2\lambda_i(S_{t+1}-S_k))$ and the second inequality uses Lemma 3. □

**Lemma 6.** *In the setting of Lemma 4, we can bound the sum of the error terms $\epsilon_k$ as*

$$\left|\sum_{k=1}^{T}\epsilon_k\right| \le 5\eta_{\max}\sum_{i=1}^{d}\lambda_i^3 S_1\exp(-2\lambda_i S_1)\theta_{0,i}^2 + 5\exp(2)\eta_{\max}^2\sum_{i=1}^{d}\Sigma_{ii}\lambda_i.$$

*Proof.* By the upper bound of $|\epsilon_k|$,

$$\begin{aligned}\left|\sum_{k=1}^{T}\epsilon_k\right| &\le \sum_{k=1}^{T}|\epsilon_k| \\ &\le 5\sum_{i=1}^{d}\Big(\underbrace{\sum_{k=1}^{T}\eta_k^2\lambda_i^3\exp(-2\lambda_i S_k)\mathbb{E}_{\boldsymbol{\theta}_{k-1}\sim\Phi(\boldsymbol{\theta}_0,E_{\le k-1})}[\theta_{k-1,i}^2]}_{=:\,\mathcal{E}_{1,i}} + \underbrace{\sum_{k=1}^{T}\eta_k^3\Sigma_{ii}\lambda_i^2\exp(-2\lambda_i S_k)}_{=:\,\mathcal{E}_{2,i}}\Big).\end{aligned}$$

For $\mathcal{E}_{1,i}$, we apply Lemma 5 and have

$$\begin{aligned}\mathcal{E}_{1,i} &\le \sum_{k=1}^{T}\eta_k^2\lambda_i^3\exp(-2\lambda_i S_k)\left(\theta_{0,i}^2\exp(-2\lambda_i(S_1-S_k)) + \frac{\exp(2)}{\lambda_i}\eta_{\max}\Sigma_{ii}\right) \\ &= \sum_{k=1}^{T}\eta_k^2\lambda_i^3\exp(-2\lambda_i S_1)\theta_{0,i}^2 + \sum_{k=1}^{T}\exp(2)\eta_k^2\lambda_i^2\eta_{\max}\Sigma_{ii}\exp(-2\lambda_i S_k) \\ &\le \eta_{\max}\lambda_i^3 S_1\exp(-2\lambda_i S_1)\theta_{0,i}^2 + \eta_{\max}^2\frac{\exp(2)}{2}\Sigma_{ii}\lambda_i,\end{aligned}$$

where the last inequality uses Lemma 3. For $\mathcal{E}_{2,i}$, we have

$$\begin{aligned}\mathcal{E}_{2,i} &= \sum_{k=1}^{T}\eta_k^3\Sigma_{ii}\lambda_i^2\exp(-2\lambda_i S_k) \\ &\le \eta_{\max}^2\sum_{k=1}^{T}\eta_k\Sigma_{ii}\lambda_i^2\exp(-2\lambda_i S_k) \\ &\le \eta_{\max}^2\frac{1}{2\lambda_i}\Sigma_{ii}\lambda_i^2\end{aligned}$$

Adding the upper bounds of $\mathcal{E}_{1,i}$ and $\mathcal{E}_{2,i}$ together completes the proof of Lemma 6. □

Now we are ready to prove Theorem 3.

*Proof for Theorem 3.* Using Lemma 4 and Lemma 6, we have that

$$\sum_{k=1}^{T}A_k - A_{k-1} = -\frac{1}{2}\sum_{k=1}^{T}(\eta_{k-1}-\eta_k)\sum_{i=1}^{d}\frac{1-\exp(-2\lambda_i S_k)}{2}\Sigma_{ii} + \epsilon,$$

where the error bound $\epsilon$ can be bounded as

$$\epsilon \le 5\eta_{\max}\sum_{i=1}^{d}\lambda_i^3 S_1\exp(-2\lambda_i S_1)\theta_{0,i}^2 + 5\exp(2)\eta_{\max}^2\sum_{i=1}^{d}\Sigma_{ii}\lambda_i.$$

According to (B), we have

$$\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_T)] = A_0 + \sum_{k=1}^{T}(A_k - A_{k-1}).$$

Plugging in the expression of each $A_k$, we get the results in Theorem 3. □

Next, we prove Theorem 1 as follows.

*Proof for Theorem 1.* First, we recap some definitions used in the following calculation. The expectation of $\Sigma_{ii}$ for any $i \in \{1,\dots,d\}$ is denoted by $\mathbb{E}[\Sigma] = \mu$ from Assumption 1. Also, in Assumption 1, we have that $\mathbb{E}[\Sigma|\lambda] \propto \lambda^\rho\exp(-G\lambda)$. Here for the alignment of scale and making the derivation neat, we let $\mathbb{E}[\Sigma|\lambda] = F\mu\lambda^\rho\exp(-G\lambda)$, where $F$ is a universal constant and $\mu$ is the expectation of $\Sigma$'s marginal distribution.

We take the expectation of $M(\boldsymbol{\theta}, E)$ over all $\lambda_i$ and $\Sigma_{ii}$ as a direct result of integral

$$\mathbb{E}[M(\boldsymbol{\theta}_0, E)] = \underbrace{\frac{1}{2}\|\boldsymbol{\theta}_0\|_2^2 \mathbb{E}[\lambda \exp(-2\lambda S_1)]}_{:=I_1} + \underbrace{\frac{d}{4}\eta_{\max}\mathbb{E}[\Sigma] - \frac{d}{4}\eta_{\max}\mathbb{E}[\Sigma \exp(-2\lambda S_1)]}_{:=I_2}$$
$$\underbrace{- \frac{d}{4}\sum_{k=2}^{T}(\eta_{k-1} - \eta_k)\left(\mathbb{E}[\Sigma] - \mathbb{E}[\Sigma \exp(-2\lambda S_k)]\right)}_{:=I_3}$$

Separately, we have that

$$\begin{aligned} I_1 &= \frac{1}{2}\|\boldsymbol{\theta}_0\|_2^2 \mu \frac{1}{Z_\lambda}\int_0^D \lambda^{\alpha+1}\exp(-2\lambda S_1)\mathrm{d}\lambda \\ &= \frac{\|\boldsymbol{\theta}_0\|_2^2 \mu \gamma(\alpha+2, D)}{2^{\alpha+3} Z_\lambda} S_1^{-\alpha-2}, \end{aligned}$$

and that

$$\begin{aligned} I_2 &= \frac{d}{4}\eta_{\max}\mu - \frac{d}{4}\eta_{\max}\mu F \frac{1}{Z_\lambda}\int_0^D \lambda^{\alpha+\rho}\exp(-(2S_1+G)\lambda)\mathrm{d}\lambda \\ &= \frac{d}{4}\eta_{\max}\mu - \frac{d\eta_{\max}\mu F \gamma(\alpha+\rho+1, D)}{2^{\alpha+\rho+3} Z_\lambda}(S_1 + \frac{G}{2})^{-\alpha-\rho-1}, \end{aligned}$$

and that

$$\begin{aligned} I_3 &= \frac{d\mu}{4}\sum_{k=2}^{T}(\eta_{k-1}-\eta_k)(1 - F\frac{1}{Z_\lambda}\int_0^D \lambda^{\alpha+\rho}\exp(-(2S_k+G)\lambda))\mathrm{d}\lambda \\ &= \frac{d\mu}{4}\sum_{k=2}^{T}(\eta_{k-1}-\eta_k)\left(1 - \frac{F\gamma(\alpha+\rho+1, D)}{G^{\alpha+\rho+1}Z_\lambda}(\frac{2}{G}S_k + 1)^{-\alpha-\rho-1}\right). \end{aligned}$$

Putting $I_1, I_2$ and $I_3$ all together then we have that

$$\begin{aligned} \mathbb{E}[M(\boldsymbol{\theta}_0, E)] = &\frac{\|\boldsymbol{\theta}_0\|_2^2 \mu \gamma(\alpha+2, D)}{2^{\alpha+3} Z_\lambda} S_1^{-\alpha-2} \\ &+ \frac{d}{4}\eta_{\max}\mu - \frac{d\eta_{\max}\mu F \gamma(\alpha+\rho+1, D)}{2^{\alpha+\rho+3} Z_\lambda}(S_1 + \frac{G}{2})^{-\alpha-\rho-1} \\ &- \frac{d\mu}{4}\sum_{k=2}^{T}(\eta_{k-1}-\eta_k)\left(1 - \frac{F\gamma(\alpha+\rho+1, D)}{G^{\alpha+\rho+1}Z_\lambda}(\frac{2}{G}S_k + 1)^{-\alpha-\rho-1}\right). \end{aligned}$$

where $\gamma(\cdot,\cdot)$ denote the lower incomplete gamma function such that $\gamma(s,x) := \int_0^x t^{s-1}e^{-t}\mathrm{d}t$, $Z_\lambda$ denote the partition function such that $Z_\lambda := \int_0^D \boldsymbol{p}(\lambda)\mathrm{d}\lambda$. The second equality uses Assumption 1, and the last equality uses the property of Laplace Transform. To make the expression clear, we let $F = \frac{G^{\alpha+\rho+1}Z_\lambda}{\gamma(\alpha+\rho+1, D)}$, and we define the following parameters $L_0, A, B, C, R$ as

$$\begin{aligned} L_0 &:= \frac{d}{4}\eta_{\max}\mu, \\ A &:= \frac{\|\boldsymbol{\theta}_0\|_2^2 \mu \gamma(\alpha+2, D)}{2^{\alpha+3} Z_\lambda}, \\ B &:= \frac{d\mu}{4}, \\ C &:= \frac{2}{G}, \\ R &:= \frac{d\eta_{\max}\mu F \gamma(\alpha+\rho+1, D)}{2^{\alpha+\rho+3} Z_\lambda}. \end{aligned}$$

So we get that

$$\tilde{M}(\boldsymbol{\theta}_0, E) := \mathbb{E}[M(\boldsymbol{\theta}_0, E)]$$

$$= L_0 + AS_1^{-\alpha-2} - R(S_1 + \frac{1}{C})^{-\alpha\rho-1} - \sum_{k=2}^{T} B(\eta_{k-1} - \eta_k)\left(1 - (CS_k + 1)^{-\alpha-\beta-1}\right).$$

Also, we take the expectation of the error bound as

$$\begin{aligned}
\left|\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_T)] - \tilde{M}(\boldsymbol{\theta}_0, E)\right| \leq & \mathbb{E}[5\eta_{\max}\sum_{i=1}^{d}\lambda_i^3 S_1 \exp(-2\lambda_i S_1)\theta_{0,i}^2] + \mathbb{E}[5\exp(2)\eta_{\max}^2\sum_{i=1}^{d}\Sigma_{ii}\lambda_i] \\
= & 5\eta_{\max}\|\boldsymbol{\theta}_0\|_2^2\mathbb{E}[\lambda^3 S_1\exp(-2\lambda S_1)] + 5\exp(2)\eta_{\max}^2 d\mathbb{E}[\Sigma\lambda] \\
= & 5\eta_{\max}\|\boldsymbol{\theta}_0\|_2^2\frac{1}{Z_\lambda}\int_0^D \lambda^{3+\alpha}S_1\exp(-2\lambda S_1)\mathrm{d}\lambda \\
& + 5\exp(2)\eta_{\max}^2 d\frac{1}{Z_\lambda}\mu F\int_0^D \lambda^{1+\rho}\exp(-2G\lambda)\mathrm{d}\lambda \\
= & \frac{5\eta_{\max}\|\boldsymbol{\theta}_0\|_2^2\gamma(4+\alpha, D)}{2^{4+\alpha}Z_\lambda}S_1^{-\alpha-3} + \frac{5\exp(2)\eta_{\max}^2 d\mu F\gamma(2+\rho, D)}{(2G)^{\rho+2}Z_\lambda} \\
= & O(S_1^{-\alpha-3}) + O(\eta_{\max}^2)
\end{aligned}$$

Notice that, not only in the case of $T$ iterations, the results above holds for all $0 \leq t \leq T$, with the next variable replacement

$$\begin{aligned}
&\tilde{M}(\boldsymbol{\theta}_0, E) \leftarrow \tilde{M}_t(\boldsymbol{\theta}_0, E), \\
&\mathcal{L}(\boldsymbol{\theta}_T) \leftarrow \mathcal{L}(\boldsymbol{\theta}_t), \\
&S_i \leftarrow S_i(t).
\end{aligned}$$

If $S_1(t) > \frac{1}{\eta_{\max}}$, then we have

$$\begin{aligned}
R\eta_{\max}(S_1(t) + \frac{1}{C})^{-\alpha-\rho-1} &\leq R\eta_{\max}^2(S_1(t) + \frac{1}{C})^{-\alpha-\rho} \\
&= O(\eta_{\max}^2).
\end{aligned}$$

This completes the proof of Theorem 1. □