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 derive an empirical law that predicts pretraining loss for large language models for every intermediate training step across various learning rate schedules, including 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 as learning rate decays. We validate this law extensively on Llama-2 models of varying sizes and demonstrate that, after fitting on a few learning rate schedules, it 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 slightly faster convergence. We believe these results could offer valuable insights for understanding the dynamics of pretraining and for designing learning rate schedules to improve efficiency.

1 Introduction

Language models can achieve strong performance if pretrained at a very large scale with an appropriate configuration of hyperparameters, such as model width, model depth, number of training steps, and learning rate. However, a full-scale grid search over these hyperparameters is often impossible since one large-scale pretraining run can take weeks or even months.

To reduce the cost of hyperparameter tuning, researchers have proposed various scaling laws that aim to predict the final pretraining loss or downstream performance at scale. These laws usually try to capture an empirical relationship between the final performance and a few key hyperparameters, and use a simple parameterized function to approximate this relationship. A notable example is the Chinchilla scaling law (Hoffmann et al., 2022), which approximates the final pretraining loss as a

[†]Corresponding authors

Mathematics of Modern Machine Learning Workshop at NeurIPS 2024.

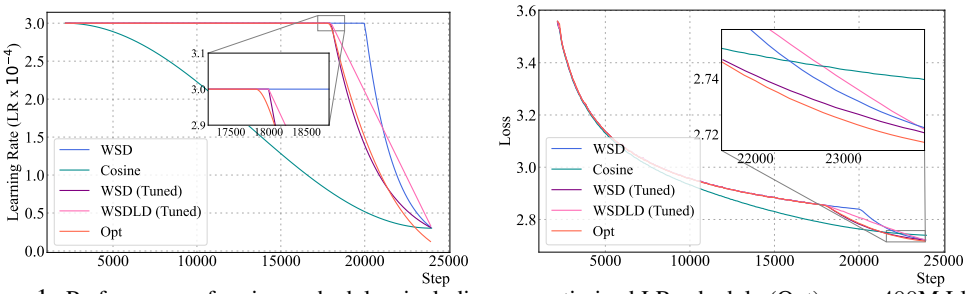


Figure 1: Performance of various schedules, including our optimized LR schedule (Opt), on a 400M Llama-2 (Touvron et al., 2023) model over 12B tokens. Zoom in/out facilitates the readers who are interested in the local details. **(a)** Our optimal schedule comprises constant and decay stages post-warmup, aligning with WSD (Hu et al., 2024). See Section 5 for details. **(b)** Our optimized schedule outperforms cosine LR and tuned WSD variants (WSD uses exponential decay; WSDLD uses linear decay).

function of the model size N and the total number of training steps T (or alternatively, the number of training tokens), $\mathcal{L}(N, T) = L_0 + A \cdot T^{-\alpha} + B \cdot N^{-\beta}$. Based on a few experiments with varying N and T, one can fit the parameters L_0, A, B, α, β and use the formula to infer the optimal choice of N and T given a fixed compute budget $C = NT$.

However, existing scaling laws fall short in providing guidance on the choice of *Learning Rate* (LR), which is arguably the most critical hyperparameters in optimization. It is indeed very challenging to incorporate the effect of LR into the scaling laws, as its impact on the training speed and stability is intricate and not yet well understood in a quantitative manner. A qualitative understanding is as follows: a large LR can reduce the training loss quickly, but in the long term, it may cause overshooting and oscillation along sharp directions on the loss landscape. In contrast, a small LR leads to a more stable training process, but at the cost of slowing down the convergence. Practitioners often trade-off between these two extremes 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 is gradually increased from a small value 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 language model 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 which schedule is the best.

In this paper, we aim to obtain a quantitative understanding of the empirical relationship between the LR schedule and the final training loss in language model pretraining. 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?* Following the standard practice in pretraining, we assume that each training step is taking fresh samples from a data stream, thus there is no generalization gap between the training and test loss. We focus on learning rate schedules that decay the LR over time ($\eta_t \le \eta_{t-1}$) as these schedules are the most common in practice.[2] Moreover, we assume that we have already picked a good initial LR $\eta_{\max}$ that is nearly optimal for short training runs without LR decay. Starting from this initial LR $\eta_1 = \eta_{\max}$, we are interested in predicting the loss curve as the LR decays over time.

Existing scaling laws are insufficient for this problem because they are usually overfitted to one predetermined LR schedule. For example, Hoffmann et al. (2022) fitted the parameters L_0, A, B, α, β in the Chinchilla scaling law $\mathcal{L}(N, T) = L_0 + A \cdot T^{-\alpha} + B \cdot N^{-\beta}$ for training runs that have gone through the entire cosine LR schedule, which makes the law inapplicable to other LR schedules, or even to the same LR schedule with early stopping. Finding a good scaling law for this problem also requires a more sophisticated approach. In contrast to existing laws that make predictions based on two or three hyperparameters, which are easy to visualize and analyze, here we need to predict the loss curve based on the entire LR schedule, which is inherently high-dimensional. A more careful experimental design is thus needed to speculate what a good approximation formula could be.

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

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

[2] If there is a LR warmup phase in the schedule, we focus on the decay phase right after the warmup phase.

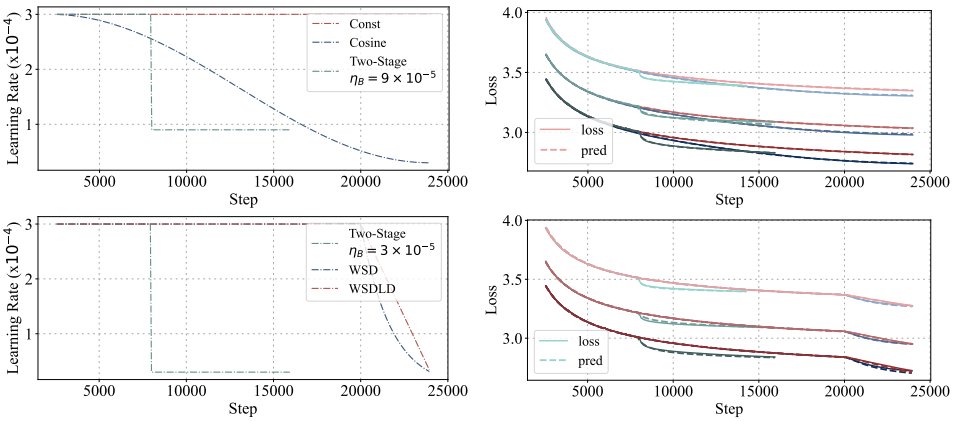


Figure 2: Loss Curves of 25M, 100M, and 400M models from up to down. (a) **Fit on Training Set:** Our multi-power law is reducible to two-stage and Constant LR schedules, and captures Cosine LR decay effects. (b) **Prediction on Test Set:** Our law generalizes to unseen schedules like WSDLD and WSD, and handles steep decays in Two-Stage cases.

Here, $L_0 + A \cdot S_1(t)$ can be seen as a naïve extension of the Chinchilla scaling law by replacing the number of steps T with the sum of LRs and neglecting the dependence on the model size. This alone can be seen as a crude approximation of the loss curve that linearizes the contribution of the LR at each step, but it is agnostic to the shape of LR decay. The remaining term $\mathrm{LD}(t)$ serves as a correction term that captures how decaying the LR to smaller values leads to a reduction in the loss:

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

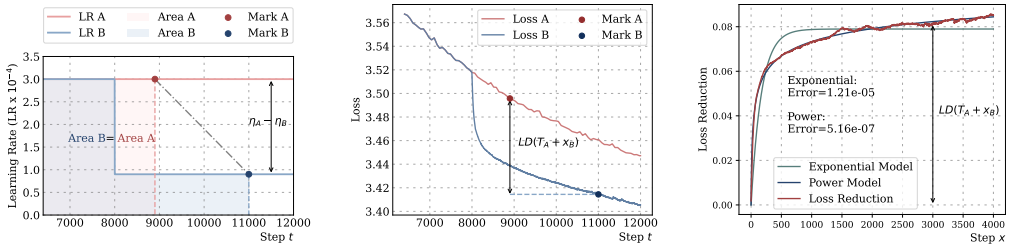
More specifically, $\mathrm{LD}(t)$ is the sum of the LR reduction $\eta_{k-1} - \eta_k$ at each step k multiplied by a nonlinear factor. The factor gradually saturates to a constant as the training progresses, and the speed of saturation follows a power law in a scaled sum of LRs $\eta_k^{-\gamma} S_k(t)$.

We call this law of $\mathcal{L}(t)$ the *multi-power scaling law* as it consists of multiple power-law forms. See also Appendix F.3 for the practical version of our law that accounts for the warmup phase. $L_0, A, B, C, \alpha, \beta, \gamma$ are the parameters of the law and can be fitted by running very few pretraining experiments with different LR schedules. We summarize our main contributions 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 1). 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 7). 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 2).
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 (Section 3).

2 Empirical Derivation of the Multi-Power Law

In this section, we present our empirical derivation of the multi-power law for schedule-aware loss curve prediction. In the first place, we reduce the problem to studying a loss reduction term led by LR decay. Then we take a "bottom-up" approach to study this term for LR schedules with increasing complexity, from two-stage, multi-stage, to general LR decay schedules. For the first two cases, we



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

Figure 3: Example of Two-Stage cases: 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 $LD(T_A + x_B) = L_A(t_A) - L_B(t_B)$. **(c)** Fitting Loss Reduction $\hat{LD}(T_A + x_B)$ 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 instead of exponential.

conduct extensive ablation studies on the behavior of the training loss and derive formulas that can accurately predict the loss reduction term. This has finally inspired us to propose the multi-power law for general cases as a natural unification and generalization of the formulas derived for the two special cases. We will further validate our law with extensive experiments in Section 4.

Background: Learning Rate Schedule. An LR schedule is a sequence $E := \{\eta_1, \dots, \eta_T\}$ that specifies the LR at each step of the training process. In the domain of language model pretraining, the cosine LR schedule (Loshchilov & Hutter, 2017) is the most popular one, which can be expressed as $\eta_t = \frac{1+\alpha}{2}\eta_{\max} + \frac{1-\alpha}{2}\eta_{\max}\cos(\frac{\pi t}{T})$, where $\eta_{\max}$ is the maximum LR and α 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 with W steps, then maintains at a stable LR $\eta_{\max}$ with T_{stable} steps, and finally decays in the form $f(s - T_{\text{stable}})\eta_{\max}$ during stage $T_{\text{stable}} \le s \le T_{\text{total}}$. Here $f(x) \in (0, 1)$ can be chosen as linear or exponential decay functions. The visualization of these two LR schedules is in Figure 1(a).

Background: Warmup Phase. Many LR schedules, such as WSD, contain a warmup phase that increases the LR gradually from 0 to the maximum LR $\eta_{\max}$ over a few steps. Our discussion focuses on the training after the warmup, where the LR is non-increasing in almost all LR schedules. The steps are counted after the warmup phase, i.e., $t = 1$ is the first step after the warmup.

2.1 Our Approach: Learning Rate Sum Matching

Auxiliary Training Process. We first introduce an auxiliary training process to aid our analysis of the loss curve of the actual training process with LR schedule $E := \{\eta_1, \dots, \eta_T\}$. This auxiliary training process is exactly the same as the actual training process for the first K steps, where K is the largest number such that $\eta_1 = \eta_2 = \cdots = \eta_K$. Then the auxiliary training process continues training with a constant LR schedule, where the LR is set to η_1 for all the remaining steps. We denote the training loss at step t in this auxiliary process as $\mathcal{L}_{\text{const}}(t)$.

Learning Rate Sum Matching. The multi-power law for approximating the loss curve $\mathcal{L}(t)$ of the actual training process is based on the following decomposition. Define $Z(t)$ as the step in the auxiliary process that has the same sum of LRs as the actual training process at step t. Then,

$$\mathcal{L}(t) = \mathcal{L}_{\text{const}}(Z(t)) - \underbrace{(\mathcal{L}_{\text{const}}(Z(t)) - \mathcal{L}(t))}_{=:\ \text{LD}(t)}, \quad \text{where} \quad Z(t) := \frac{1}{\eta_1}\sum_{\tau=1}^{t}\eta_\tau. \tag{3}$$

Here, we first use the training loss at step $Z(t)$ in the auxiliary process, $\mathcal{L}_{\text{const}}(Z(t))$, as an approximation for $\mathcal{L}(t)$, and then write the approximation error term as $\text{LD}(t)$. We call $\text{LD}(t)$ the ***L**oss re**D**uction term* as it is a quantification of the reduction of loss due to learning rate decay.

The rationale behind this approach is that matching the LR sum between the two training processes should result in similar training losses, and thus a more accurate approximation can be obtained by further exploring the loss reduction term LD(t). See Appendix F.2 for more discussion.

Power-Law Ansatz for the Auxiliary Loss. Under constant LR schedules, it is easy to predict the loss curve accurately. Taking inspiration from previous works (Hoffmann et al., 2022; Kaplan et al., 2020), we choose to take a power-law form to approximate the training loss of the auxiliary process, which works reasonably well in our experiments. That is,

$$\mathcal{L}_{\text{const}}(t) \approx \widehat{\mathcal{L}}_{\text{const}}(t) := L_0 + \tilde{A} \cdot t^{-\alpha}, \tag{4}$$

where $L_0, \tilde{A}, \alpha$ are parameters. Replacing t with $Z(t) := \frac{1}{\eta_1}\sum_{\tau=1}^{t}\eta_\tau = \frac{1}{\eta_1}S_1(t)$ gives $\widehat{\mathcal{L}}_{\text{const}}(Z(t)) = L_0 + \tilde{A}\eta_1^{\alpha}S_1^{-\alpha}(t)$, where $S_1(t) := \sum_{\tau=1}^{t}\eta_\tau$. Finally, we reparameterize $\tilde{A}$ as

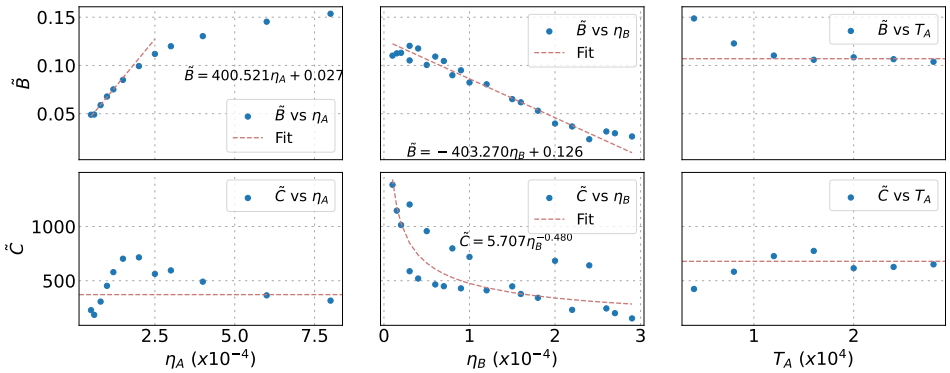


Figure 4: The patterns of $\tilde{B}, \tilde{C}$ over η_A, η_B and T_A in Two-Stage cases. For illustration purposes, the second row uses $\tilde{C}$ as the y-axis. $\tilde{B}$ can be approximated to be proportional to $\eta_A - \eta_B$, and $\tilde{C}$ manifests power-law pattern over η_B. The effect of η_A on $\tilde{C}$ and the impact of T_A are unpredictable or negligible, which is approximately ignored in our discussion.

$\tilde{A} := A\eta_1^{-\alpha}$, where A is a parameter, and obtain the formula:

$$\widehat{\mathcal{L}}_{\text{const}}(Z(t)) = L_0 + A \cdot S_1^{-\alpha}(t). \tag{5}$$

See Figure 13 for an empirical validation of eq. (5) for various constant LR schedules.

Loss Reduction Term. It remains to understand the behavior of the loss reduction term $\text{LD}(t)$, which is inherently complex since it depends on each LR used in training. In the rest of the section, we conduct a series of experiments to determine an accurate approximation form for $\text{LD}(t)$.

2.2 Case 1: Two-stage Learning Rate Schedule

To understand the behavior of $\text{LD}(t)$, we start with the simplest form of LR decay that consists of two stages: In Stage 1, the LR remains constant at η_A for T_A steps ($\eta_1 = \eta_2 = \cdots = \eta_{T_\text{A}} = \eta_\text{A}$); in Stage 2, the LR suddenly decreases to η_B and the rest of training continues with η_B for T_B steps ($\eta_{T_\text{A}+1} = \eta_{T_\text{A}+2} = \cdots = \eta_{T_\text{A}+T_\text{B}} = \eta_\text{B}$). We call this LR schedule a *two-stage LR schedule*. In this case, the first T_A steps of the auxiliary and actual training processes are the same, and the loss reduction term $\text{LD}(t)$ becomes non-zero only in Stage 2.

The Loss Reduction Term Follows a Power Law. In Figure 3, we plot the loss reduction term $\text{LD}(t)$ of a two-stage learning rate schedule with $\eta_\text{A} = 3 \times 10^{-4}$, $T_\text{A} = 8000$, $\eta_\text{B} = 9 \times 10^{-5}$. As the number of steps $x := t - T_\text{A}$ in Stage 2 increases, $\text{LD}(T_\text{A} + x)$ monotonically rises from 0 to around 0.09 and eventually saturates. This motivates us to approximate $\text{LD}(T_\text{A} + x)$ in the form $\tilde{B} \cdot (1 - U(\eta_\text{B} x))$, where $\tilde{B}$ is a parameter and $U(s)$ is a function that decreases from 1 to 0 as $s = \eta_\text{B} x$ increases from 0 to infinity. The reason we choose $\eta_\text{B} x$ instead of x as the argument of U will be clear when we generalize this to multi-stage schedules.

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

$$\text{LD}(T_\text{A} + x) \approx \widehat{\text{LD}}(T_\text{A} + x) := \tilde{B}(1 - (\tilde{C} \cdot \eta_\text{B} x + 1)^{-\beta}). \tag{6}$$

Figure 3(c) shows that this power law aligns well with the actual loss reduction term $\text{LD}(T_\text{A} + x)$. In contrast, the exponential form $U(s) = e^{-Bs}$ (so $\text{LD}(T_\text{A} + x) \approx A(1 - e^{-B\eta_\text{B} x})$) struggles to match the slow and steadily increase of $\text{LD}(T_\text{A} + x)$ when x is large.

We further investigate how to estimate the parameters $\tilde{B}, \tilde{C}, \beta$ in the power law. Our preliminary experiments suggest that the power law fits the loss reduction term very well with a constant β that is independent of $\eta_\text{A}, \eta_\text{B}, T_\text{A}$, so we just set $\beta = 0.4$, which is a constant that works well. Then we conduct experiments to understand how the best parameters $\tilde{B}, \tilde{C}$ to fit $\text{LD}(t)$ depend on $\eta_\text{A}, \eta_\text{B}, T_\text{A}$, where we set default values $\eta_\text{A} = 3 \times 10^{-4}$, $\eta_\text{B} = 3 \times 10^{-5}$, $T_\text{A} = 8000$ and change one variable at a time. See Appendix G for experiment details.

$\tilde{B}$ is Linear to LR Reduction. Our first observation is that $\tilde{B} \propto \eta_\text{A} - \eta_\text{B}$. As shown in the first row of Figure 4, $\tilde{B}$ linearly decreases with η_B and approximately increases linearly with η_A, especially when η_A is not too large. Moreover, the slope of $\tilde{B}$ over η_A and η_B are approximately opposite to each other. This motivates us to hypothesize that $\tilde{B} \propto \eta_\text{A} - \eta_\text{B}$ and reparameterize $\tilde{B}$ as $\tilde{B} = B(\eta_\text{A} - \eta_\text{B})$, where B is a constant.

$\tilde{C}$ Follows a Power Law of η_B. Our second observation is that $\tilde{C}$ follows a power law. As shown in the second row of Figure 4, we find that $\tilde{C}$ is very sensitive to η_B but much less dependent on η_A.

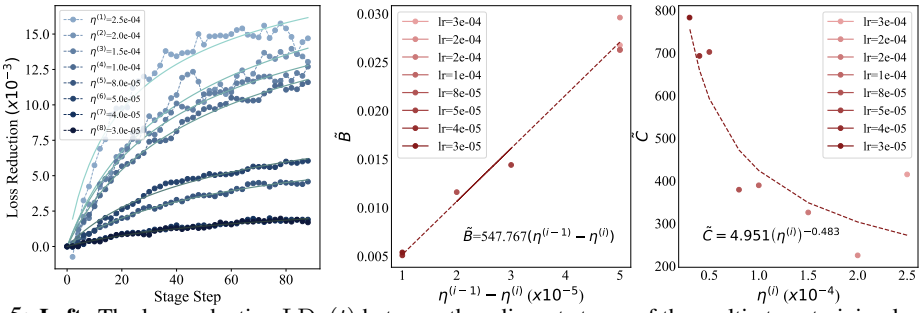


Figure 5: **Left:** The loss reduction $\mathrm{LD}_k(t)$ between the adjacent stages of the multi-stage training loss curve still follows the power form. The multi-stage loss curve refers to Figure 12. **Right:** The parameter patterns in the two-stage setting hold in the multi-stage setting approximately. $\tilde{B} \propto \eta^{(i-1)} - \eta^{(i)}$ and $\tilde{C} \propto \eta^i$ keep and the shape of patterns are similar to the patterns in our ablation experiments, as shown in Figure 4.

We hypothesize that $\tilde{C}$ follows a power law $\tilde{C} \propto \eta_{\mathrm{B}}^{-\gamma}$, and reparameterize $\tilde{C}$ as $\tilde{C} = C\eta_{\mathrm{B}}^{-\gamma}$, where $C > 0$ and $\gamma > 0$ are constants.

LR Reduction Term Depends Less on T_{A}. We also find that $\tilde{B}$ and $\tilde{C}$ are less sensitive to T_{A}. As shown in the last column in Figure 4, $\tilde{B}$ and $\tilde{C}$ are relatively stable as T_{A} varies. This suggests that the loss reduction induced by a single LR decay is approximately independent from the time that LR decays. This insight will be revisited when we generalize the two-stage case to multi-stage cases.

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

$$\mathrm{LD}(T_1 + x) \approx \widehat{\mathrm{LD}}(T_1 + x) := B(\eta_{\mathrm{A}} - \eta_{\mathrm{B}})(1 - (C\eta_{\mathrm{B}}^{1-\gamma}x + 1)^{-\beta}). \tag{7}$$

2.3 Case 2: Multi-Stage Leaning Rate Schedule

We go one step further from two-stage step decay schedule to multi-stage step decay schedules. This class of schedules consists of multiple stages, where the LR decays when a new stage starts but remains constant within each stage. Now, we consider an n-stage LR schedule $E = \{\eta_1, \ldots, \eta_T\}$, where the i-th stage lasts from step $t = T_{i-1} + 1$ to $t = T_i$ and uses the LR $\eta_t = \eta^{(i)}$ ($0 = T_0 < T_1 < \cdots < T_{n-1} < T_n = T$, $\eta^{(1)} \geq \eta^{(2)} \geq \cdots \geq \eta^{(n)}$). See Figure 12 for an example.

Multi-Stage Loss Reduction. To draw insights into the behavior of $\mathrm{LD}(t)$ in the multi-stage case, we use the following strategy. Recall that $\mathrm{LD}(t)$ is the difference in training losses between the auxiliary and actual training processes at equal LR sums. In addition to these processes, we construct some intermediate processes: for $1 \leq i \leq n$, we define the i-th process to be the same as the actual training process in stages 1 to i but continue to use the learning rate $\eta^{(i)}$ for all stages after i. The first and last processes are the auxiliary and actual training processes themselves.

We again use the trick of LR sum matching: we find the steps of these n processes that have the same LR sum, and then conduct experiments to analyze the loss difference between adjacent processes. Let $\mathcal{L}_i(t)$ be the training loss of the i-th process at step t. For $1 \leq i \leq j \leq n$ and $t \geq T_{j-1}$, we define $Z_{i,j}(t)$ as the step number in the i-th process that has the same LR sum as the j-th process at step t, i.e., $Z_{i,i}(t) := t$ and $Z_{i,j}(t) := T_i + \frac{1}{\eta^{(i)}} \sum_{\tau=T_i+1}^{t} \eta_\tau$ for $i < j$. Then we have

$$\mathrm{LD}(t) = \sum_{k=2}^{i} \mathrm{LD}_k(Z_{k,n}(t)), \quad \text{where} \quad \mathrm{LD}_k(t) := \mathcal{L}_{k-1}(Z_{k-1,k}(t)) - \mathcal{L}_k(t). \tag{8}$$

Here, $\mathrm{LD}_k(t)$ is the difference between the $(k-1)$-th and k-th processes at equal LR sums. These two processes are the same for the first $k-1$ stages and diverge only at the beginning of the k-th stage: the former continues to use $\eta^{(k-1)}$ but the latter switches to $\eta^{(k)}$. This is similar to the two-stage case, except that the first $k-1$ stages may not use the same LR.

Interestingly, the power law behavior of the loss reduction term observed in the two-stage case also approximately holds for $\mathrm{LD}_k(t)$. As the training enters a new stage $i+1$, a new loss reduction term $\mathrm{LD}_i(\cdot)$ is introduced in (8). We observe that $\mathrm{LD}_i(T_i + x)$ follows a similar power law behavior as in the two-stage case when x increases. As shown in Figure 5, each $\mathrm{LD}_i(T_i + x)$ can be individually approximated by a power law $\tilde{B}(1 - (\tilde{C} \cdot \eta^{(i)}x + 1)^{-\beta})$, and $\tilde{B} \propto (\eta^{(i-1)} - \eta^{(i)})$, $\tilde{C} \propto (\eta^{(i)})^{-\gamma}$.

Final Approximation Form. Inspired by the above observation and our approximation (7) for the two-stage case, we propose to approximate $\mathrm{LD}_k(t)$ with a power law:

$$\mathrm{LD}_k(T_{k-1} + x) \approx \widehat{\mathrm{LD}}_k(T_{k-1} + x) := B(\eta^{(k-1)} - \eta^{(k)})(1 - (C(\eta^{(k)})^{1-\gamma}x + 1)^{-\beta}). \tag{9}$$

Plugging this approximation into (8), we can approximate the loss reduction term $\mathrm{LD}(t)$ at step $T_{i-1} < t \leq T_i$ of the actual training process as $\mathrm{LD}(t) \approx \widehat{\mathrm{LD}}(t) := \sum_{k=2}^{i} \widehat{\mathrm{LD}}_k(Z_{k,n}(t))$, then

$$\mathrm{LD}(t) \approx \widehat{\mathrm{LD}}(t) = \sum_{k=2}^{i} B(\eta^{(k-1)} - \eta^{(k)})(1 - (C(\eta^{(k)})^{1-\gamma}(Z_{k,n}(t) - T_k) + 1)^{-\beta}).$$

By definition of $Z_k(t)$, we have $Z_{k,n}(t) - T_k = \frac{S_k(t)}{\eta^{(k)}}$, where $S_k(t) := \sum_{\tau=T_k+1}^{t} \eta_\tau$ is the sum of LRs from the beginning of Stage $k+1$ to step t. We can further simplify the above formula as

$$\mathrm{LD}(t) \approx \widehat{\mathrm{LD}}(t) = \sum_{k=2}^{i} B(\eta^{(k-1)} - \eta^{(k)})(1 - (C(\eta^{(k)})^{-\gamma} S_k(t) + 1)^{-\beta}). \tag{10}$$

2.4 General Case

Ansatz for the Loss Reduction Term. A general learning rate schedule $E := \{\eta_1, \ldots, \eta_T\}$ could be viewed as a T-stage schedule, where the i-th stage uses learning rate η_i for $l_i = 1$ step. This motivates us to make the ansatz that the formula for the loss reduction term $\mathrm{LD}(t)$ in multi-stage schedules (10) can continue to hold even when every stage only lasts for one step.

$$\mathrm{LD}(t) \approx \widehat{\mathrm{LD}}(t) = \sum_{k=2}^{t} B(\eta_{k-1} - \eta_k)(1 - (C\eta_k^{-\gamma} S_k(t) + 1)^{-\beta}), \tag{11}$$

where $S_k(t) := \sum_{\tau=k}^{t} \eta_\tau$ is the sum of LRs from step k to step t, and B, C, γ, β are parameters.

Multi-Power Law. Following the approach of learning rate sum matching in Section 2.1, we first decompose $\mathcal{L}(t)$ as $\mathcal{L}_{\mathrm{const}}(Z(t)) - \mathrm{LD}(t)$ (see (3)). Then we combine the above ansatz for the Loss reduction term with the power-law ansatz for the auxiliary loss, leading to our multi-power law:

$$\mathcal{L}(t) \approx L_0 + A \cdot S_1^{-\alpha}(t) - \sum_{k=2}^{t} B(\eta_{k-1} - \eta_k)(1 - (C\eta_k^{-\gamma} S_k(t) + 1)^{-\beta}).$$

See also Appendix F.3 for the practical version of our law that accounts for the warmup phase by slightly changing the $A \cdot S_1^{-\alpha}(t)$ term. See Appendix B.1 for the discussion about the simplification of the multi-power law.

3 How Might the Multi-Power Law Arise?

In this section, we aim to understand how the multi-power law might arise from the optimization. However, it is generally hard to prove convergence bounds for deep learning under realistic assumptions. For this reason, most previous theoretical works trying to establish a scaling law have mostly focused on linear models or even simpler cases (see **??** for a literature review), and capturing the effect of learning rate schedules in theory can be even more challenging.

Here, we present a preliminary theoretical analysis for a solvable model: when optimizing quadratic loss functions, if the Hessian and noise covariance exhibit a power-law decay in their eigenvalues, then our multi-power law can be proved to emerge.

3.1 Setup

We consider a quadratic loss $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^\top \boldsymbol{H}\boldsymbol{\theta}$, where $\boldsymbol{\theta} \in \mathbb{R}^d$. Linear regression can be viewed as a special case if we shift the minimizer to the origin. We use $\Phi(\boldsymbol{\theta}_0, E)$ be the distribution of the T-th iteration $\boldsymbol{\theta}_T$ of gradient descent, defined by the recursion $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \eta_t \boldsymbol{g}_t$, where $\boldsymbol{g}_t$ is the stochastic gradient at step t, following a normal distribution $\mathcal{N}(\boldsymbol{H}\boldsymbol{\theta}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ being the covariance matrix.

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 argue that the loss follows a multi-power law.

Assumption 1. *Let λ_i be the ith eigenvalue of $\boldsymbol{H}$, and Σ_{ii} be the element of $\boldsymbol{\Sigma}$ in the ith column and ith 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*

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

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

3.2 Loss Formula

Based on the setup and assumption, we could derive the loss formula achieved by Stochastic Gradient Descent with iteration t steps.

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

$$\begin{aligned}\tilde{M}_t(\boldsymbol{\theta}_0, E) :=& L_0 + AS_1(t)^{-\alpha-2} - R\eta_{\max}(S_1(t) + \frac{1}{C})^{-\alpha-\rho-1} \\ & - B\sum_{k=2}^{T}(\eta_{k-1} - \eta_k)\left(1 - (C\,S_k(t) + 1)^{-\alpha-\rho-1}\right),\end{aligned}$$

where $L_0 = \frac{d}{4}\eta_{\max}\mu$, and A, B, C, R are positive constants independent of LR schedule E and $S_i(t) := \sum_{k=i}^{t}\eta_k$. The estimation error is bounded as

$$|\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_t)] - \tilde{M}_t(\boldsymbol{\theta}_0, E)| = O(S_1(t)^{-\alpha-3} + \eta_{\max}^2).$$

Compared with law (1), the above derivation contains an additional term that $R\eta_{\max}(S_1(t) + \frac{1}{C})^{-\alpha-\rho-1}$. The next corollary from Theorem 1 states that this term can be negligible under quite a mild condition so that, in our theoretical setting, we can completely match the multi-power law derived empirically.

Corollary 1. *If $S_1(t) > \frac{1}{\eta_{\max}}$, with the same setting in Theorem 1, we have the following estimate of $\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_t)]$ for any $0 \leq t \leq T$:*

$$\tilde{M}_t(\boldsymbol{\theta}_0, E) := \underbrace{L_0 + AS_1(t)^{-\alpha-2}}_{\textit{constant LR term}} \underbrace{- B\sum_{k=2}^{T}(\eta_{k-1} - \eta_k)\left(1 - (C\,S_k(t) + 1)^{-\alpha-\rho-1}\right)}_{\textit{loss reduction term}}$$

The estimation error is bounded as

$$|\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_t)] - \tilde{M}_t(\boldsymbol{\theta}_0, E)| = O(S_1(t)^{-\alpha-3} + \eta_{\max}^2)$$

The detailed proof of Theorem 3 and Corollary 1 can be found in Appendix L. Beyond this quadratic case, to get a more systematic theory, which is 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).

4 Empirical Validation of the Multi-Power Law

The multi-power law (MPL) comes from our speculations from 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 learning rate (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. Details of the experimental setup, fitting approaches, and configurations are provided in Appendix H.

4.1 Results

Generalization to Unseen LR Schedules. The MPL can accurately predict loss curves for LR schedules outside the training set. As illustrated in Figure 2, despite the absence of WSD LR 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 η_{B} values from the training set, effectively extrapolating for both continuous and discontinuous cases.

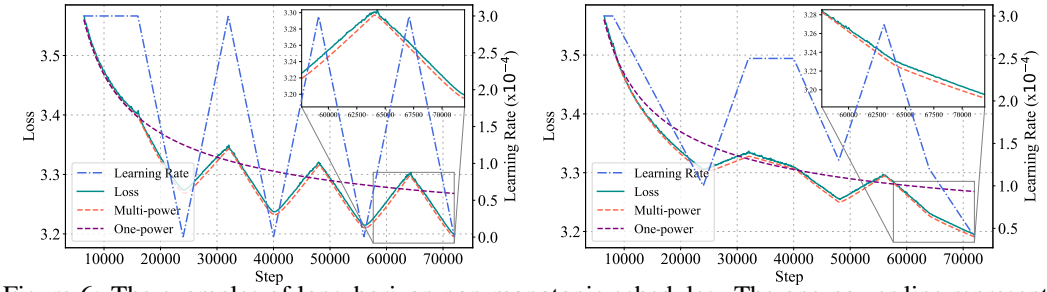


Figure 6: The examples of long-horizon non-monotonic schedules. The one-power line represents the auxiliary process curve. **Left:** The cyclic schedule with 72,000 steps, where each half-cycle spans 8,000 steps, and the first decay begins after 16,000 steps. **Right:** The random-polyline schedule, consisting of piecewise linear interpolation between randomly selected intermediate learning rates in the range of 3×10^{-5} to 3×10^{-4}, with LR milestones occurring at intervals of 8,000 steps.

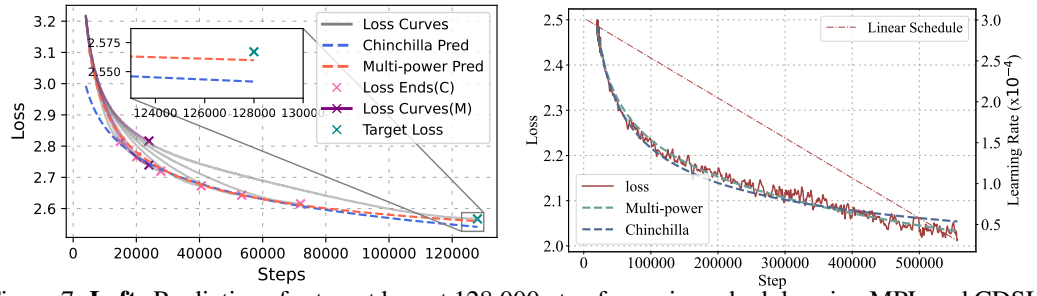


Figure 7: **Left:** Predictions for target loss at 128,000-step for cosine schedule using MPL and CDSL fitting. The CDSL uses the final losses of six cosine losses from 14960 steps to 72000 steps, marked as Loss Ends(C). The MPL uses 24000-step constant and cosine curves, marked as Loss Curves(M). **Right:** Comparison of MPL and CDSL fits on the open-source 7B OLMo curve generated with a linear schedule.

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. The test set includes 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 6, 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 utilized (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 restricted to final loss predictions. Based on these advantages, the MPL shows higher sample efficiency than the CDSLs. Moreover, we find that two curves of different schedules are enough to fit the MPL with generalizability, as details are discussed in Appendix B.3. As shown in Figure 7, MPL achieves $1/4$ error in final loss prediction with $1/4$ compute budget compared to CDSL. MPL also shows advantages in the fitting of open-source 7B OLMo (Groeneveld et al., 2024) in Figure 7.

Comparison with Momentum Law. The MPL shows higher accuracy and can apply to the discontinuous schedules compared to the recent Momentum Law (Tissue et al., 2024). The Momentum Law (MTL) (Tissue et al., 2024) incorporates LR annealing effects by modeling loss reduction based on the momentum of LR decay. However, MTL indicates an exponential loss reduction in two-stage LR schedules, which contradicts our observations (see Figure 3). Additionally, as shown in Figure 20, MPL outperforms MTL in predicting loss reduction for WSD schedules with linear LR decay. In the highlighted regions, MPL achieves high accuracy in the decay stage, whereas MTL exhibits substantial error. A summary of prediction results across test sets is provided in Table 1,

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**

where MPL consistently outperforms MTL in both average and worst-case scenarios. The details of the MTL and its relation to the MPL can be found in Appendix B.1.

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 very impossible to tune the LR for every training step. However, in this section, we show that by using the predicted final loss from the MPL, we can optimize the entire LR schedule to significantly reduce the final loss and beat the cosine schedule.

5.1 Method

Given that the Multi-Power Law (MPL) provides an accurate estimation of the loss, the final loss prediction by MPL can serve as a surrogate for evaluating schedules. Consider the learning rate (LR) as a T-dimensional vector $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_T)$ and the final loss $\mathcal{L}(\boldsymbol{\eta})$ under given hyperparameters. The goal is to identify the optimal LR schedule $\boldsymbol{\eta}^* = \arg\max_{\boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\eta})$. We parameterize the final loss prediction as $\mathcal{L}_\Theta(\boldsymbol{\eta})$ using MPL with parameters $\Theta = \{L_0, A, B, C, \alpha, \beta, \gamma\}$. The parameters $\hat{\Theta}$ can be estimated as described in Section 4. Using $\mathcal{L}_{\hat{\Theta}}(\boldsymbol{\eta})$ as a surrogate for $\mathcal{L}(\boldsymbol{\eta})$, we approximate $\boldsymbol{\eta}^*$ by solving:

$$\hat{\boldsymbol{\eta}} = \min_{\boldsymbol{\eta}} \mathcal{L}_{\hat{\Theta}}(\boldsymbol{\eta}) \quad \text{s.t.} \quad 0 \leq \eta_{i+1} \leq \eta_i, \forall\, 1 \leq i \leq T-1. \tag{12}$$

This process induces an "optimal" schedule $\hat{\boldsymbol{\eta}}$ derived from MPL with parameter $\hat{\Theta}$. We set the initial learning rate η_0 to 3×10^{-4} and assume η_i is monotonically non-increasing based on prior knowledge. The high-dimensional vector $\boldsymbol{\eta}$ is optimized using the Adam optimizer. Additional details are provided in Appendix I.

5.2 Results

Induced LR Schedule Exhibits Stable-Decay Behavior. The induced learning rate schedule follows a Warmup-Stable-Decay (WSD) pattern, comprising two main stages after the warmup phase. It maintains a peak LR for an extended period, followed by a rapid decay to a near-zero LR, as shown in Figure 1 and Figure 17.

Induced LR Schedule Outperforms Cosine Schedule. Figures 1 and 17 compare the induced schedules with the cosine and WSD schedules across models ranging from 25M to 400M. Figure 18 extends this comparison to longer training horizons. The induced schedules consistently outperform the cosine schedule, achieving a margin over 0.02. Notably, no WSD-like schedule is present in the training set, predicting such loss curves an extrapolation by MPL.

Characteristics of the Induced Schedules. The induced schedules provide insights into hyperparameter tuning for WSD schedules. Observations from Figures 1 and 17 highlight the following: (1) Our findings suggest that a lower ending LR—typically below $1/20$ of the peak LR—is more effective in most scenarios, compared to $1/10$ in prior research (Hoffmann et al., 2022; Kaplan et al., 2020). Further details are provided in Appendix I. (2) $f(x) = (1-x)^{-\alpha}$, where $\alpha \approx 1.5$, well captures relationship between normalized steps $\tilde{\boldsymbol{t}}$ and normalized LRs $\tilde{\boldsymbol{\eta}}_{\text{avg}}$ in our experiments. This simplified version, referred to as WSD with Sqrt-Cube Decay (WSDSC), is effective across various model sizes and types, as shown in Figures 8 and 21. See Appendix B.2. (3) The induced schedules align closely with the optimal decay steps identified via grid search, as illustrated in Figure 1. See Appendix I.

6 Conclusions and Future Directions

In this paper, we introduce the multi-power law for scheduler-aware loss curve prediction, accurately predicting loss curves and inspiring optimal scheduler derivation. Future work includes refining the law, exploring its underlying mechanisms, and studying the LR relationship with unfixed max LR. Our findings enhance understanding of training dynamics in large language models, potentially 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. *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.

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.

Zhengxiao Du, Aohan Zeng, Yuxiao Dong, and Jie Tang. Understanding emergent abilities of language models from the loss perspective, 2024. URL https://arxiv.org/abs/2403.15796.

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.

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.

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.

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.

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.

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.

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.

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

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.

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.

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.

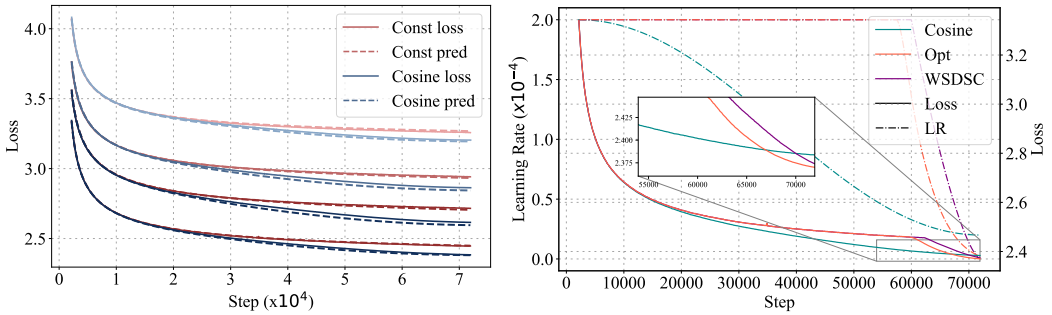


Figure 8: **Left:** Long horizon prediction for the cosine and constant schedules. From up to down, the model sizes range from 25M to 1B. **Right:** The comparison on 1B models between the optimized schedule (Opt), cosine schedule (Cosine), and the simplified optimized schedule (WSDSC, see Section 5.2), a WSD schedule with sqrt-cube decay.

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

Downstream Dataset	LAMBADA	HellaSwag	PIQA	ARC-E
Cosine Schedule	46.54	37.12	**65.13**	43.56
Induced Schedule	**48.71** (↑ 2.17%)	**37.74** (↑ 0.62%)	65.07 (↓ 0.06%)	**44.09** (↑ 0.53%)

A Discussion

In this section, we conduct experiments over hyper-parameters to check the applicability range of the multi-power law (MPL). The hyperparameters include the model types, model sizes, peak learning rates, and random seeds. In addition to empirical results, we can theoretically derive a multi-power law under a case with a quadratic loss function, providing insight into the nature of the MPL.

Model Types. We validate the MPL on GPT-2 (Radford et al., 2019) and OLMo (Groeneveld et al., 2024) models to evaluate the generalizability of the MPL across model architectures. In the preceding experiments, we used the Llama2 (Touvron et al., 2023). For experiments on GPT-2, the validation process followed the procedure fit with curves of cosine and constant schedules, described in Section 4. For the 7B OLMo model, we fit the MPL on the open-source training curve, which employs a linear decay schedule, as shown on the right of Figure 7. Our results show that the MPL presents a high prediction accuracy across different model types for both self-run and open-source experiments. Details see Appendix J.

Model Size. We extended the MPL and its induced schedule to a larger scale by training a 1B-parameter model on 144B data tokens. The MPL was fitted over 24,000 steps and successfully predicted loss curves up to 72,000 steps, as shown in Figure 8. We tested the performance of the induced 72,000-step schedule and its simplified version (see Section 5.2) against the widely used cosine schedule. The induced schedule outperformed the cosine schedule, while the simplified version achieved results between the induced and cosine schedules. To further validate the effectiveness of the induced schedules, we compared downstream task performance for models trained using the cosine and induced schedules. As shown in Table 2, the induced schedule led to overall improvements in downstream tasks. Details see Appendix J.

Peak Learning Rate Ablation. We evaluated the applicability of the MPL across different peak learning rates. In previous experiments, the peak learning rate was fixed at 3×10^{-4}. However, as shown in Figure 4, the empirical behavior of two-stage learning rate schedules deviates when the peak learning rate increases. To investigate this, we conducted experiments with peak learning rates of 4×10^{-4} and 6×10^{-4}. The MPL achieved an average R^2 value of 0.9965 for the 4×10^{-4} case and 0.9940 for the 6×10^{-4} case, demonstrating consistently high accuracy. Details see Appendix J.

Batch Size Ablation. We conduct ablation experiments on sequence batch sizes of 64 and 256 over 25M models, apart from 128 in previous experiments. The MPL presents a consistent accuracy with R^2 higher than 0.9970. See Appendix J.

Random Seed. We performed an ablation study to examine the impact of random seed variability on curves. We trained a 25M-parameter model for 24,000 steps using the cosine schedules with three random seeds. As shown in Figure 20, the standard deviation of the resulting loss values was approximately less than 0.001, establishing a lower bound for prediction errors. It highlights the prediction accuracy of the MPL discussed in Section 4.

Theoretical Results. We also present a theoretical analysis for quadratic loss functions optimizing Gradient Descent (GD) with noise. We can prove that the multi-power law arises when the Hessian

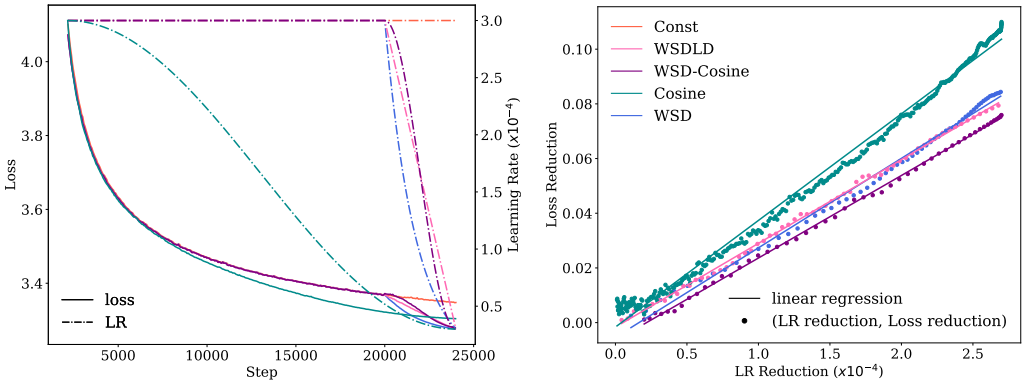


Figure 9: Linear regression between loss reduction and LR reduction over different schedules. The total step number is 24000 and the model size is 25M. WSD-Cosine denotes the WSD schedule with cosine decay function. The decay steps for the WSD schedule and variants are 4000. **Left:** the learning rate schedules and corresponding loss curves. **Right:** the loss reductions over LR reductions for different schedules, as well as their linear regression. The mean R^2 value is 0.9980.

and noise covariance matrices follow a power-law distribution in their eigenvalues. See Appendix 3 for more explicit derivation.

B Simplification of Formula, Usage and Fitting.

B.1 Simplification of Formula

We simplify the full multi-power law (MPL; see Equation (1)) at various levels, trading computational complexity for reduced accuracy. Table 3 summarizes the fitting performance of simplified versions and variants of the MPL. The fitting experiments are conducted over 25M models.

Table 3: Summary of fitting results for simplified laws. Each row corresponds to a specific law, reporting metrics including R^2, MAE, RMSE, PredE, and WorstE. Higher R^2 values and lower MAE, RMSE, PredE, and WorstE indicate better fitting performance. See Table 1 for metric definitions.

Formula	Differences from MPL	$R^2 \uparrow$	MAE$\downarrow$	RMSE$\downarrow$	PredE$\downarrow$	WorstE$\downarrow$
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-γ	$\gamma = 0$	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)	**0.9975**	**0.0039**	**0.0046**	**0.0012**	**0.0040**

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 $B = 0$:

$$\mathcal{L}_{\text{OPL}}(t) = L_0 + A \cdot S_1(t)^{-\alpha}, \quad S_1(t) := \sum_{\tau=1}^{t} \eta_\tau. \tag{13}$$

This formulation approximates the loss curve by linearizing the cumulative learning rate (LR) effects. 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. The loss reduction term $LD(t)$ (defined in Equation (2)) can be simplified by treating the scaling function $G(x)$ as a constant:

$$\mathrm{LD}(t) \approx \sum_{k=2}^{t} B(\eta_{k-1} - \eta_k) = B(\eta_1 - \eta_t).$$

Despite its simplicity, we observe a near-linear relationship between $LD(t)$ and the LR reduction $(\eta_1 - \eta_t)$, regardless of the LR schedule type, as shown in Figure 9. This motivates the Linear Loss reDuction Law (LLDL):

$$\mathcal{L}_{\mathrm{LLDL}}(t) = L_0 + A \cdot S_1(t)^{-\alpha} + B(\eta_1 - \eta_t). \tag{14}$$

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 trivial solutions.

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

$$\mathcal{L}_{\mathrm{No}-\gamma} = L_0 + A \cdot S_1(t)^{-\alpha} + B \sum_{k=2}^{t} (\eta_{k-1} - \eta_k) \cdot G(S_k(t)). \tag{15}$$

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

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

$$\mathcal{L}_{\mathrm{SPL}} = L_0 + A \cdot S_1(t)^{-\alpha} + B \sum_{k=2}^{t} (\eta_{k-1} - \eta_k) \cdot G(t-k). \tag{16}$$

While simpler, this approximation reduces prediction accuracy and contradicts empirical results, as it implies loss reduction 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):

$$\mathcal{L}_{\mathrm{MEL}} = L_0 + A \cdot S_1(t)^{-\alpha} + B \sum_{k=2}^{t} (\eta_{k-1} - \eta_k) \cdot G(S_k(t)). \tag{17}$$

Results (fifth row of Table 3) show a performance drop compared to the power-based MPL, consistent with observations in Section 2.

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

$$\mathcal{L}_{\mathrm{MTL}}(t) = L_0 + A \cdot S_1(t)^{-\alpha} + B \cdot S_2, \text{ where } S_1 = \sum_{i=1}^{t} \eta_i,\ S_2 = \sum_{i=2}^{t} \sum_{k=2}^{i} (\eta_{k-1} - \eta_k) \lambda^{i-k}.$$

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

$$S_2 = \sum_{i=2}^{t} \sum_{k=2}^{i} (\eta_{k-1} - \eta_k) \lambda^{i-k} = \sum_{k=2}^{t} (\eta_{k-1} - \eta_k) \sum_{i=k}^{t} \lambda^{i-k} = \sum_{k=2}^{t} (\eta_{k-1} - \eta_k) \left(\frac{1 - \lambda^{t-k+1}}{1 - \lambda} \right).$$

Thus, the Momentum Law (MTL) is a variant of MPL with an exponential step-based approximation:

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

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

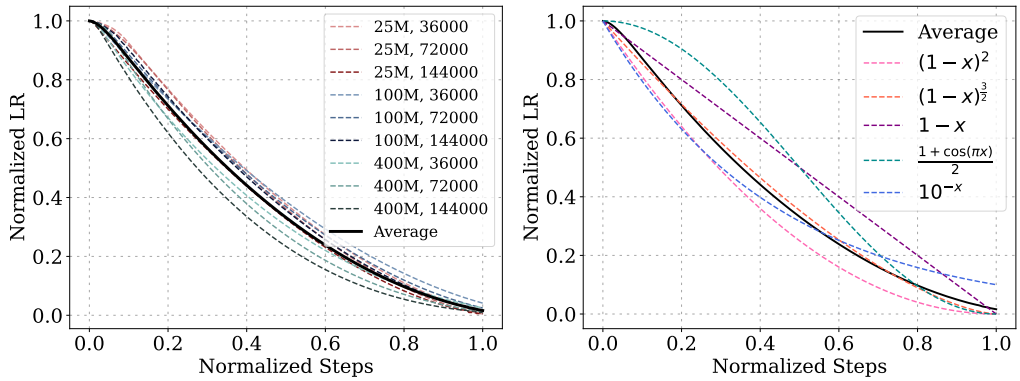


Figure 10: Approximation of decay functions.

B.2 Approximation of Decay Function

To facilitate the use of the optimized LR schedules and find out their decaying trends, we try to approximate the decay function of the WSD-like optimized schedules. Given the optimized schedules $S = \{t, \eta\}_{N,T}$, where N and T denote model sizes and training steps, we compute the normalized LRs and steps as follows:

$$\tilde{\eta} = \frac{\eta}{\eta_{\max}}, \ \tilde{t} = \frac{t - t_{\min}}{t_{\max} - t_{\min}}.$$

Then we average across N and T as shown in the left of Figure 10, computing an averaged LR schedule $\tilde{\boldsymbol{\eta}}_{\text{avg}}$. Then we utilize a symbolic regression approach to search for the approximate function form of the decay function. We get that $f(x) = (1-x)^{-\alpha}$ best capture the relation between $\tilde{\boldsymbol{t}}$ and $\tilde{\boldsymbol{\eta}}_{\text{avg}}$. In our experiments, $\alpha = \frac{3}{2}$ fits well and we show the average schedule against different candidates function form in the right of Figure 10. We use the WSD with the decay function f to train a 25M model with 24000 steps and 4000 decay steps. The result with a final loss of 3.274 slightly outperforms the WSD with exponential function (Hu et al., 2024) with a final loss of 3.276, but can not match the directly optimized schedule, which reaches below 3.269.

B.3 The Selection of Training Set

We conduct ablation experiments over the loss curves in the training sets, including two-stage, cosine, and constant LR schedules. We remove one of them and keep the other two as training sets. Then we fit over these subsets of the full training sets in the same approach with the multi-power law. The runs are over 25M models. The resulting coefficients are shown in Table 4 and the resulting test metrics are shown in Table 5. The test metrics are measured over the full test sets, including different schedule types and horizons. There are some observations as follows:

- As shown in Table 4, the coefficients of different fittings are consistent overall, while there are a few parameters that vary, like C. We conjecture there are some correlations between C and other parameters like γ. A refined form of multi-power law would be expected in future work.
- As shown in Table 5, the multi-power law shows its robustness over the training sets, with comparable performances between the full-set fitting and the subset-fitting results.

C Sanity Check on Derivation and Optimization

Sanity Check on Two(Multi)-Stage LR Schedule. We provide an empirical sanity check of our multi-power law in the case of the two-stage and multi-stage LR schedules.

From the perspective of coefficients of the multi-power law, in 25M experiments, we have the final fitted coefficients as follows: $A = 0.507$, $B = 446.4$, $C = 2.070$, $\alpha = 0.531$, $\beta = 0.406$,

Items	A	B	C	α	β	γ	L_0
Full	0.507	446.4	2.070	0.531	0.406	0.522	3.1
Cosine + 2-stage	0.5272	455.0	6.276	0.5032	0.3622	0.4172	3.147
Constant + 2-stage	0.5279	457.0	7.569	0.5067	0.3613	0.4002	3.149
Constant + Cosine	0.5292	477.4	0.854	0.5041	0.3189	0.6256	3.146

Table 4: Parameters for Different Fittings. "Full" denotes the fitting with the full training set, all three loss curves.

Model	$R^2 \uparrow$	MAE $\downarrow$	RMSE $\downarrow$	PredE $\downarrow$	WorstE $\downarrow$
Full	0.9975	0.0039	0.0046	0.0012	0.0040
Cosine + 2-stage	0.9971	0.0040	0.0046	0.0012	0.0048
Constant + 2-stage	0.9976	0.0037	0.0045	0.0011	0.0039
Constant + Cosine	0.9993	0.0020	0.0031	0.0006	0.0060

Table 5: Performance Metrics for Different Fittings.

$\gamma = 0.522$, $L_0 = 3.1$. It is noteworthy that the scales of the coefficients align with the experiments of two-stage and multi-stage cases, shown in Figure 4 and Figure 5.

From the perspective of experimental validation, the multi-power law should be applicable to two-stage cases and multi-stage cases. On the one side, the training set contains a two-stage LR schedule with $\eta_B = 9 \times 10^{-5}$, so our law could overfit the two-stage LR schedule loss along with other schedule losses. On the other side, we test the fitted law onto the two-stage learning rate schedules with η_B at 3×10^{-5} and 1.8×10^{-4}. As shown in Figure 2, the multi-power law can be extended to more extreme two-stage LR schedule cases. Moreover, the prediction of fitted law over the multi-stage schedules is presented in the right of Figure 11.

In this regard, the multi-power law could capture both the continuous and discontinuous LR schedule, which might shed light on the training dynamics of large language models.

Sanity Check for Optimized LR Schedule. In Figure 1, we present the optimized LR schedule based on the multi-power law. As a sanity check, we utilize the multi-power law to predict the loss curve based on the optimized LR schedule. We compare the predicted curve with the actual training curve in Figure 11. Overall, the prediction of multi-power law over the optimized LR schedules aligns with the ground truths. In this way, we could validate that the optimized LR schedules lie in the regime where the multi-power law holds. We verify that under the discussion above scenarios,

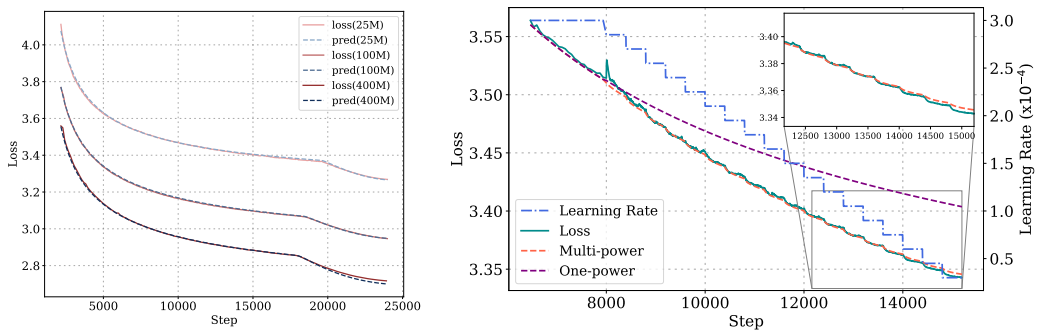


Figure 11: **Left:** The prediction of optimized LR schedule. The average maximum error ratio of loss curves is around 0.007. **Right:** The multi-power Law can predict the training curves with Multi-Stage LR schedules for 25M models. The multi-power law can be applied consistently over different stages of the schedule. The schedule consists of 19 stages. Each stage, except the initial one, consists of 400 steps. The LR reductions between adjacent stages are 1.5×10^{-5}, so in total, the learning rate decreases from 3×10^{-4} to 3×10^{-5}.

the performance improvement obtained through optimization is not overwhelmed by the noise errors introduced.

D Related Work

Optimal Learning Rate Schedule. Designing an effective learning rate schedule for deep learning has been a prominent research focus. 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) utilized reinforcement learning algorithms to learn a learning rate schedule adaptively. 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 scheduler. Geiping & Goldstein (2023) experimentally compared the performance differences of various learning rate schedules, concluding that quick annealing of the schedule aids in performance improvement. Recently, Hu et al. (2024) introduced a three-phase learning rate schedule with warm-up, stable, and decay phases, showcasing its superior performance across multiple datasets.

However, some of these papers focus on heuristically designing high-performance learning rate schedules without a comprehensible, principled approach to optimize the schedule. Some of the others try to optimize schedules within a function subspace. The effectiveness of the resulting function may be restricted by the subspace. 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). While our paper does not delve into the discussion about the model dimension scaling, we discuss the scaling along the data dimension along with the LR dimension. We believe it offers new perspectives and a novel starting point for theoretical investigations.

Hyperparameters Optimization. Hyperparameter optimization (HO) has long been a focal point of research within the machine learning community. 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. However, these works typically parameterized LR schedule as a learnable constant or a family of functions with learnable parameters, without fully exploring the potential of LR schedule. While this form of parameterization offers theoretical and experimental convenience, it often lacks interpretability. Furthermore, methods proposed in Teng et al. (2021); Jin et al. (2021) aim to adjust LR schedule during training automatically, but these approaches cannot identify the optimal LR schedule before training begins, and they fail to fully generalize across different datasets, underutilizing the scaling law information followed by the model. In contrast, Klein et al. (2022) selects hyperparameters based on differences in learning curves for various hyperparameters, while Kadra et al. (2024) recognizes the power law phenomenon and develops HP methods based on power law. However, we propose 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 laws, our understanding of the theoretical explanation and origins of scaling laws 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 is the first to provide 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.

E Limitations

Our study presents several limitations that warrant acknowledgment. Primarily, our investigation into the influence of hyperparameters on the loss curve is confined to the learning rate. Additionally, in our examination of the learning rate, we constrain the maximum learning rate to a predetermined value, owing to the complex relationship between the maximum learning rate and the loss. Moreover, we note the prediction bias in some extreme cases, as mentioned in Appendix A, requiring a more sophisticated formula to alleviate the error accumulation. We also note the parameter redundancy issue shown in Appendix B.3. Furthermore, our current work does not provide deep insights into the underlying mechanisms deriving this multi-power law. Without a more comprehensive theoretical analysis, our proposed law may be incomplete or merely match the surface form of the observed phenomena.

Nevertheless, despite these limitations, we contend that our contributions, including the schedule-aware loss curve prediction and optimized LR schedule, offer valuable insights into the dependency of loss-decaying on the LR schedules and the trade-off in schedule optimization. These advancements pave the way for a more nuanced understanding of training dynamics and potentially more efficient training strategies for large language models.

F Discussions of Multi-Power Law Derivation (Section 2)

F.1 Chinchilla Data Scaling Laws for the Final Loss Prediction.

Considering a fixed hyper-parameter setting, including the batch size, learning rate schedule, model type, previous work Muennighoff et al. (2023); Hoffmann et al. (2022) mainly follows the Chichilla Scaling Laws to extrapolate model size N and data quantity D: $\mathcal{L}(N, D) = L_0 + A \cdot D^{-\alpha} + B \cdot N^{-\beta}$. According to the form of law, the data scaling is roughly independent of the model scaling. Thus, we could focus on the data scaling and refer to $\mathcal{L}(T) = L_0 + A \cdot T^{-\alpha}$ as the Chinchilla Data Scaling Laws (CDSLs), where T denotes the training steps given the fixed batch size. The Chinchilla law is exclusively applicable to the final training loss because the Chinchilla law is LR-independent and the mid-training parts of loss curves commonly bear insufficient learning rate decay compared to the final loss (Hoffmann et al., 2022). As shown on the left of Figure 7, to extrapolate the final loss, we first need to generate several (typically more than 10 (Hoffmann et al., 2022; Dubey et al., 2024)) loss curves given a specific schedule (typically the Cosine schedule). Then we could fit CDSL over the final losses. Noticeably, the validation loss decreasing by 0.001 matters in the LLM scenario, because slight progress in loss may require intense computation practically, especially on a large scale. Moreover, the validation loss probably correlates with the emergent ability in downstream tasks. A little difference in the loss scale may indicate a steep deviation in the downstream performance (Du et al., 2024).

F.2 Motivation: Continuous Approximations of the Training Dynamics.

The rationale behind this approach is that matching the learning rate sum between the two training processes should result in similar training losses, and thus a more accurate approximation can be obtained by further exploring the loss reduction term $\mathrm{LD}(t)$. To see this, we take SGD as an example. In theory, if the learning rates used in training $\eta_1, \ldots, \eta_T$ are small, then SGD is known to be a first-order approximation of its continuous counterpart (Li et al., 2017; Cheng et al., 2020; Elkabetz

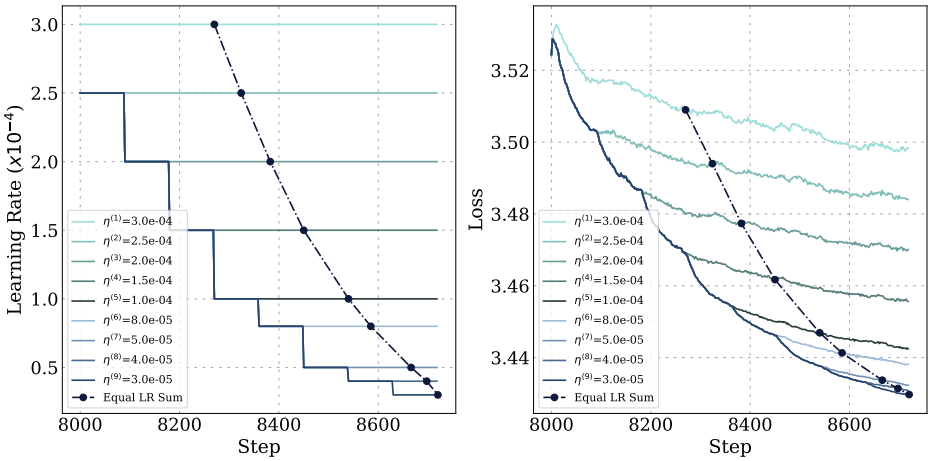


Figure 12: **Left:** Multi-stage schedule and interpolated LR schedules between the multi-stage LR schedule and the auxiliary schedule. There are 9 stages in our case, and the length of each stage, except the first one, is 90. The step points with the equal LR sum as the final step are marked in black and linked with the dash-point line. The learning rates before 8000 steps are constant at 3×10^{-4}. **Right:** Corresponding training curves for the actual multi-stage training curve, the auxiliary schedule as well as their interpolation.

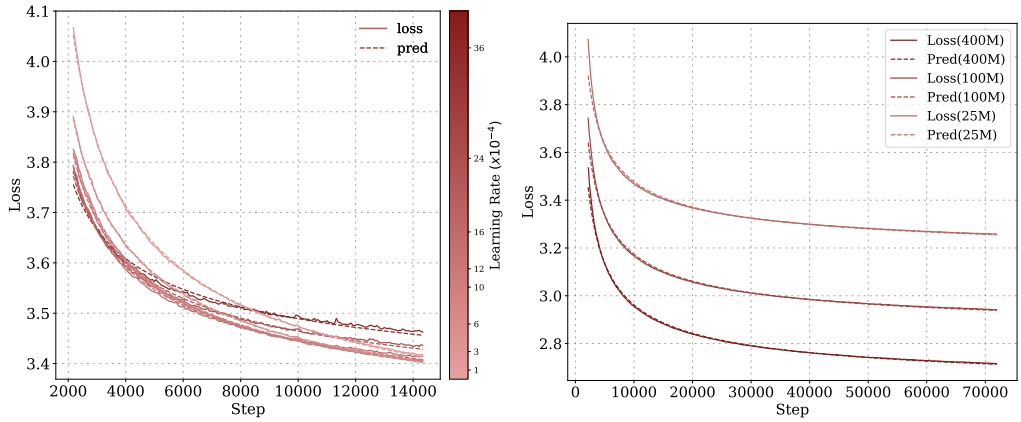


Figure 13: Loss curves trained with constant LR schedules. **Left:** The learning rates of schedules range from 3.0×10^{-4} to 3.6×10^{-3}. The total step number is 14400. The loss curves are all fit with Equation (5). The mean MSE is 1.55×10^{-5} and the mean coefficient of determination (R^2) is 0.9976. **Right:** Model sizes include 25M, 100M, and 400M. The total step number is 72000 and LR is 3.0×10^{-4}. The mean MSE is 8.04×10^{-5} and the mean R^2 is 0.9947.

& Cohen, 2021), gradient flow, which evolves the parameters $\boldsymbol{\theta}(\tau)$ according to the differential equation, $\frac{\mathrm{d}\boldsymbol{\theta}(\tau)}{\mathrm{d}\tau} = -\nabla\mathcal{L}(\boldsymbol{\theta}(\tau))$, where $\nabla\mathcal{L}(\boldsymbol{\theta})$ stands for the gradient of the loss function at $\boldsymbol{\theta}$, and τ is a continuous time variable. In this continuous approximation, the step t of SGD corresponds to the evolution of $\boldsymbol{\theta}(\tau)$ for a small continuous time interval of length η_t. When the LRs are extremely small, the parameter after t steps of SGD should be close to $\boldsymbol{\theta}(\tau)$ with $\tau = \sum_{k=1}^{t} \eta_k$. This motivates us to match the learning rate sum $\sum_{k=1}^{t} \eta_k$ between the two training processes to approximate the loss curve. This argument can be extended to other optimization algorithms, such as Adam (Ma et al., 2022). However, these continuous approximations can be loose for realistic LR schedules, thus necessitating a more detailed analysis of the loss reduction term $\mathrm{LD}(t)$.

F.3 Incorporating the Warmup Stage.

In practice, many learning rate schedules include a warmup stage where the learning rate gradually increases from zero to a peak value. To incorporate the effect of this stage, we can change the definition of auxiliary process to include the same warmup stage before using the constant learning rate. Then similar to the argument in Section 2.1, letting W be the sum of learning rates in the warmup stage, we can change the auxiliary loss formula to $\widehat{\mathcal{L}}_{\text{const}}(Z(t)) = L_0 + A \cdot (W + S_1(t))^{-\alpha}$. Our multi-power law then becomes

$$\mathcal{L}(T) \approx L_0 + A \cdot (W + S_1(t))^{-\alpha} - \sum_{k=2}^{t} B(\eta_{k-1} - \eta_k)(1 - (C\eta_k^{-\gamma} S_k(t) + 1)^{-\beta}). \tag{18}$$

This is the actual law we use in experiments since practical schedules often include a warmup stage.

G Two-Stage Experiments (Section 2.2)

In this section, we show the details of the investigation of the variation of coefficients of the power law of two-stage LR schedules.

Experiment Setting and Law Fitting. The default setting is $\eta_{\text{A}} = 3 \times 10^{-4}$, $\eta_{\text{B}} = 3 \times 10^{-5}$, $T_{\text{A}} = 8000$. In the ablation experiment, η_{A} ranges from 5×10^{-5} to 1×10^{-3}, η_{B} ranges from 4×10^{-5} to 2.9×10^{-4}, and T_{A} ranges from 4000 to 28000. The second stage lengths range from 1000 to over 6000. We follow Hoffmann et al. (2022) to utilize Huber loss as the objection function (Huber, 1992),

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

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 1×10^{-4}, and total steps of 20000. Here we do not conform to the L-BFGS algorithm due to the function form of $U(s)$. The parameters are initialized based on the estimation of asymptotic values of loss reduction and the slopes at the beginning of the second stage.

Fixed β Results. We propose a function form in Equation (6) to fit the loss reduction curve. The form guarantees the loss reduction zero when there is no LR reduction and fits with a power law. To explore the coefficient relation with the learning rate, we first investigate the coefficients fit in the ablation experiments. For the sake of further derivation and based on the fitted coefficients in the experiments, we fix the exponent β as LR-independent parameter 0.4. Then we re-fit the loss curves given $\beta = 0.4$ to validate the power form holds and further investigate the dependency of different parameters on the η_{A}, η_{B}, and T_{A}. The relation pattern is presented in Figure 4. Part of the two-stage schedules experiments are shown in Figure 14, including the ablations over the first stage as well as the second stage learning rates. Although β is fixed, the error margin is feasible for further derivation.

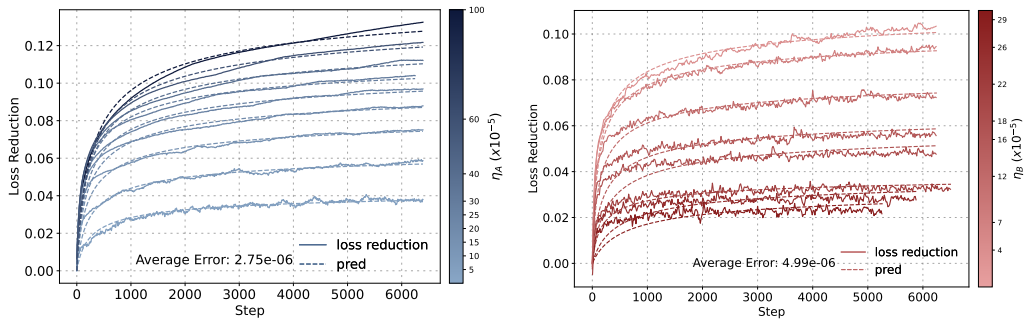


Figure 14: **Left:** Varying η_{A}, the loss reductions over steps x of two-stage LR schedules; **Right:** Varying η_{B}, the loss reductions over steps x of two-stage LR schedules.

H Details of Validation Experiments (Section 4)

Training Set, Test Set, and Model Training Settings. The validation experiments are framed as a machine learning task, where the loss curves in the training set are used to fit the multi-power law, which is then evaluated on the test set to report prediction accuracy.

The default training set consists of three loss curves: one trained with a cosine learning rate schedule, one with a constant learning rate schedule, and one with a two-stage learning rate schedule. The default test set includes unseen learning rate schedule types, loss curves over longer horizons, and more extreme two-stage learning rate schedules. Detailed descriptions of the training and test sets are provided in Table 6. Unless otherwise specified, the ending learning rate is set to $1/10$ of the peak learning rate (3×10^{-5} by default). The warmup phase spans 2,160 steps, but as the focus is on the post-warmup phase, only the post-warmup sections are used for fitting (Hu et al., 2024; Tissue et al., 2024).

The loss curves used in these experiments are generated from training the Llama2 model. The batch size is fixed at 128, and the sequence length is set to 4,096 across all configurations, resulting in 0.5M tokens per step. To simplify, data volume is described in terms of steps, where 10,000 steps consume 5B tokens. Validation loss is used as the default performance measure. Detailed model training hyperparameters are listed in Table 7, and a summary of the model series parameters used in the experiments is presented in Table 8.

Set	Schedule Type	Total Lengths	η_B/η_A
Training	Constant	24000	
	Cosine	24000	
	Two-stage	16000	0.3
Test	WSD	24000	
	WSDLD	24000	
	Two-stage	16000	0.1
	Two-stage	16000	0.6
	Constant	72000	
	Cosine	72000	

Table 6: Summary of training and test sets.

Default Hyperparameter	**Value**
Sequence Batch Size	128
Sequence Length	4096
Optimizer Type	AdamW
Beta1	0.9
Beta2	0.95
Epsilon	1×10^{-8}
Weight Decay	0.1
Gradient Clipping	1.0
Peak Learning Rate	3×10^{-4}
Final Learning Rate	3×10^{-5}
Warmup Steps	2160

Table 7: Hyperparameters related to model training.

Fit the Multi-power Law. Similar to the two-stage fitting, we utilize the Huber loss as the objective function (Huber, 1992),

$$\min_{\Theta} \sum_{t} \text{Huber}_{\delta}(\log \mathcal{L}_{\Theta}(t) - \log \mathcal{L}_{\text{gt}}(t)), \tag{19}$$

where $\Theta = \{A, B, C, \alpha, \beta, \gamma, L_0\}$, $\delta = 1 \times 10^{-3}$ and $\mathcal{L}_{\text{gt}}(t)$ denotes the ground truth of validation losses. We adopt the Adam optimizer, with a learning rate at 5×10^{-3} for the index parameters that

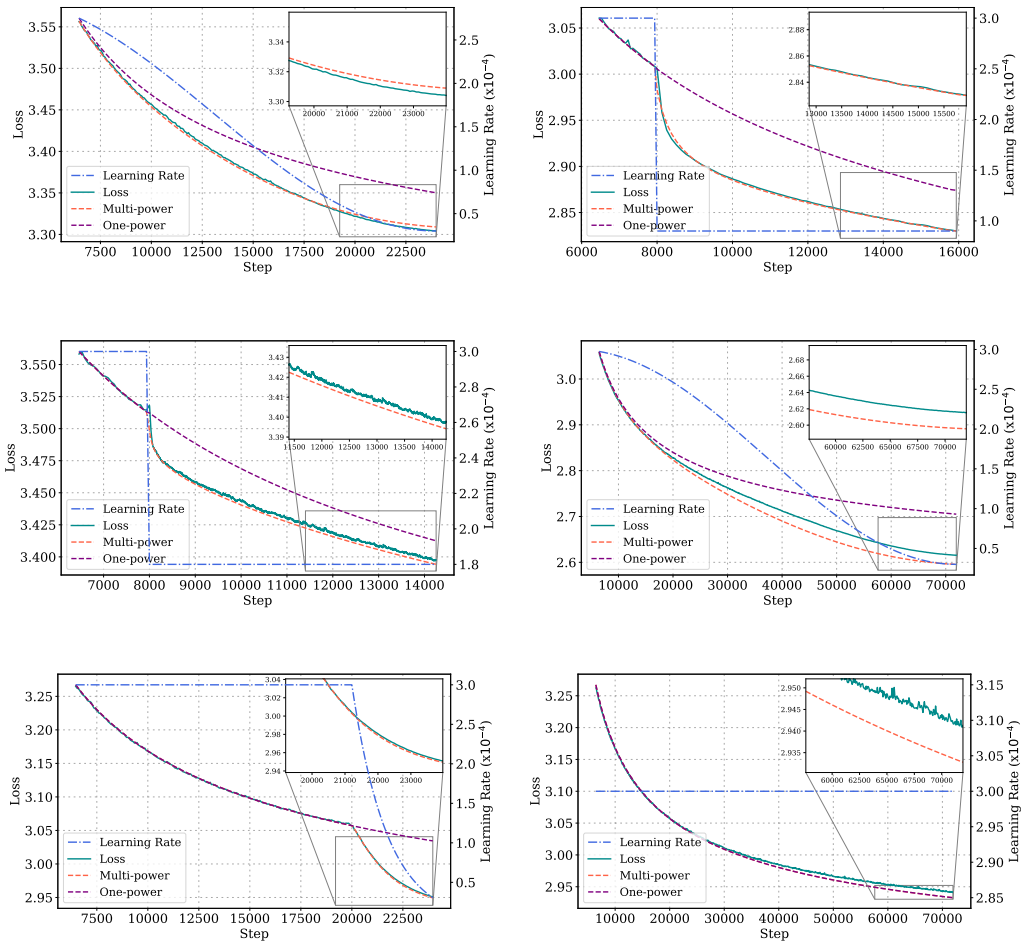


Figure 15: **Details of fitting and prediction.** The subfigures illustrate loss curve fitting (training set) and prediction (test set) for various configurations. (X, Y) indicates the subfigure at row X, column Y. The columns in the accompanying table describe: **F/P** for Fitting (F) or Prediction (P), **Model Size (M)**, **Step Length (S)**, and **Learning Rate Schedule (LRS)**. Details of each subfigure are provided below:

(X, Y)	F/P	Model Size (M)	Step Length (S)	LR Schedule (LRS)
(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

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	36	16	822	1026

Table 8: The model series used 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.

are α, β, and γ in our law, and 5×10^{-2} for the coefficient or constant parameters, that are A, B, C and L_0 in our law. We also take a learning rate at 1×10^{-5} and 1×10^{-6} with initialization as the previous fitting parameters. We select the result with lower training loss from the first optimization result and the second one. Each optimization takes over 5×10^4 steps. For 400M results, we find that $A = 0.658$, $B = 614.3$, $C = 0.164$, $\alpha = 0.421$, $\beta = 0.883$, $\gamma = 0.564$, $L_0 = 2.524$. For 100M cases, we have $A = 0.592, B = 521.4, C = 0.242, \alpha = 0.460, \beta = 0.604, \gamma = 0.647, L_0 = 2.792$. Part of fitting and prediction examples are shown in Figure 15.

Fit the Momentum Law. We mainly follow the approach proposed by Tissue et al. (2024). The objective function follows Equation (19) and we adopt the L-BFGS algorithm to minimize it. For a fair comparison, we grid search over its hyperparameter λ in $\{0.95, 0.99, 0.995, 0.999, 0.9995\}$ and select the best hyperparameter based on the fitting accuracy over the training set. We also evaluate the law over the full test set listed in Table 6. The prediction accuracy comparison between our multi-power law and the momentum law is shown in Table 5.

I Details of Optimized LR schedule (Section 5)

Details of Optimizing the Surrogate Objective. To make the optimization more stable, we define the following quantities $\boldsymbol{d\eta} := \{d\eta_1, d\eta_2, \ldots, d\eta_T\}$, where $d\eta_i := \eta_{i-1} - \eta_i$. Thus we can conduct optimization with an easier constraint over $\min_{d\eta} \tilde{\mathcal{L}}_{\hat{\Theta}}(\boldsymbol{d\eta})$. Notice that $\eta_i = \eta_0 - \sum_{k=1}^{i} d\eta_k$ and $\boldsymbol{\eta}$ is one-to-one with $\boldsymbol{d\eta}$. We can denote $\tilde{\mathcal{L}}_{\hat{\Theta}}(\boldsymbol{d\eta}) = \mathcal{L}_{\hat{\Theta}}(\boldsymbol{\eta})$. So now, instead of directly optimizing $\min_{\boldsymbol{\eta}} \mathcal{L}_{\hat{\Theta}}(\boldsymbol{\eta})$ in Equation (12), we can conduct optimization with an easier constraint over $\min_{\boldsymbol{d\eta}} \tilde{\mathcal{L}}_{\hat{\Theta}}(\boldsymbol{d\eta})$, which is,

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

In practice, we find we can solve the optimization problem with a relaxed constraint,

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

The optimized results $\boldsymbol{d\eta}$ also satisfy the constraint $\sum_{i=1}^{T} d\eta_i \leq \eta_0, \ \forall 1 \leq 1 \leq T$. The constraints are forced by clipping. Our optimization is applied to the law fitted over the training set mentioned in Appendix H. Regarding the optimization details, we use the Adam optimizer with a constant learning rate. The learning rate scale is searched ranging from 2×10^{-8} to 1×10^{-9} and the optimization step number ranges from 50000 to 200000 for better convergence.

Decay Ratio Details. Following Hu et al. (2024), we take the decay function of both exponential decay and linear decay. We grid search over $3000, 4000, 5000, 6000$ and 7000 to find that the best decay step number is 6000, with a total steps of 24000. The ending learning rate is set to $1/10$ of the peak learning rate following Hu et al. (2024). According to Figure 1, we find that the decay ratio of our optimized learning rate schedule aligns with the grid-searched WSD. The ending learning rate is lower than the empirical one and the decay shape is between linear and exponential functions.

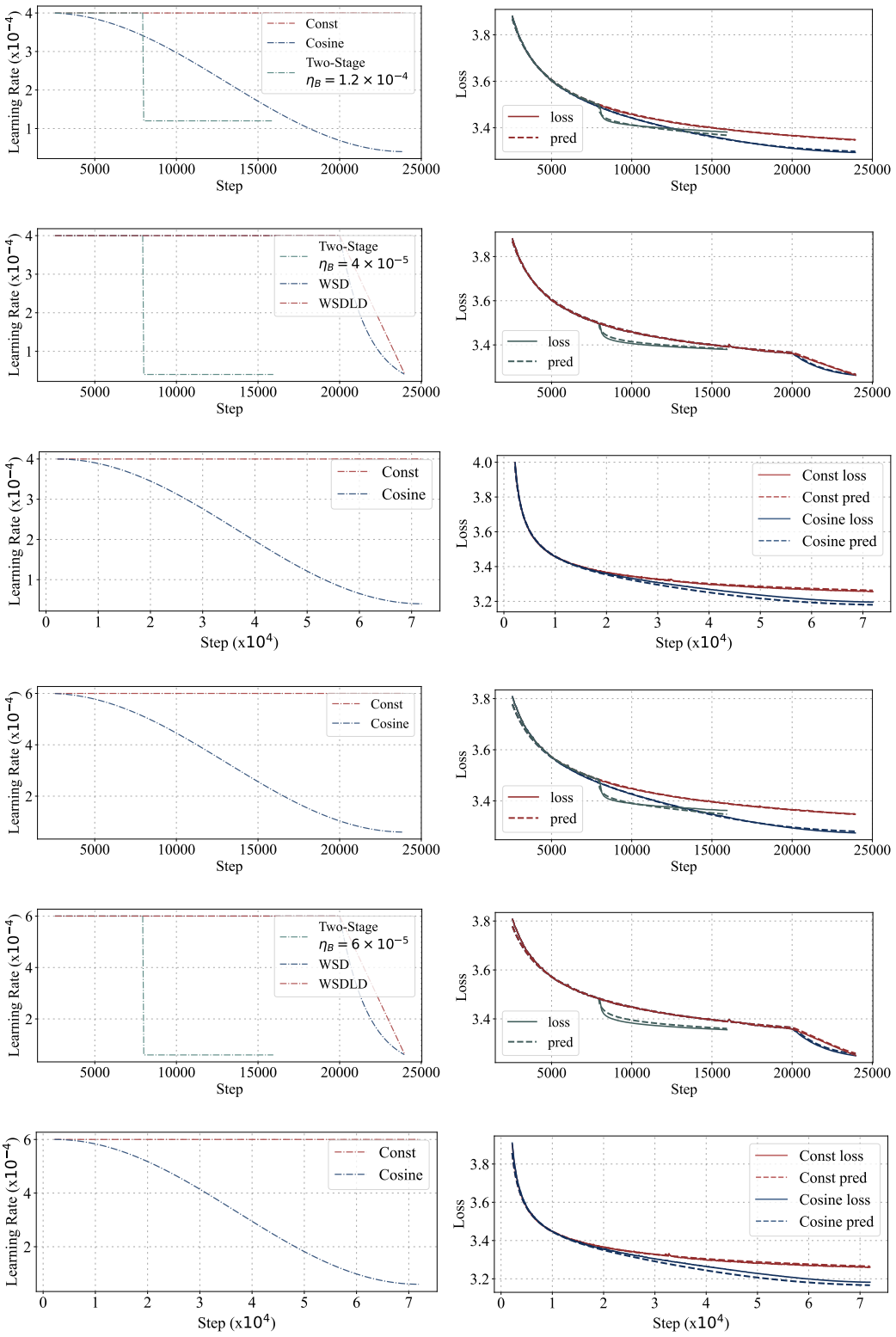


Figure 16: Ablation over peak learning rates. **Left:** the learning rates of the schedules; **Right:** the loss curves of schedules. **Updown:** the first three rows are the results for the peak learning rate at 4×10^{-4} and the last three rows are for the peak learning rate at 6×10^{-4}. For each set of the three rows, the first row shows the fitting on the training set, the second row shows the prediction over unseen schedules and the third row shows the extrapolation on a long horizon loss curve.

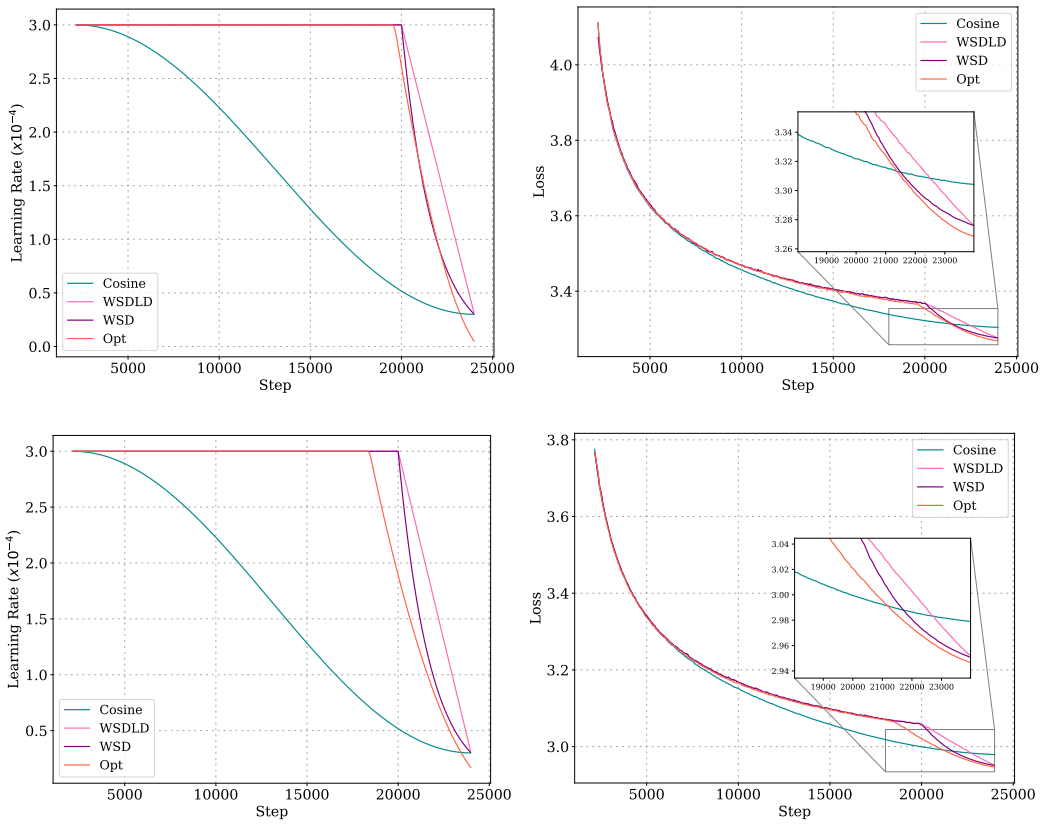


Figure 17: Our optimized LR schedules and their loss curve compared with Cosine, WSD, and WSDLD schedules. The total step number is 24000. The decay step number of WSD and its variant is 4000. **Upper:** 25M; **Lower:** 100M; **Left:** Learning rates over step; **Right:** Losses over step.

Optimized Schedule of Longer Horizons and Different Model Sizes Apart from the optimized LR schedules shown in Figure 1 and Figure 17, we further validate the optimized schedules of longer horizons and different model sizes. We optimize the LR schedules of 72000 steps based on the multi-power law fit over the training set. The training set conforms to the default setting only containing curves with lengths no longer than 24000, and we conduct experiments from 25M to 400M. As shown in Figure 18, the resulting schedules are also in the shape of WSD schedules, consisting of a stable phase and a decay phase. We compare the loss curves of the optimized LR schedules with those of commonly used Cosine LR schedules, we find that the optimized LR schedules outperform the Cosine LR schedules across different model sizes.

Zero-Ending Learning Rate Experiments. The optimized schedules consistently outperform WSD variants with "zero-ending" learning rates. As shown in Figure 19, we compare WSD(LD) variants with near-zero ending learning rates, the optimized schedules, and the original WSD(LD) schedules. In this experiment, the ending learning rate is set to 3×10^{-7}, which is 1/100 of the previous setting. Notably, a lower ending learning rate does not consistently lead to improved final loss. For example, the final loss of the WSD schedule increases with a near-zero ending learning rate. This suggests a complex interaction between the ending learning rate and the decay function, highlighting the challenges of jointly optimizing these hyperparameters in WSD schedules. In this context, the optimized schedule demonstrates its advantage by reducing the need for extensive hyperparameter tuning in WSD variants.

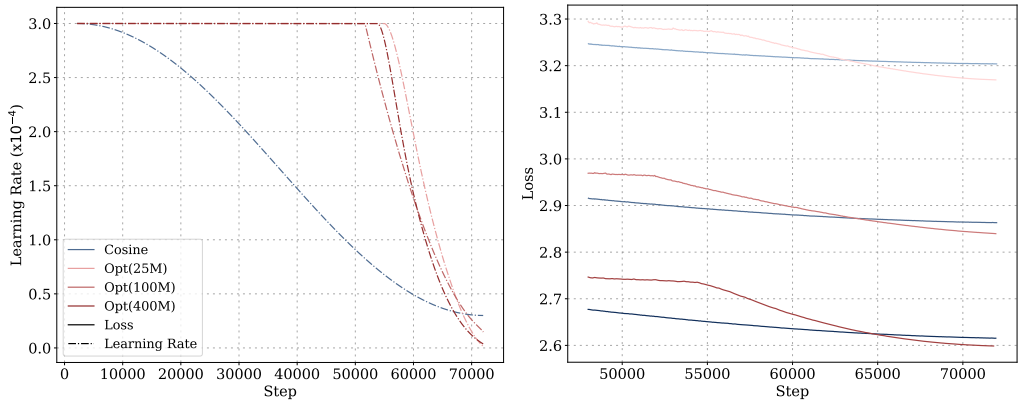


Figure 18: **Left:** optimized LR schedule vs Cosine LR schedule. The total step number is 72000, and model sizes range from 25M to 400M. **Right:** The loss curves of optimized schedules and Cosine schedules.

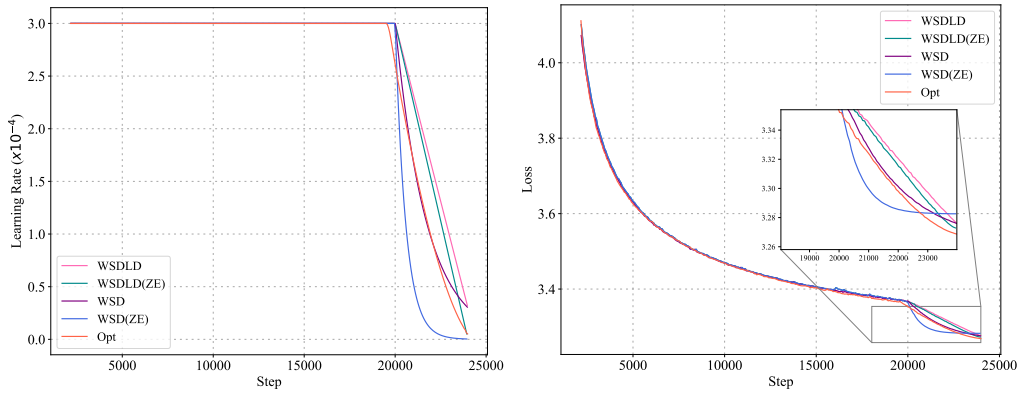


Figure 19: The comparison between the optimized schedules with the WSD variants with end LR as 0. WSD (ZE) and WSDLD (ZE) represent the WSD and WSDLD variants with ending learning rate as 3×10^{-7}, approximately "zero" ending LR compared to default 3×10^{-5}. **Left:** the learning rate comparison; **Right:** the loss comparison.

J Details of Discussion (Appendix A)

Model Types Ablation The multi-power law (MPL) is applicable across different model types, validated over GPT2 (Radford et al., 2019) and OLMo (Groeneveld et al., 2024) apart from the Llama2 (Touvron et al., 2023). For GPT2, the MPL is fitted on the constant and cosine schedules of 24000 steps. The fitted MPL can accurately predict the loss curve of the 72000-step cosine schedule, as shown on the right of Figure 21. Moreover, as shown on the left of Figure 21, even the simplified induced schedule (WSDSC, see Section 5.2) is superior to the WSD schedule and the commonly used cosine schedule.

Model Sizes Ablation The multi-power law (MPL) experiments are conducted on 1B models and 144B tokens and the downstream tasks are tested for the optimized schedules. The architecture matches 1B Llama3 (Dubey et al., 2024), with 32 heads and embedding dimension of 2048. The sequence batch size is extended to 512 and the total batch size is close to 2M. For the training stability, the peak learning rate is 2×10^{-4} across the experiments. The downstream tasks include LAMBADA (Paperno et al., 2016), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), and ARC-easy (Gu & Dao, 2023; Clark et al., 2018), following the Mamba (Gu & Dao, 2023) practice.

Peak Learning Rate Ablation In the previous discussion, we fix the peak learning rate at 3×10^{-4}. To validate the scope of application for multi-power law, we run experiments over different peak

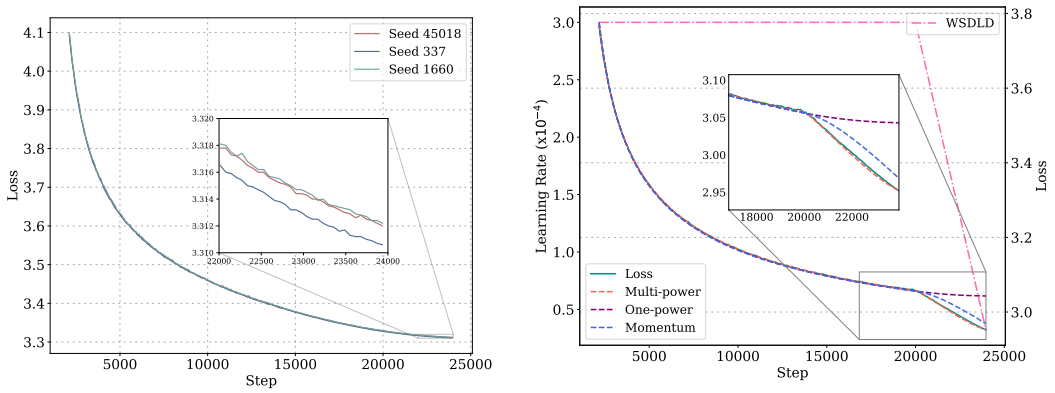


Figure 20: **Left:** The experiments on 25M and 24000 steps of different seeds. The standard variance of final loss is 0.0007 and the max gap is 0.0014. **Right:** Comparison between multi-power law and momentum law. In the decay stage, the multi-power law not only presents higher accuracy to fit the loss curve but also aligns with the curvature of the curve. As a comparison, the momentum law can also fit the loss in the stable stage, but it predicts a counterfactual concave curve in the decay stage.

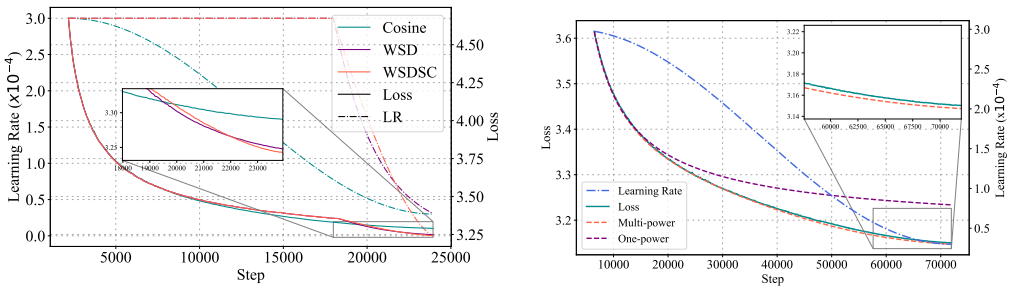


Figure 21: The loss curves of GPT2 models. The multi-power law is fit over 24000-step constant and cosine schedule losses. **Left:** The comparison between the cosine, WSD, and WSDSC (see Section 5.2) schedules; **Right:** Prediction on the 72000-step loss curve of cosine schedule.

learning rates. As shown in Figure 4, the empirical law of two-stage LR schedules deviates when the peak learning rate increases. Therefore, we run the experiments over the cases where the peak learning rates are 4×10^{-4} and 6×10^{-4}. The training set and the test set of loss curves conform to the setting of peak learning rate 3×10^{-4} except the peak LR. The two-stage case in the training set has $\eta_B = 0.3\eta_A$ while the two-stage cases in the test set have $\eta_B = 0.1\eta_A$ and $\eta_B = 0.6\eta_A$ respectively. The results are shown in Figure 16. The multi-power law reaches average R^2 value at 0.9965 in 4×10^{-4} case and 0.9940 in 6×10^{-4} case, exhibiting high accuracy in general. However, we note that there are over-underestimations in the $\eta_B = 0.6\eta_A$ two-stage case and the long-horizon Cosine case. We conjecture that as the peak learning rate increases, the error introduced by our approximation of the two-stage case law increases, and so does the error of loss reduction.

Batch Size Ablation We conduct ablation experiments on sequence batch sizes of 64 and 256 to validate our method. The default sequence batch size in our experiments is 128, with a sequence length of 4096, resulting in a default total batch size of approximately 0.5M tokens. Batch size is a critical parameter that influences the training process, particularly the peak learning rate. The chosen sequence batch sizes of 64 and 256 correspond to total batch sizes of 0.25M and 1M tokens, respectively. These experiments are performed using 25M models, following the procedure outlined in Appendix H. As illustrated in Figure 22, the multi-power law (MPL) consistently demonstrates high predictive accuracy, with R^2 values exceeding 0.9970 across both cases. Additionally, for 1B model experiments, the batch size is set to 512. While the coefficients of MPL are batch size-dependent, the functional form of MPL remains robust across varying batch size configurations.

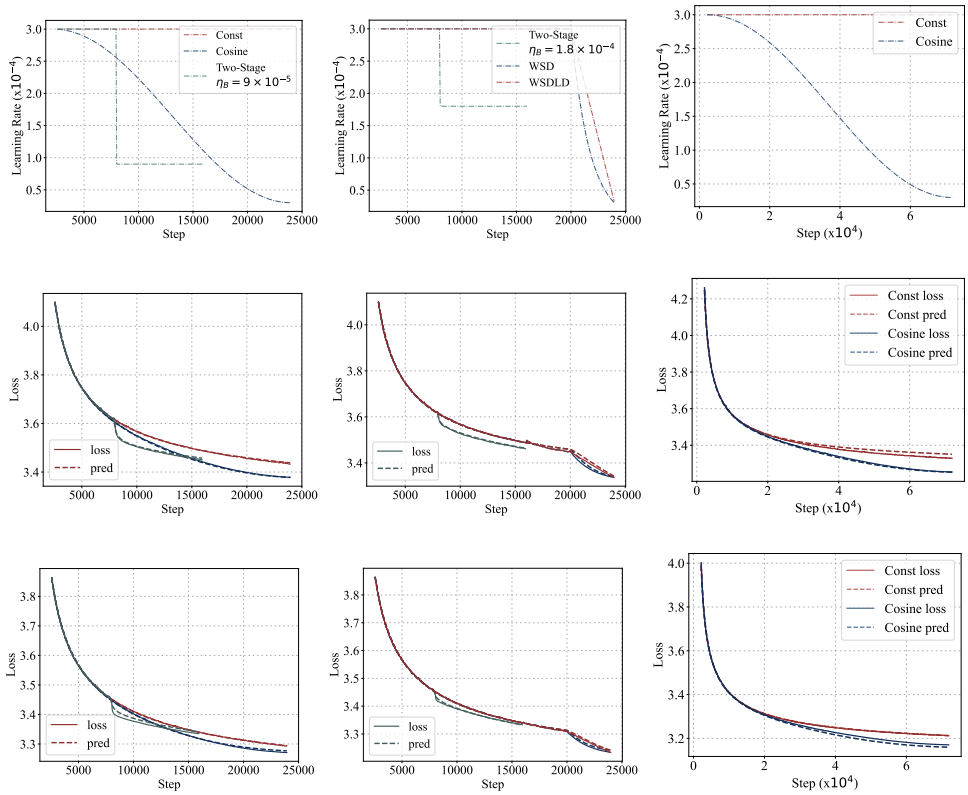


Figure 22: Ablation study on batch sizes. The R^2 values for batch sizes of 64 and 256 are 0.9977 and 0.9973, respectively. **Row One:** Learning rate schedules. **Row Two:** Loss curves for experiments with a sequence batch size of 64. **Row Three:** Loss curves for experiments with a sequence batch size of 256. **Column One:** Training set results. **Column Two:** Test set results, focusing on loss curves with the same horizon as the training set. **Column Three:** Test set results, focusing on loss curves with an extended horizon.

K Recover the Optimal Learning Rate Schedule for Momentum Law

In Section 5, by optimizing the multi-power law proposed in this paper, we greedily determine the learning rate at each step that allows the loss curve to converge most rapidly, resulting in a faster-converging learning rate. Experimentally, this optimized learning rate unsurprisingly converged to a smaller training loss, outperforming the cosine learning rate schedule and common WSD schedules. Meanwhile, recent concurrent work Tissue et al. (2024) has introduced a scaling law with a momentum term that can predict loss curve under various learning rate schedules, which can be written as

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

where $S_1 = \sum_{i=1}^{s} \eta_i$ and $S_2 = \sum_{i=1}^{s} \sum_{k=1}^{i} (\eta_{k-1} - \eta_k) \cdot \lambda^{i-k}$. λ is hyper-parameter typically ranges from 0.99 to 0.999. L_0, A, C are undetermined positive constants.

Similar to Section 5, here we could also optimize this law to get a learning rate schedule with faster convergence as

$$\begin{aligned} &\min_{\eta_1,\eta_2,\ldots,\eta_s} L_{\Xi}(\eta_1, \eta_2, \ldots, \eta_s) \\ &s.t.,\ 0 \le \eta_i \le \eta_{i-1},\ \forall 1 \le i \le T, \\ &\eta_i \le \eta_0, \end{aligned} \tag{A}$$

where Ξ represents the hyper-parameter and undetermined constants in $L(s)$, which is fixed in our setting to optimize $\eta_1, \ldots, \eta_s$. And for simplicity of derivation, we introduce η_0 in front of $\boldsymbol{\eta}$ as the

maximal LR. Compared with multi-power law, this optimization problem is obviously convex, so we could get its minimizer in theory. Surprisingly, the optimal learning rate schedule for this law is a two-stage learning rate schedule, with learning rate $\eta_2 = 0$ in the second stage. In conclusion, we get a quite trivial optima learning rate schedule for momentum law in (Tissue et al., 2024). This shows the superiority of our multi-power law over theirs. Next, we will formalize the above arguments mathematically.

Theorem 2. *For $s \to \infty$, the optimal learning rate schedule $\{\eta_0, \eta_1, \ldots, \eta_s\}$ which minimize the optimization problem (A) is that*

$$\underbrace{\eta_0 \to \cdots \to \eta_0}_{\textit{of number } i+1} \to \underbrace{0 \to \cdots \to 0}_{\textit{of number } s-i}.$$

proof.

Firstly, we reparameterize (A) as

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

where $\Delta_i := \eta_{i-1} - \eta_i$. Then we write out the $\hat{L}_{\Xi}(\Delta_1, \Delta_2, \ldots, \Delta_s)$

$$\hat{L}_{\Xi}(\Delta_1, \Delta_2, \ldots, \Delta_s) = L_0 + A \cdot (s\eta_0 - \sum_{j=1}^{s} \sum_{i=1}^{j} \Delta_i)^{-\alpha} - C \cdot \sum_{j=1}^{s} \sum_{i=1}^{j} \Delta_i \lambda^{j-i}.$$

We take the partial derivative of L with respect to Δ_i

$$\begin{aligned}
\frac{\partial L}{\partial \Delta_i} &= \alpha A \mathbf{\Phi}^{-\alpha-1} \cdot (s - i + 1) - C \cdot (\lambda^0 + \lambda^1 + \cdots + \lambda^{s-i}) \\
&= \alpha A \mathbf{\Phi}^{-\alpha-1} \cdot (s - i + 1) - C \cdot \frac{1 - \lambda^{s-i+1}}{1 - \lambda},
\end{aligned}$$

where $\mathbf{\Phi} := (s\eta_0 - \sum_{j=1}^{s} \sum_{i=1}^{j} \Delta_i)$. Setting $\frac{\partial L}{\partial \Delta_i} = 0$, then we have

$$\mathbf{\Phi}^{-\alpha-1} = \frac{C}{\alpha A} \frac{1 - \lambda^{s-i+1}}{(1 - \lambda)(s - i + 1)}.$$

Notice that by the definition of $\mathbf{\Phi}$, it's invariant with i, but here we write $\mathbf{\Phi}$ as a function of i. We also know from KKT condition that $\frac{\partial L}{\partial \Delta_i} = 0$ *or* $\delta_i = 0$ is satisfied for all $i \in \{1, 2, \ldots, s\}$. So now we can get a lemma below

Lemma 1. *There exists at most 1 index $i \in \{1, 2, \ldots, s\}$ such that $\Delta_i \neq 0$.*

proof.

Assume that there are $i \neq j$, and $\Delta_i \neq 0$ *and* $\Delta_j \neq 0$, then $\frac{\partial L}{\partial \Delta_i} = \frac{\partial L}{\partial \Delta_j} = 0$. Further we have

$$\mathbf{\Phi}^{-\alpha-1} = \frac{C}{\alpha A} \frac{1 - \lambda^{s-i+1}}{(1 - \lambda)(s - i + 1)} = \frac{C}{\alpha A} \frac{1 - \lambda^{s-j+1}}{(1 - \lambda)(s - j + 1)}.$$

Since $i, j \in \mathbb{N}^+$, $\frac{C}{\alpha A} \frac{1-\lambda^{s-i+1}}{(1-\lambda)(s-i+1)}$ is impossible to get the same output from two different positive integers. Thus $i = j$. □

Now assume that $\Delta_i \neq 0$, then the lemma below implies the results in Theorem

Lemma 2. *For $s \to \infty$, and all $i \in \{1, 2, \ldots, s\}$, it holds that*

$$\mathbf{\Phi}^{-\alpha-1} \neq \frac{C}{\alpha A} \frac{1 - \lambda^{s-i+1}}{(1 - \lambda)(s - i + 1)}.$$

proof.

We prove this lemma by showing that the left-hand side is a $o(\frac{1}{s})$ asymptotic and the right-hand side is $\omega(\frac{1}{s})$, so that when $s \to \infty$, the left side will always less than the right side, then our proof ends. Next, we show that the right-hand side is $\omega(\frac{1}{s})$. We first let $t := s - i + 1 \in \{1, 2, \ldots, s\}$. We notice that $\frac{1-\lambda^t}{1-\lambda} \in [1, \frac{1}{1-\lambda}]$, so we have

$$\begin{aligned}\frac{C}{\alpha A}\frac{1-\lambda^{s-j+1}}{(1-\lambda)(s-j+1)} &\geq \frac{C}{\alpha A}\frac{1}{s-j+1} \\ &= \omega(\frac{1}{s}).\end{aligned}$$

Then we show that the left-hand side is an $o(\frac{1}{s})$ asymptotic. We consider a two-stage learning rate schedule such that the 1-stage learning rate is $\eta_{\max}$, the two-stage learning rate is 0, and the two-stage lasts for time $10\log_\lambda \epsilon$. Since $\mathbf{\Phi}$ is the minimizer of L, so we have inequality

$$\begin{aligned}L_0 + A\cdot\mathbf{\Phi}^{-\alpha} - C\cdot\sum_{j=1}^{s}\sum_{i=1}^{j}\Delta_i\lambda^{j-i} &\leq L_0 + A\cdot[(s-10\log_\lambda\epsilon)]^{-\alpha} - c\cdot\frac{1-\epsilon^{10}}{1-\lambda}\eta_{\max} \\ A\cdot\mathbf{\Phi}^{-\alpha} &\leq A\cdot[(s-10\log_\lambda\epsilon)]^{-\alpha} + c\cdot\frac{\epsilon^{10}}{1-\lambda}\eta_{\max},\end{aligned}$$

where in the second inequality we use the property that $\sum_{j=1}^{s}\sum_{i=1}^{j}\Delta_i\lambda^{j-i} \leq \frac{\eta_{\max}}{1-\lambda}$. We set $\epsilon = [(s-10\log_\lambda\epsilon)]^{-\alpha}$, then we have

$$\begin{aligned}\mathbf{\Phi}^{-\alpha} &\leq \epsilon + \frac{C}{A}\cdot\frac{\epsilon^{10}}{1-\lambda}\eta_{\max} \\ &= O(\epsilon) \\ &= O\left((s)^{-\alpha}\right)\end{aligned}$$

Thus we have

$$\mathbf{\Phi}^{-\alpha-1} = O(\frac{1}{s^{1+\alpha}}) = o(\frac{1}{s}).$$

Then we get the results in Lemma 2. □

According to KKT condition, for all $i \in \{1, 2, \ldots, s\}$, it holds that

$$\mathbf{\Phi}^{-\alpha-1} = \frac{C}{\alpha A}\frac{1-\lambda^{s-i+1}}{(1-\lambda)(s-i+1)} \ \ or \ \ \sum_{i=1}^{s}\Delta_i = \eta_0.$$

So if the i we choose for $\Delta_i \neq 0$ can't satisfy the first condition, then it should satisfy the second one, which is equivalent to that $\Delta_i = \eta_0$. Thus we get the optimal learning rate schedule for momentum law, which is

$$\underbrace{\eta_0 \to \cdots \to \eta_0}_{\text{of number } i+1} \to \underbrace{0 \to \cdots \to 0}_{\text{of number } s-i}.$$

□

L Proof of Theorem 1

To prove the whole theorem, we first treat all λ_i and Σ_{ii} as constants, 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 $\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}_T)]$:*

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

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

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

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 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 $A_0, A_1, \ldots, A_T$:

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

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 lemma.

Lemma 3. *If* $x \in [0, 1]$*, then*

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

Proof. The above inequalities hold for $x = 0$. For $x \in (0, 1]$, we have $\frac{1-(1-2x)\exp(2x)}{x^2} \geq \frac{1-(1-x)^2\exp(2x)}{x^2} \geq \frac{1-\exp(-2x)\exp(2x)}{x^2} = 0$ since $1 - 2x \leq (1-x)^2 \leq \exp(-2x)$. Also note that $\frac{1-(1-2x)\exp(2x)}{x^2}$ is an increasing function of x. So we have $\frac{1-(1-2x)\exp(2x)}{x^2} \leq \frac{1-(-1)\cdot\exp(2)}{1^2} \leq 10$. □

Lemma 4.

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

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

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

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

$$\begin{aligned}\sum_{t=1}^{k-1}\eta_t \exp(-2\lambda_i S_{t+1}) &= \sum_{t=1}^{k-1}\eta_t \exp(-2\lambda_i S_t)\exp(2\lambda_i \eta_t) \\ &\leq \sum_{t=1}^{k-1}\eta_t \exp(-2\lambda_i S_t)\exp(2) \\ &\leq \frac{\exp(2)}{2\lambda_i}\exp(-2\lambda_i S_k).\end{aligned}$$

It completes the proof. □

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

Lemma 5. *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 ϵ_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

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

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:

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

Since $\eta_k \lambda_i \in [0, 1]$ for all i, by Lemma 3, 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 $\mathbb{E}_{\boldsymbol{g}_k \sim \mathcal{N}(\boldsymbol{H}\boldsymbol{\theta}_{k-1}, \boldsymbol{\Sigma})}[U(\boldsymbol{\theta}_{k-1} - \eta_k \boldsymbol{g}_k, \eta_k, S_{k+1}) \mid \boldsymbol{\theta}_{k-1}]$ as

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

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

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

Taking the expectation 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 6. *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

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

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 4. □

Lemma 7. *In the setting of Lemma 5, we can bound the sum of the error terms ϵ_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 $|\epsilon_k|$,

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

For $\mathcal{E}_{1,i}$, we apply Lemma 6 and have

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

where the last inequality uses Lemma 4. For $\mathcal{E}_{2,i}$, we have

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

□

Now we are ready to prove Theorem 3.

Proof for Theorem 3. Using Lemma 5 and Lemma 7, 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 ϵ 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, to prove Theorem 1, we take the expectation of $M(\boldsymbol{\theta}, E)$ over all λ_i and Σ_{ii} as

$$\begin{aligned}
\mathbb{E}[M(\boldsymbol{\theta}_0, E)] =& \frac{1}{2}\|\boldsymbol{\theta}_0\|_2^2\mathbb{E}[\lambda\exp(-2\lambda S_1)] + \frac{d}{4}\eta_{\max}\mathbb{E}[\Sigma] - \frac{d}{4}\eta_{\max}\mathbb{E}[\Sigma\exp(-2\lambda S_1)] \\
&- \frac{d}{4}\sum_{k=2}^{T}(\eta_{k-1}-\eta_k)\left(\mathbb{E}[\Sigma] - \mathbb{E}[\Sigma\exp(-2\lambda S_k)]\right) \\
=& \frac{1}{2}\|\boldsymbol{\theta}_0\|_2^2\mu\frac{1}{Z_\lambda}\int_0^D \lambda^{\alpha+1}\exp(-2\lambda S_1)\mathrm{d}\lambda + \frac{d}{4}\eta_{\max}\mu - \frac{d}{4}\eta_{\max}\mu F\frac{1}{Z_\lambda}\int_0^D \lambda^{\alpha+\rho}\exp(-(2S_1+G)\lambda)\mathrm{d}\lambda \\
&- \frac{d\mu}{4}\sum_{k=2}^{T}(\eta_{k-1}-\eta_k)(1 - F\frac{1}{Z_\lambda}\int_0^D \lambda^{\alpha+\rho}\exp(-(2S_k+G)\lambda)\mathrm{d}\lambda) \\
=& \frac{\|\boldsymbol{\theta}_0\|_2^2\mu\gamma(\alpha+2,D)}{2^{\alpha+3}Z_\lambda}S_1^{-\alpha-2} + \frac{d}{4}\eta_{\max}\mu - \frac{d\eta_{\max}\mu F\gamma(\alpha+\rho+1,D)}{2^{\alpha+\rho+3}Z_\lambda}(S_1+\frac{G}{2})^{-\alpha-\rho-1} \\
&- \frac{d\mu}{4}\sum_{k=2}^{T}(\eta_{k-1}-\eta_k)\left(1 - \frac{F\gamma(\alpha+\rho+1,D)}{G^{\alpha+\rho+1}Z_\lambda}(\frac{2}{G}S_k+1)^{-\alpha-\rho-1}\right)
\end{aligned}$$

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_λ 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

$$\begin{aligned}
&\tilde{M}(\boldsymbol{\theta}_0, E) := \mathbb{E}[M(\boldsymbol{\theta}_0, E)] \\
&\quad = L_0 + AS_1^{-\alpha-2} - R(S_1+\frac{1}{C})^{-\alpha\rho-1} - \sum_{k=2}^{T}B(\eta_{k-1}-\eta_k)\left(1-(CS_k+1)^{-\alpha-\beta-1}\right).
\end{aligned}$$

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

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

So we complete the proof of Theorem 1.

L.1 Proof of Corollary 1

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

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

It completes the proof.