Hop, skip, jump to Convergence: Dynamics of Learning Rate Transitions for Improved Training of Large Language Models

Shreyas Subramanian
Amazon.com
subshrey@amazon.com

Vignesh Ganapathiraman
Amazon.com
vignesga@amazon.com

Corey Barrett
Amazon.com
corbarre@amazon.com

Abstract

Various types of learning rate (LR) schedulers are being used for training or fine tuning of Large Language Models today. In practice, several mid-flight changes are required in the LR schedule either manually, or with careful choices around warmup steps, peak LR, type of decay and restarts. To study this further, we consider the effect of switching the learning rate at a predetermined time during training, which we refer to as “SkipLR”. We model SGD as a stochastic gradient flow and show that when starting from the same initial parameters, switching the learning rate causes the loss curves to contract towards each other. We demonstrate this theoretically for some simple cases, and empirically on large language models. Our analysis provides insight into how learning rate schedules affect the training dynamics, and could inform the design of new schedules to accelerate convergence.

1 Introduction

Modern deep neural networks have achieved state-of-the-art performance across a wide range of machine learning tasks. One critical hyperparameter that significantly influences the training dynamics is the learning rate (LR). Aside from using adaptive optimizers such as Adam (Kingma and Ba, 2014), AdamW (Loshchilov and Hutter, 2016), and Adafactor (Shazeer and Stern, 2018), LR schedulers have become a necessity for training large language models (LLMs) (Zhao et al., 2023). Despite being used pervasively, the choice of LR schedules is often made based on empirical observations or best practices, rather than any known theoretical basis. Table 1 presents various optimizer-scheduler combinations used in recent large language models (LLMs), highlighting this pattern. This issue is addressed in part by adaptive learning rate schedulers, such as the (Baydin et al., 2017; Subramanian and Ganapathiraman, 2023). While these combinations are frequently chosen based on experience, best practices, or trial and error, their importance for overall training progress cannot be understated. In practice, drastic changes on top of the LR scheduler are often necessary to achieve optimal performance. For instance, in the training of OPT models, several mid-flight, drastic changes were required for convergence (Zhang et al., 2022). For the Llama 3 405B model, batch size (and effectively LR) is doubled once after pre-training with 252M tokens, and doubled again after pre-training with 2.87T tokens to avoid training divergence.(Dubey et al., 2024) A recent survey on large language models (LLMs) also highlights a common practice of using the AdamW optimizer with a scheduler that includes a warmup phase followed by a gradual decay to 10% of the initial maximum learning rate (Zhao et al., 2023). This initial warmup, followed by a sudden shift to a decay mode, introduces significant changes in the early stages of convergence.

Why are such abrupt changes in LR required to achieve peak performance, and how do they impact training? In this paper, we investigate these questions by studying the impact of drastic LR changes on the training of language models. Our theoretical and empirical analyses elucidate the dynamics of optimizers under varying LR schedules, focusing on abrupt LR transitions at predetermined epochs, which we term the “SkipLR” experiment.

First, we aim to understand the impact of LR schedules on training by utilizing the SkipLR framework to study the relationship between multiple LR transitions and changes in loss. While traditional schedulers emphasize the long-term effects of continuous changes in the LR, we argue that even instantaneous changes in LR during a single step can have long-term impacts on loss. We clarify here that SkipLR is a framework to study this specific phenomenon, and not a scheduler to be used in place of a scheduler like Cosine or exponential decay. Through theoretical analysis and hundreds

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 16349–16362
November 12-16, 2024 ©2024 Association for Computational Linguistics

Model	Optimizer	Initial LR	Scheduler	Citation
Falcon 40B	AdamW	$1.85e^{-4}$	Cosine decay to 10% Initial LR	(Almazrouei et al., 2023)
Meta Llama 2	AdamW	$1.5e^{-4}$	Cosine decay to 10% Initial LR	(Touvron et al., 2023)
Flan T5, PaLM	Adafactor	$5e^{-4}$	Constant	(Chung et al., 2022)
GPT-J	AdamW	$1.2e^{-4}$	Polynomial decay to 10% Initial LR	(Wang and Komatsuzaki, 2021)
OPT 1.3B	AdamW	$2e^{-4}$	Warmup from 0 to Initial LR followed by decaying down to 10% of the maximum LR over 300B tokens along with a number of mid-flight changes	(Zhang et al., 2022)
OPT 6.7B	AdamW	$1.2e^{-4}$		
OPT 30B	AdamW	$1e^{-4}$		
OPT 66B	AdamW	$0.8e^{-4}$		
OPT 175B	AdamW	$1.2e^{-4}$		
InstructGPT	Adam	$9e^{-6}$	Cosine decay to 10% Initial LR	(Ouyang et al., 2022)
Llama 3.1 70B	AdamW	$1.5e^{-4}$	Warmup 2000 steps with Cosine decay	(Dubey et al., 2024)
Llama 3.1 405B	AdamW	$8e^{-5}$	Warmup 8000 steps with Cosine decay to 1% of peak LR over 1.2M steps	(Dubey et al., 2024)

Table 1: Survey of recent LLMs, with optimizer and schedulers used

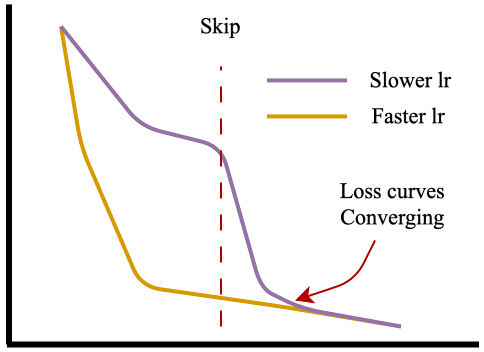


Figure 1: An illustration of the SkipLR phenomenon.

of experiments, We show the unintuitive ranking of loss curves corresponding to different learning rates for real transformer experiments compared to what is seen in theory, faster than asymptotic contraction from one arbitrary loss curve to another, as well as the high probability of transitioning to another loss curve corresponding to a different LR. This is a useful feature when designing effective schedulers in follow-on work that could speed up experimental runs significantly.

To demonstrate this, we first investigate how SkipLR transitions affect the descent on the loss surface at any given iteration. We model SGD dynamics as simple yet functionally correct continuous-time gradient flow, with $\frac{d\theta}{dt} = -\eta \nabla \Phi(\theta(t))$, where $\Phi(\theta)$ denotes the loss function. Our theoretical results show that, under common assumptions, loss function trajectories contract towards each other after a SkipLR transition as illustrated in Figure 1. This suggests that SkipLR techniques could help escape sharp local minima or transition between high and low loss regions during training in a principled manner.

Next, to support our theoretical findings, we present extensive numerical simulations visualizing loss curve contractions on synthetic functions. Specifically, for quadratic objectives, these simulations validate our analysis by demonstrating the contraction of loss curves following skips in strongly convex functions. We also demonstrate successful SkipLR transitions on large transformer networks for causal language modeling, translation, and named entity recognition (NER) tasks, empirically showing convergence in these models.

In this work, through our comprehensive analysis and experimentation with SkipLR, we aim to provide valuable insights that can guide informed decisions on learning rate schedules and adjustments during training. By quantifying the impact of abrupt LR changes and exploring their interaction with the loss landscape and gradient dynamics, we hope to enhance the design of adaptive learning rate schedules tailored to specific models and training conditions.

2 Related Work

Learning rate schedulers, such as linear or cosine scheduling (Loshchilov and Hutter, 2016), have been extensively studied and shown to accelerate convergence and improve training performance in neural networks. For instance, (Nakkiran, 2020) discusses the effect of learning rate annealing on generalization performance in convex learning problems, demonstrating that annealing leads to better generalization compared to using a constant small learning rate. This improvement is attributed to the mismatch between the test and train loss landscapes and the benefits of early stopping. Additionally, they have studied the impact of learning rate schedules on SGD and their influence on the local geometry of the parameter space.

In linear scheduling (Loshchilov and Hutter, 2016), where the learning rate decreases linearly over time. This approach has been found effective for achieving fast convergence and improving training performance by allowing the model to fine-tune its parameters gradually. Another technique, warmup steps (Gotmare et al., 2018), involves gradually increasing the learning rate at the beginning of training before applying the scheduled learning rate. Warmup steps help the model explore the parameter space more effectively and avoid local minima, thus improving convergence.uickly adapt to the training data and improve convergence.

Cosine scheduling (Loshchilov and Hutter, 2016) is another widely used approach, where the learning rate decreases following a cosine function over time. This method has been shown to improve convergence and prevent models from getting stuck in sub-optimal solutions.

Research has also been conducted on restarts in learning rate scheduling and their impact on convergence and training. Restarting the learning rate schedule involves periodically resetting the learning rate to its initial value during training. This technique has been found to improve convergence and prevent the model from getting trapped in poor local minima (Huang et al., 2017). By periodically resetting the learning rate, restarts allow the model to explore different regions of the parameter space and potentially find better solutions.

Other schedulers that control the training have also been shown to accelerate convergence and improve the performance of models. For example, in the paper by Wei (2019), the authors propose a method called "decaying loss" where a fixed learning rate is used, but the magnitude of the update is controlled by gradually reducing the impact of noise on the network.

Another paper by Pan et al. (2021) introduces Eigencurve, a family of learning rate schedules that achieve minimax optimal convergence rates for stochastic gradient descent (SGD) on quadratic objectives with skewed Hessian spectrums. The authors show that Eigencurve outperforms step decay in image classification tasks, especially when the number of epochs is small. The proposed schedulers are designed to approximate Eigencurve and show superior performance compared to cosine decay in certain situations.

In federated learning, Shi et al. (2020) propose a device scheduling policy to achieve fast convergence by considering the trade-off between the number of rounds required to attain a certain model accuracy and the latency per round. Their greedy policy selects devices with the least time consumption in model updating, resulting in a good trade-off between learning efficiency and latency per round. Our previous work also explored scheduling such as using Reinforcement Learning (RL) to learn LR schedules as a policy (Subramanian et al., 2023).

Overall, these papers provide evidence that learning rate schedulers can accelerate convergence and improve the performance of models. This has been well known in research in Deep Learning. The proposed methods, such as decaying loss, Eigencurve, learning rate annealing, and device scheduling policies, offer different strategies for controlling the learning rate and achieving better convergence rates. Several of these schedulers impose drastic changes to the LR schedule which we hypothesize can cause long term changes to the loss trajectory. Our intention is to show a specific transition phenomenon via theoretical and experimental analysis. In the next section, we model SGD dynamics and prove that after a transition, a loss curve can converge or contract towards another loss curve that is predetermined by the learning rates involved in the transition.

3 Proof sketch

SGD dynamics have been modeled in various ways in the past for the purpose of studying theoretical characteristics of the algorithm, and for informing the design of new optimizers and schedulers. In this section we model SGD dynamics in continuous time, and use this to prove how sudden changes in learning rate (LR) can cause one loss trajectory corresponding to the original LR to converge or contract towards the loss trajectory corresponding to the the changed LR. We then use these results and demonstrate the same phenomenon using theoretical and real-world experiments.

First we state the standard SGD algorithm.

Algorithm 1 Stochastic gradient descent

```
Initialize $\theta_0 \in \mathbb{R}^K$ deterministically or randomly
Define non-increasing sequence $(\eta_k)_{k=1}^{\infty} \in (0, \infty)$
for $k = 1, 2, \ldots$ do
    Sample $i_k \sim \text{Unif}(1, \ldots, N)$
    $\theta k \leftarrow \theta_{k-1} - \eta_k \nabla \Phi_{i,k}(\theta_{k-1})$
end for
return $(\theta_k)_{k=0}^{\infty}$
```

Here, $\theta_k \in \mathbb{R}^K$ is the model parameters at iteration k, $\Phi_i : \mathbb{R}^K \to \mathbb{R}$ is the loss function for data subset i, $\eta_k > 0$ is the learning rate at iteration k, N is the number of data subsets or batches of data. This matches the SGD algorithm description in Section 1.1 of (Latz, 2021). At a small enough discretization of steps, we model SGD dynamics by replacing $\theta \leftarrow \theta - \eta \Delta \Phi_i(\theta)$, by $\frac{d\theta}{dt} = -\eta \Delta \Phi_i(\theta(t))$. In practice, we also commonly see fixing $\eta_k = \eta_0$, or using a scheduler (for example, Cosine scheduler) to control the learning rate. While several choices of schedulers that are available continuously change the learning rate, our intuition is that even a single step change to a different learning rate would introduce a drastic change to the dynamics even if starting from the same initial point. To study the effect of changes in LR, we define the algorithm to be studied slightly differently by adding steps related to the *SkipLR* transition to the original **Algorithm 1**, as shown **Algorithm 2**. Here, we study the dynamics of a process where the learning rate switches at a predetermined time step or epoch.

3.1 Assumptions

We present the assumptions required for our proof. The potentials $\Phi_i : \mathbb{R}^K \to \mathbb{R}, i \in 1, \ldots, N$ represent the loss functions associated with different fractions of the training data as mentioned in (Latz, 2021). This is particularly relevant when we consider that stochastic gradient process (SGP) SGP is a valid continuum limit of SGD.

Algorithm 2 SkipLR experiments

```
Initialize $\theta_0 \in \mathbb{R}^K$ deterministically by setting a known seed
Set initial, Skip LR and skip epoch - $\eta_0, \eta_s, k_s$
for $k = 1, 2, \ldots$ do
    if $k \geq k_s$ then
        $\eta = \eta_0$
    else
        $\eta = \eta_s$
    end if
    Sample $i_k \sim \text{Unif}(1, \ldots, N)$
    $\theta k \leftarrow \theta_{k-1} - \eta \nabla \Phi_{i,k}(\theta_{k-1})$
end for
return $(\theta_k)_{k=0}^{\infty}$
```

Specifically, the index i in $\Phi_{i,k}$ refers to a particular subset y_i of the full training data set y at time step k. For the remainder of the paper we drop the k index in $\Phi_{i,k}$ without any loss of generality. Then $\Phi_i(\theta)$ gives the loss of the model parameters θ on the subset y_i (Eq. 1 in (Latz, 2021)).

The full potential is given by the average over all subsets:

$$\bar{\Phi}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \Phi_i(\theta). \tag{1}$$

The training process results in optimal parameters $\theta^* \in \arg\min_{\theta \in X} \bar{\Phi}(\theta)$. For clarity, $\bar{\Phi}$ represents the actual loss function, with inaccurate gradient evaluations arising from randomly substituting $\bar{\Phi}$ by some Φ_i. As in (Latz, 2021) we consider gradient descent algorithms as time stepping discretizations of a certain continuous time gradient flow process. Where we differ from (Latz, 2021) (without any loss of applicability of the methods presented therein) is that in (Latz, 2021) the step sizes η_k represent the length of the time interval in which the flow follows a certain potential Φ_i at the given iteration k, i.e. the time between two switches of potentials. We assume the switches between potentials happen uniformly randomly at every step, but the learning rate switches once in the process of training. By assuming η_k to be at the lowest resolution of discretization, we start our discussion from a better approximation of the actual SGD dynamics shown in **Algorithm 1**. We now continue discussing the potentials Φ_i and other assumptions.

Assumption 1. *The following assumptions are made on the potentials* Φ_i*:*

1. $\Phi_i \in C^2(\mathbb{R}^n; \mathbb{R})$ $\forall i$ *(twice continuously differentiable)*

2. *$\nabla\Phi_i$, $H\Phi_i$ are continuous on $\mathbb{R}^n$ $\forall i$*

3. *There exists $\kappa > 0$ such that for all θ_0, $\theta_0' \in \mathbb{R}^n$:*

$$\langle\theta_0 - \theta_0', \nabla\Phi_i(\theta_0) - \nabla\Phi_i(\theta_0')\rangle \geq \kappa|\theta_0 - \theta_0'|^2, \ \forall i$$

The strong convexity assumption (3) implies that the gradient flows associated with each potential Φ_i are contractive, as shown in the following lemma:

Lemma 1. *If the inequality*

$$\langle\theta_0 - \theta_0', \nabla\Phi_i(\theta_0) - \nabla\Phi_i(\theta_0')\rangle \geq \kappa_i|\theta_0 - \theta_0'|^2$$

holds for some $i \in I$, then the corresponding flows $\phi_i : \mathbb{R}^n \times \mathbb{R}+ \to \mathbb{R}^n$ contract exponentially:

$$|\Phi_i(\theta_0, t) - \Phi_i(\theta_0', t)| \leq c \cdot \exp(-\kappa_i t)|\theta_0 - \theta_0'|$$

for some $c \geq 0$ and for all $\theta_0, \theta_0' \in \mathbb{R}^n$, $t \geq 0$.

Proof. This follows from the strong convexity assumption and Lemma 1 in (Latz, 2021), and implied by Lemma 4.1 given in (Cloez and Hairer, 2015). □

4 Main Result

We can now state and prove the main result:

Theorem 1. *Let $\theta_1(t), \theta_2(t), \theta_3(t)$ be trajectories evolving according to the SGP dynamics:*

$$\frac{d\theta_i}{dt} = -\eta_i \nabla\Phi_i(\theta_i(t)), \quad i = 1, 2, 3$$

where η_i,$\forall i$ are constant learning rates, and $\theta_3(t)$ switches learning rate according to:

$$\eta_3(t) = \begin{cases} \eta_1, & \text{if } t < t_k \\ \eta_2, & \text{if } t \geq t_k. \end{cases}$$

If Φ_1, Φ_2, Φ_3 satisfy Assumption 1, then $\Phi_3(t) \to \Phi_2(t)$ as $t \to \infty$.

What we intend to show is that when starting from the same initial point θ_0, that the conditions presented around η using Skip LR as shown in **Algorithm 2** cause the solutions Φ_i to contract towards each other.

Proof. Let $\theta_{1,k} = \theta_{3,k} = \theta_3(t_k)$. Due to the stochastic nature of SGP dynamics, $\theta_{1,k}$ may not be exactly equal to $\theta_{3,k}$, but given that we fix $\theta_{1,0} = \theta_{2,0} = \theta_{3,0}$, we continue with this assumption. For $t \geq t_k$, θ_3 evolves as:

$$\theta_3(t) = \Phi_3(\theta_{3,k}, t - t_k).$$

Since Φ_2, Φ_3 satisfy Assumption 1, by the contraction lemma we have:

$$\left|\Phi_3(\theta_{3,k}, t - t_k) - \Phi_2(\theta_{2,k}, t - t_k)\right| \leq$$
$$c \cdot \exp(-\kappa_3(t - t_k)) \left|\theta_{3,k} - \theta_{2,k}\right|.$$

Taking $t \to \infty$ and using the continuity of Φ_2, Φ_3, we obtain:

$$\Phi_3(\theta_3(t)) \to \Phi_2(\theta_2(t)).$$

Therefore, after the learning rate switch at t_k, $\Phi_3(t)$ converges to $\Phi_2(t)$. □

4.1 Practical considerations and ϵ bounded contraction

The above proof outlines the asymptotic convergence of the potential functions Φ_3 to Φ_2 as t approaches infinity. To establish a finite-time bound for Φ_3 to come within an ϵ bound of Φ_2, we need to consider the exponential contraction rates derived from Lemma 1. In this section we derive an expression for a critical time t_c after the switch t_k when Φ_3 is within ϵ of Φ_2.

Given that $\Phi_3(t) \to \Phi_2(t)$ as $t \to \infty$, let's consider the behavior of the difference between Φ_3 and Φ_2 using the bound from the contraction lemma:

$$|\Phi_3(\theta_3(t)) - \Phi_2(\theta_2(t))| \leq$$
$$c \cdot \exp(-\kappa_3(t - t_k))|\theta_{3,k} - \theta_2(t_k)|.$$

We want this difference to be within ϵ of each other. Thus, we have:

$$|\Phi_3(\theta_3(t)) - \Phi_2(\theta_2(t))| \leq \epsilon.$$

Substituting the bound expression:

$$c \cdot \exp(-\kappa_3(t - t_k))|\theta_{3,k} - \theta_{2,k}| \leq \epsilon.$$

To find the time t_c when the above inequality holds:

$$\exp(-\kappa_3(t_c - t_k)) \leq \frac{\epsilon}{c|\theta_{3,k} - \theta_{2,k}|}.$$

Taking the natural logarithm of both sides and solving for t_c, and considering discretized case for implementation we get:

$$t_c - t_k \geq \left\lceil \frac{1}{\kappa_3} \ln\left(\frac{c|\theta_{3,k} - \theta_{2,k}|}{\epsilon}\right)\right\rceil.$$

This is the finite time after t_k (i.e. $t_c - t_k$) when Φ_3 comes within an ϵ bound of Φ_2 after the learning rate switch at time t_k, assuming all the conditions and assumptions hold, and with $\kappa_3, c, \epsilon > 0$.

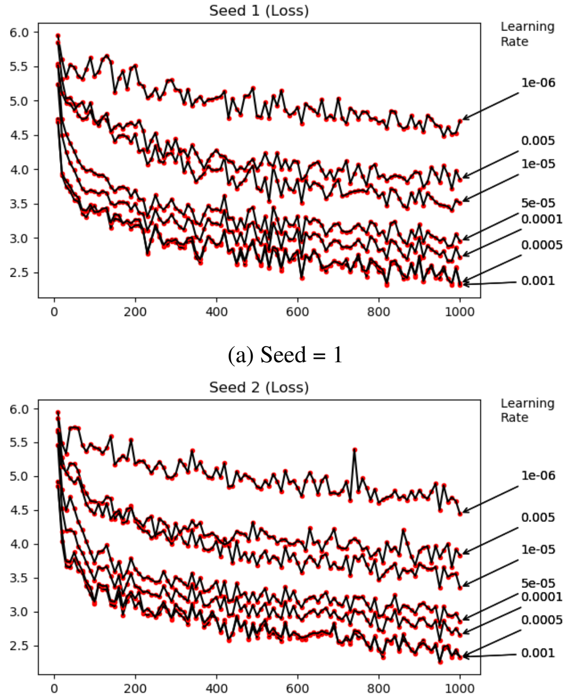
5 Experiments

5.1 Testing for determinism

We test our hypothesis using sythetic data experiments, which we show in Section C of the Appendix. Next we test this behavior in more realistic settings. In all our experiments below, we use the Huggingface library with no modifications except for introducing the new *Skip LR* scheduler. First, we confirm the behavior of fixed seed experiments and the level of determinism that can be achieved. This is important since all the theoretical experiments above are with fixed, known potential functions. In reality, we do not have a closed form expression for the number, or the form of these potential functions involved. The following experiments were all performed on an "ml.g5.16xlarge" GPU instance on Aamzon SageMaker Studio. All experiments use SGD as the optimizer. For Phi-3 (3.8B parameter model), we used a larger "ml.p4d.24xlarge" instance and loaded the model in Bfloat16 datatype. We provide sample code to replicate and use the SkipLR framework on Github.[1]

Our motivation behind the choice of SGD for experiments is two fold 1) to maintain a close relation with the optimizer analyzed in the theoretical analysis, and 2) to test progression of loss curves with fixed learning rates, and study whether it is possible to transition from one loss curve to another using skipLR, a scheduler used specifically for this test. As such we require optimizers used to necessarily not adapt the learning rate during training. Other optimizers used in practice today like Adam sets and adapts parameter-specific learning rates by using the average of the second moments of the gradients. It also calculates the exponential moving average of gradients and square gradients. As such, the effective learning rate is not one fixed (global) learning rate; additionally finding the equivalent global learning rate is not trivial, making the study of fixed LR transitions difficult. To isolate the effects of self-adaptive parameters, we choose to specifically only study changes in the global learning rate, rather than a combination of global learning rates controlled by the SkipLR scheduler along with self-adapted per-parameter learning rates of Adam/AdamW style optimizers.

Given a fixed seed and fixed learning rate (LR), we expect loss progression to be fixed; this has been known for a while through research on determinism. To show the level of determinism attainable, we use the same hardware (ml.g4.16xlarge on Amazon SageMaker), and are experimenting with 3 different seeds, with two runs/trials per learning rate and for many learning rates $1e^{-3}$ to $1e^{-6}$. We use the google/long-t5-tglobal-base model with the News Commentary dataset, and train for 1000 steps. As we can see below, all other parameters remaining constant, the black lines (experiment 1) and red markers (experiment 2) coincide exactly. This is not surprising given algorithmic determinism, and allowing for some platform/implementation non-determinism.



(a) Seed = 1

(b) Seed = 2

Figure 2: Determinism experiments for two different seeds, 7 learning rates from $1e^{-6}$ to $1e^{-3}$.

We note in Fig. 2 that even for complex transformer based language models, loss curves for different LRs can look very similar and can be "grouped" to represent similar trajectories in the function space. The rank-order of these loss trajectories are not correlated with the magnitude of learning rates in this more realistic setting, as opposed to the toy experiments. This grouping and ordering is non-trivial - i.e. LRs that are not close to each other in value can also be grouped, and the order of loss trajectories ordered by loss value at any given point of time cannot be predetermined. That is, for this case, the trajectories are not rank ordered based on the magnitude of the learning

[1] https://github.com/amazon-science/aws-research-science/tree/main/SkipLR

rate.

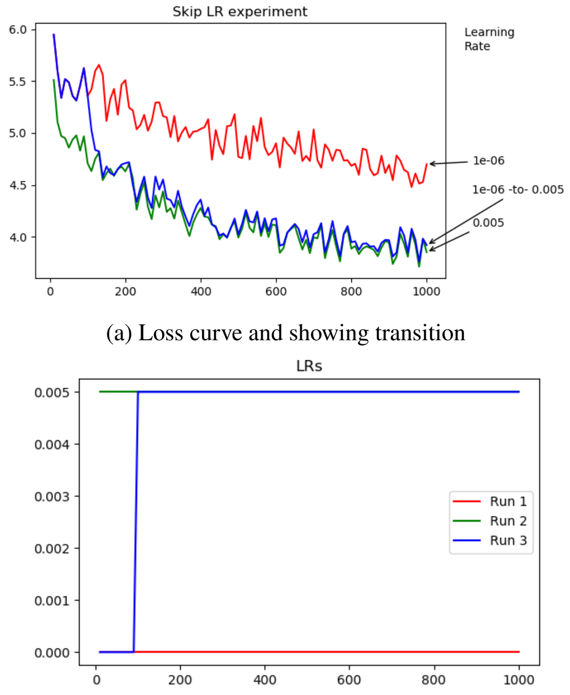
5.2 Fine-tuning experiments

Now that we have seen deterministic behavior with fixed seeds, we conduct furhter experiments with fine-tuning models. We define transition success as ϵ for converging one loss curve to another as 0.01 or 1%, and measure the transition success % for two different models and datasets as shown in Table 2. For each model and dataset combination, we test with 100 different seeds. For each experiment, we choose two learning rates to transition between and the transition point t_k. We then run three sub-experiments with the same seed - Φ_1 representing flow under η_1, Φ_2 representing flow under η_2, and Φ_3 representing flow under Skip LR scheduler between η_1, η_2. In all cases $t_k = 200$, and the total number of steps is 1000. Although this is a small number of steps (less than an epoch for our batch size of 8), we see clear transitions. Depending on the set up, we see several runs where the final loss (even at a fraction of the total steps we experimented for) is significantly close to what is achievable through longer training durations (see 5c in the supplementary material for instance). This is of course true for many experiments where the trajectory leads to a sub-optimal region with higher loss; in these cases, we do not expect to achieve a very low loss when training to many more iterations. In fact, these are conditions where we see that the transition to other loss curves are less likely. Table 2 shows that the constants κ, c, ϵ defining the likelihood of transitioning is inherent to the problem (defined by the model, task, dataset, batch size etc.). Even for complicated problems, we see (as shown in Fig 3 that transition from one loss curve to another is possible; this can inform researchers in the creation of new schedulers to accelerate convergence.

Supplementary material provided includes more examples of transition for both experiments mentioned in 2.

5.3 Pretraining experiments

To further validate our theoretical findings and explore the dynamics of learning rate transitions in a more extensive setting, we conducted pretraining experiments with the Phi-3 model, a large language model with 3.8 billion parameters. Unlike the previous fine-tuning experiments, where we started from a pretrained checkpoint, in this case, we trained the model from scratch for 20,000 steps using the same dataset as in Table 2.



(a) Loss curve and showing transition

(b) LR schedule

Figure 3: Skip LR schedule (in blue) compared to fixed LR schedules (red and green), and corresponding loss curves showing successful transition between very different learning rates ($1e^{-6}$ and $5e^{-3}$) for the Google T5 base model when fine tuned with the News commentary dataset on a translation task.

Figure 4 illustrates the loss curves for different learning rate transitions during the pretraining process. We tested multiple transition points at steps 2500, 5000, 7500, and 10000, switching between learning rates of $1e^{-3}$ and $1e^{-6}$. The results clearly demonstrate that the loss curves exhibit the contraction behavior predicted by our theoretical analysis, converging towards the trajectory determined by the new learning rate after the transition point.

Notably, we observe that the loss curves closely agree with the theoretical results, even in this large-scale pretraining setting. A detailed plot of our theoretical experiments can be found in Section C of the Appendix. This suggests that the contraction phenomenon holds true not only for fine-tuning scenarios but also during the initial pretraining phase, where the model is learning from scratch.

Our findings from the Phi-3 pretraining experiments further corroborate the hypothesis that loss curves can transition and converge towards predetermined trajectories governed by the new learning rate, even in the case of extremely large language models. We conjecture that the fine-tuning experi-

Model	Steps	Dataset	Transition %	Transition before step %			Transition statistics		
				25	50	75	Median	Min	Max
Google T5 base	1000	News Commentary (translation)	98	1	41	93	525	250	1000
Roberta Large	1000	Xglue (NER)	80	7	67	77	410	210	1000
GPT2	1000	Wikitext (CLM)	100	100	100	100	200	200	200
GPT2 (1e-4 to 1e-3)	1000	Wikitext (CLM)	100	100	100	100	200	200	210
Gemma 2B	1000	Wikitext (CLM)	100	0	4	43	780	420	950
Gemma 2B	10, 000	Ultra Textbooks (CLM)	100	40	100	100	2540	200	2800
Phi-3 3.8B *	10, 000	Ultra Textbooks (CLM)	100	100	100	100	1000	1000	1000

Table 2: Transition statistics for multiple models tested on translation, NER and CLM tasks. Each row represents 100 seed experiments of 1000 steps each. Transitions are from $1e-6$ to $1e-3$ unless specified. Total number of optimizer update steps taken are 100 seeds each experiment × total number of steps per seed × total runs per experiment (3) = $7.5M$. Due to the large resources demanded by LLMs beyond 3B in size, Phi-3 (*) is run for 5 seeds instead of 100.

ments presented in Table 2 essentially zoom into the bottom-right portion of the pretraining loss curves, where transitions continue to occur, albeit on a smaller scale. These pretraining results provide additional evidence supporting our theoretical analysis and demonstrate the practical implications of learning rate transitions for accelerating the training of large language models, both during pretraining and fine-tuning stages.

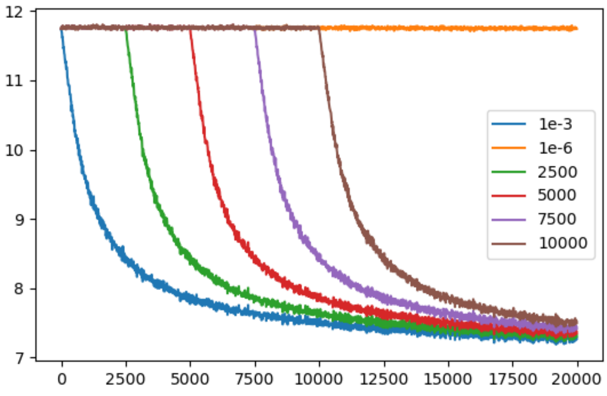


Figure 4: Pretraining experiments with Phi 3, 3.8B parameter model with transitions from various points

6 Limitations

While our findings are promising, there are several important considerations. First, the theoretical analysis relies on strong convexity and smoothness assumptions about the loss landscape. While this is a standard setup in many theoretical settings, our work doesn't explore more complex settings, especially a setting that includes modern-day LLMs. Second, our experiments are limited to fixed learning rates. We believe dynamic learning schedulers add an additional layer of complexity that can affect the Skip phenomenon, which we do not explore in this work.

7 Conclusion

In this work, we have analyzed the dynamics of SGD training under different learning rate schedules. We modeled SGD as a stochastic gradient flow and considered the effect of switching the learning rate at a predetermined time, which we call "SkipLR". Under standard assumptions on the loss landscape, we proved that the loss curves resulting from different constant learning rates contract towards each other after a SkipLR switch. We verified this behavior theoretically for simple cases and empirically for large language models. Our analysis gives insight into how learning rate schedules impact the training dynamics. This could help design new schedules that lead to faster convergence. An interesting direction for future work is to extend the analysis to stochastic processes with non-uniform sampling, and establish quantitative bounds on the convergence rates.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. Falcon-40B: an open large language model with state-of-the-art performance.

Atilim Gunes Baydin, Robert Cornish, David Martinez Rubio, Mark Schmidt, and Frank Wood. 2017. Online learning rate adaptation with hypergradient descent. *arXiv preprint arXiv:1703.04782*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *Preprint*, arXiv:2210.11416.

Bertrand Cloez and Martin Hairer. 2015. Exponential ergodicity for markov processes with random switching.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. *arXiv preprint arXiv:1810.13243*.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. 2017. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*.

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

Jonas Latz. 2021. Analysis of stochastic gradient descent in continuous time. *Statistics and Computing*, 31(4):39.

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

Preetum Nakkiran. 2020. Learning rate annealing can provably help generalization, even for convex problems. *Preprint*, arXiv:2005.07360.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

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

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Wenqi Shi, Sheng Zhou, and Zhisheng Niu. 2020. Device scheduling with fast convergence for wireless federated learning. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE.

Shreyas Subramanian and Vignesh Ganapathiraman. 2023. Zeroth order greedylr: An adaptive learning rate scheduler for deep neural network training. In *2023 IEEE 4th International Conference on Pattern Recognition and Machine Learning (PRML)*, pages 593–601. IEEE.

Shreyas Subramanian, Vignesh Ganapathiraman, and Aly El Gamal. 2023. Learned learning rate schedules for deep neural network training using reinforcement learning.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Jiakai Wei. 2019. Forget the learning rate, decay loss. *arXiv preprint arXiv:1905.00094*.

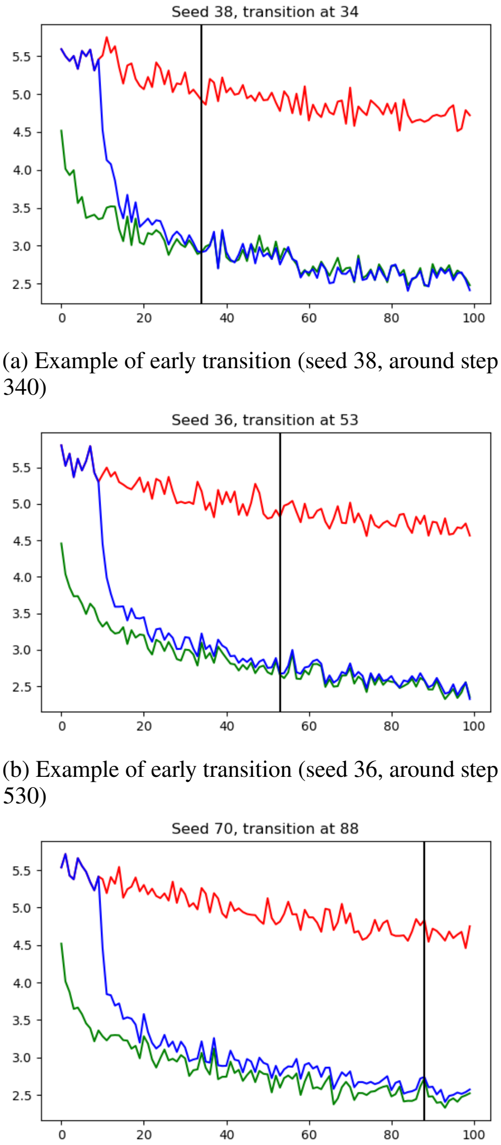
Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

A Appendix

B Additional figures for transition at different steps

In Figures 5a to 6a, we include more examples of transition for both experiments mentioned in 2. Given the large number of experiments that we need to store model loss states, optimizer and scheduler states, we only log every 10'th step. In the images below, we plot loss vs. logged step, and so the actual step can occur within 10 steps of when the loss is logged. All code needed to reproduce these results along with additional experiments will be made available.

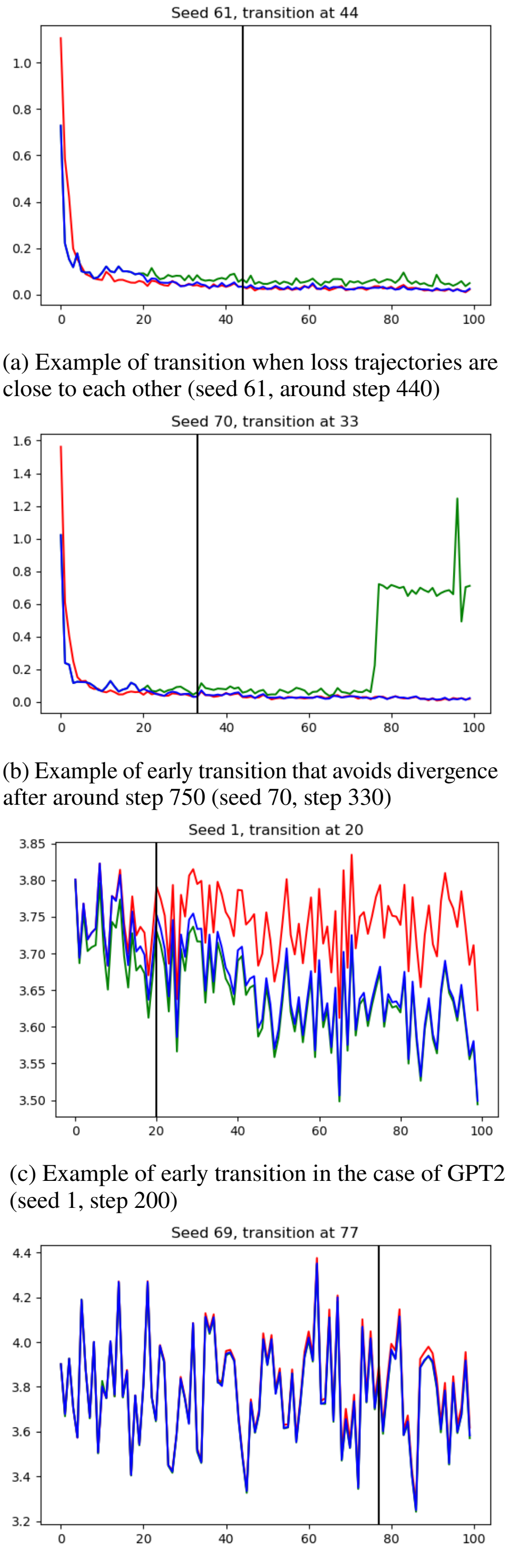


(a) Example of early transition (seed 38, around step 340)

(b) Example of early transition (seed 36, around step 530)

(c) Example of late transition (seed 70, around step 880)

Figure 5: More figures for transition at various steps



(a) Example of transition when loss trajectories are close to each other (seed 61, around step 440)

(b) Example of early transition that avoids divergence after around step 750 (seed 70, step 330)

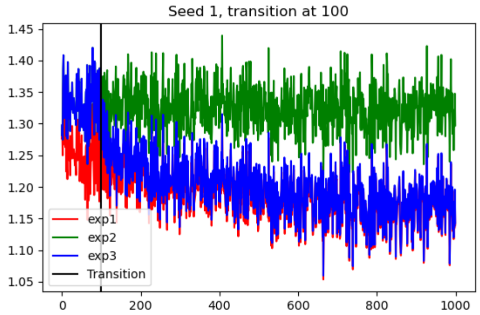
(c) Example of early transition in the case of GPT2 (seed 1, step 200)

(d) Example of very close loss curves for Gemma 2B resulting in late transition (seed 69, step 770). We continued experiments with 10x steps of the original experiments and report more results in Table 2.

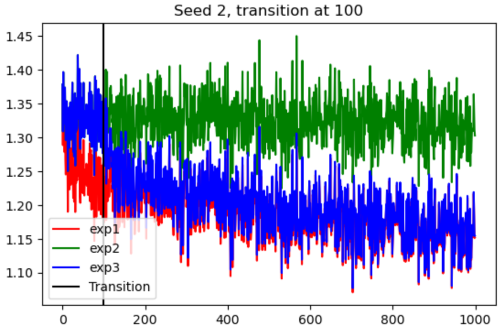
Figure 6: More figures for transition at various steps

C Synthetic data experiments

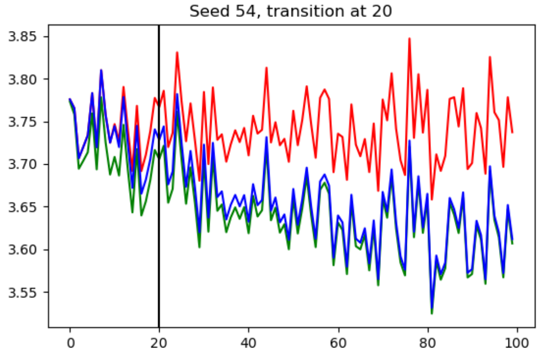
We run some theoretical experiments approximating the dynamics of $\bar{\Phi}$ using $\Phi_0(\theta) = \theta$, i.e. only a single potential function. When using skip LR scheduler, we see that the skips cause the curves to exponentially converge (Fig. 1 a) and b). When $N = 2$, we uniformly randomly switch between $\Phi_0(\theta) = \theta + 1$ and $\Phi_1(\theta) = \theta - 1$, with two sets of learning rates. We use the Runge Kutta 5 solver from Python3's Scipy library for solving the initial value problem with a fixed $\theta(0)$. The stationary point $\theta^* = 0$ for all problems, and the loss function used is RMSE; note that here since $\theta^* = 0$, the plot also reveals the parameter $\theta(t)$. As we can see in all cases of Figure 8, transition within some ϵ happens well before $t \to \infty$. Note here that with this theoretical set up, the ranking of the loss curves corresponds to the values of the LRs used. That is, a higher learning rate corresponds to a steeper descent as expected; in the main paper we demonstrate how this is different for real-world transformer based training experiments. In these experiments modeling learning dynamics using potential functions, when transitioning from a higher average loss value to a lower one, the convergence rate effectively accelerates to eventually match the lower loss curve. This matches our proof $(\Phi_2(t) \to \Phi_3(t)$. We also note that the loss curves as N increases beyond 1 with random switching between potential closely resemble what one might see when training standard deep neural networks.



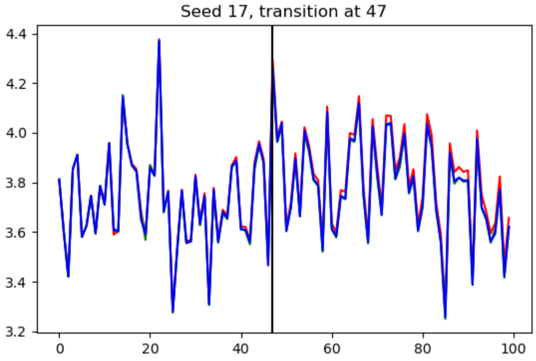
(a) Example of noisy loss curves for Phi-3, still showing very early transition - Seed=1. Although Phi 3 (3+ Billion parameters) is a large model, transitions are still made successfully.



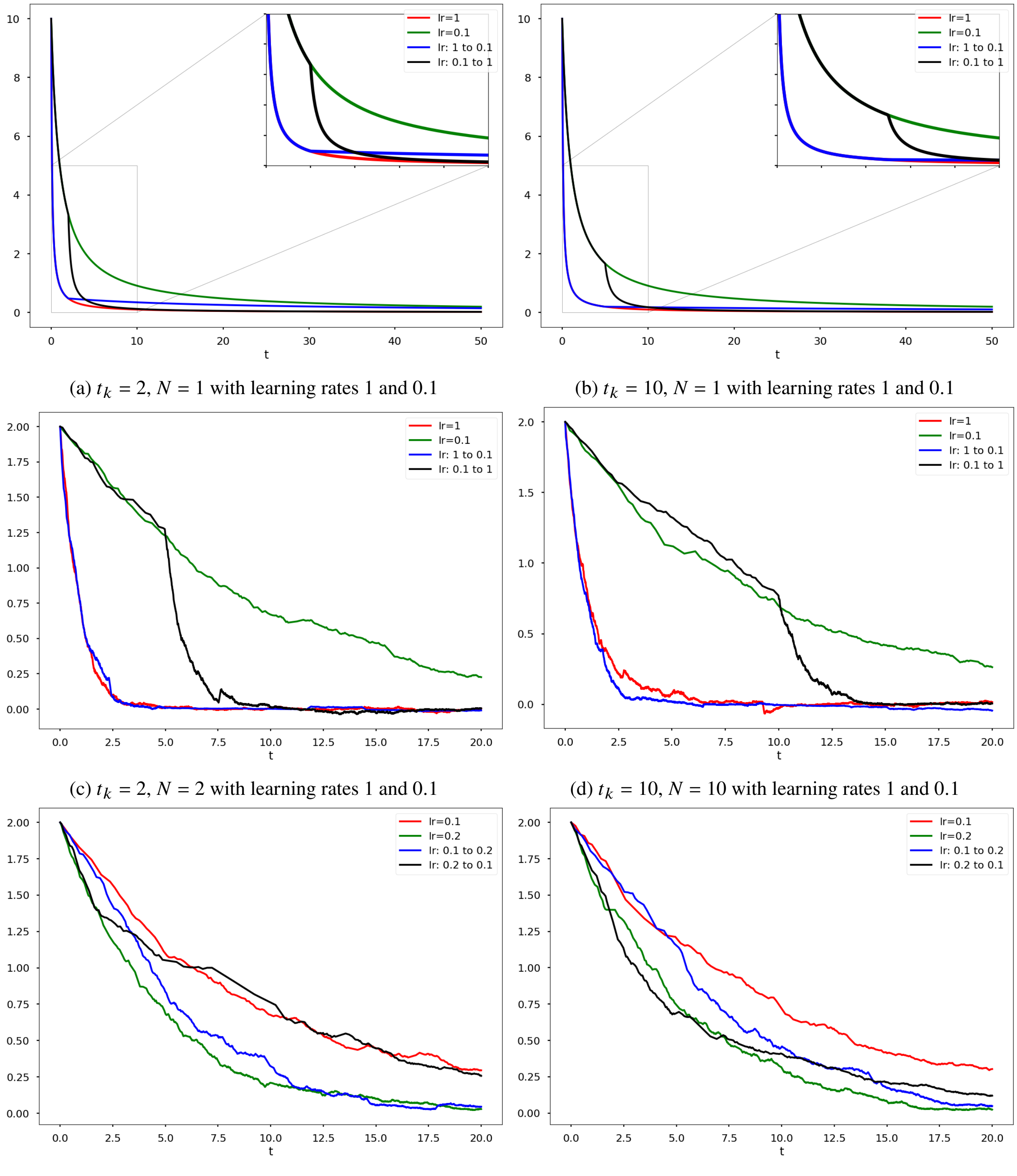
(b) Example of noisy loss curves for Phi-3, still showing very early transition - Seed=2.



(c) Example of early transition in the case of GPT2 (seed 54, step 200)



(d) Example of very close loss curves for Gemma 2B with early transition (seed 17, step 470)



(a) $t_k = 2$, $N = 1$ with learning rates 1 and 0.1

(b) $t_k = 10$, $N = 1$ with learning rates 1 and 0.1

(c) $t_k = 2$, $N = 2$ with learning rates 1 and 0.1

(d) $t_k = 10$, $N = 10$ with learning rates 1 and 0.1

(e) $t_k = 2$, $N = 2$ with learning rates 0.1 and 0.2

(f) $t_k = 10$, $N = 2$ with learning rates 0.1 and 0.2

Figure 8: Transitions demonstrated with dynamics $\bar{\Phi}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \Phi_i(\theta)$, where 1) $N = 1$ and $\Phi_0' = \theta$ in figures a. and b., with learning rates $[0.1, 1]$ 2) $N = 2$ and $\Phi_0' = \theta + 1$, $\Phi_1' = \theta - 1$ with learning rates $[0.1, 1]$ figures c. and d. and 3) Potential functions as defined in 2) but with learning rates $[0.1, 0.2]$ figures e. and f. In all cases we test two transition points, $t_k = 2$ and $t_k = 10$