Branches Switching Based Pre-Training Strategy for Version Iteration of Large Language Models

Anonymous ACL submission

Abstract

Due to the continuous emergence of online data, version iteration has become an indispensable requirement for Large Language Models (LLMs), which exacerbates the training cost of LLMs. Hence, one of the pivotal challenges for LLMs is how to reduce the total training cost across different versions. To achieve a better balance between the pre-training performance and training cost, we conduct a systematic investigation into the impact of various learning rate schedules. Extensive experiments on commonly used learning rate schedules show that these approaches primarily focus on the performance of LLMs of the current version, but overlook the mutual influence of training processes of LLMs across different versions. To address above issue, we design a pre-training strategy called Branches Switching based Pre-Training for the training of LLMs across different versions. Compared with pre-training LLMs of different versions from scratch, our strategy reduces the total training cost to 58% while maintaining optimal pre-training performance.

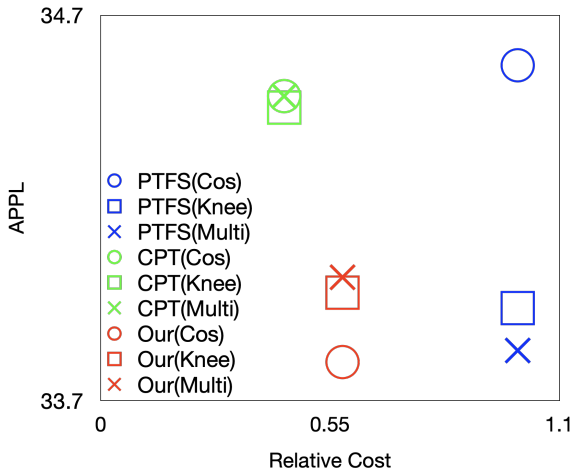

Figure 1: The pre-training performance vs. total training cost of the proposed Branches Switching based Pre-Training (BSPT), Pre-Training From Scratch (PTFS) and Continual Pre-Training (CPT) on LLaMA-153M. "APPL" denotes the average perplexity (↓) of LLMs of different versions, "Relative Cost" denotes the relative training cost of LLMs of different versions. The lower left corner achieves the best trade-off.

1 Introduction

In recent years, there has been significant progress in the research of Large Language Models (LLMs). By conducting large-scale parameters training on massive datasets, LLMs have demonstrated remarkable capabilities, contributing to unprecedented advancements in various fields (Wang et al., 2023a; Guo et al., 2024; Wu et al., 2023; Cui et al., 2023). However, the training cost of LLMs is significantly higher than that of traditional NLP models. In practical applications, LLMs have to face the requirement of version iteration due to the continuous emergence of online data, which exacerbates the training cost of LLMs. Therefore, how to reduce training cost while ensuring optimal pre-training performance of LLMs across different versions has become one of the pivotal challenges in the practical implementation of LLMs.

Approaches applicable for LLMs version iteration can be broadly categorized into two types: 1) **Pre-Training From Scratch (PTFS)**: conducting pre-training on full datasets that include both old and new data. LLMs such as LLaMA (Touvron et al., 2023a,b), GLM (Zeng et al., 2023), and Baichuan (Yang et al., 2023) employ this approach for version iteration. It can yield a better pre-training performance, but also involves extremely high training cost. 2) **Continual Pre-Training (CPT)**: conducting pre-training on new data based on the checkpoints of existing LLMs (Gogoulou et al., 2023b; Xie et al., 2023). This approach is often utilized in constrained scenarios, such as limited computational resources or unavailability of all data. Compared with PTFS, CPT incurs lower cost but may result in diminished pre-training performance of LLMs.

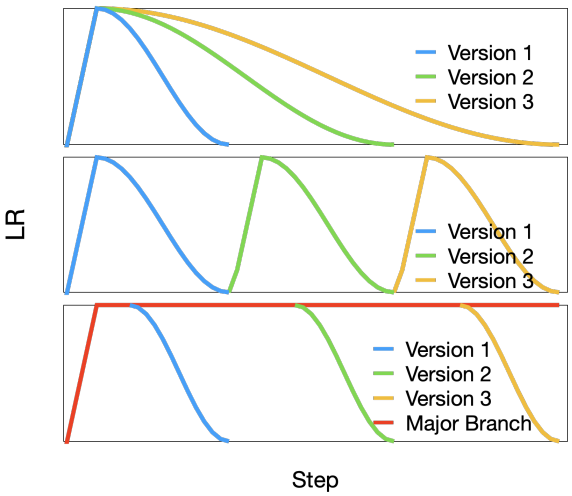

Figure 2: Learning rate curves of different approaches for version iteration of LLMs. The three approaches are Pre-Training From Scratch (PTFS), Continual Pre-Training (CPT) and the proposed Branches Switching based Pre-Training (BSPT) from top to bottom.

These approaches focus on the performance of LLMs of current version. PTFS emphasizes the performance of LLMs on all data, while CPT addresses how to balance the performance on old and new data. However, both approaches overlook the mutual influence of training processes across different versions. We can analyze the aforementioned issue from the perspective of optimization.

- **PTFS**: Experiments show that the learning rate schedules with period-related hyper-parameters can achieve the optimal performance for PTFS. When training LLMs of different versions using these schedules, it is necessary to configure distinct decay periods for each version. In this scenario, the inability to reuse checkpoints of existing LLMs results in high training cost.

- **CPT**: A complete learning rate decay period promotes the convergence of LLMs, which ensures the pre-training performance of LLMs of current version. However, it is detrimental for the parameters of LLMs of future versions to escape from the current local optimum and search for a better local optimum.

To achieve a better balance between pre-training performance and training cost, we systematically explore the optimal learning rate settings for PTFS and CPT. Based on the exploratory experiments, we design a pre-training strategy called **Branches Switching based Pre-Training (BSPT)**, that is applicable to learning rate schedules such as cosine (Smith and Topin, 2019), knee (Iyer et al., 2023), and multi-step (Bi et al., 2024) learning rate schedule. An intuitive comparison of PTFS, CPT and BSPT is shown in Figure 1. As depicted in Figure 2, our strategy comprises one major learning rate branch and multiple minor learning rate branches. The major branch maintains a higher learning rate, which facilitates the discovery of better local optimum for LLMs. The minor branches consist of decaying learning rate schedules, and the number of minor branches matches the number of versions, ensuring the convergence of different LLMs. LLMs of different versions share one primary branch. Hence, the training of current LLMs can reuse the checkpoints of previous LLMs on major branch, which reduces the total training cost. Additionally, LLMs of different versions utilize distinct minor branches for convergence. Therefore, the training of current LLMs is not influenced by previous versions.

Our main contributions are as follows:

- Compared with PTFS, our strategy reduces the total training cost to 58% while maintaining optimal pre-training performance. To the best of our knowledge, this is the first pre-training strategy designed for version iteration of LLMs.

- Empirical experiments demonstrates the generalization of our strategy in model scaling, data scaling, and maximum learning rate.

- We provide a better understanding of learning rate schedule at LLMs to help prioritize future exploration towards efficient training.

2 Preliminary

2.1 Experiment Setting

Models We conduct all experiments on LLaMA (Touvron et al., 2023a,b). The main experiments are conducted on LLaMA-153M (similar experiments on LLaMA-1.2B are listed in Appendix A.3). To verify the generalization of model scaling, we also report results on LLaMA-206M, LLaMA-406M, LLaMA-608M, LLaMA-2.1B and LLaMA-3.1B respectively.

Datasets We pre-train all LLMs on Chinese and English datasets. Similar to LLaMA (Touvron et al., 2023a,b), our pre-training data sources include: 1) Code; 2) Paper; 3) Wikipedia; 4) Books;

5) Mathematics; 6) Commoncrawl and C4; 7) Webpage; 8) Translation; 9) Others. Constrained by GPU resources, we doesn't use the entire dataset to train LLMs. The training data of LLMs is sampled from 764 million samples.

Learning Rate Schedule We conduct experiments on commonly used learning rate schedules for LLMs (Zhao et al., 2023), including constant, inverse square root, cosine, knee and multi-step learning rate schedules. The specific learning rate curves are plotted in Figure 6 in Appendix.

Hyper-Parameters The batch size is 512 samples, and the maximum length of samples is 2048. Hence, there are 1.05 million tokens per step. Limited by GPU resources, most LLMs are trained for 40K steps (about 42B tokens). To verify the generalization of our method in terms of data scaling, we also train LLaMA-153M for 320K steps (about 336B tokens), and LLaMA-1.2B for 160K steps (about 168B tokens). We set the warmup length for all LLMs as 2K steps (about 2.1B tokens). More details about hyper-parameters of LLMs are presented in Table 11 in Appendix.

Evaluation We mainly use perplexity (PPL) to evaluate the pre-training performance of LLMs.

2.2 Pre-Training From Scratch

Different Learning Rate Schedules The learning rate schedules can be broadly categorized into the following four types (Wu et al., 2019; Wu and Liu, 2023; Jin et al., 2024): 1) Fixed learning rate policy; 2) Decaying learning rate policy; 3) Cyclic learning rate policy; 4) Composite learning rate policy. To choose the optimal learning rate schedules for PTFS, we firstly conduct systematic experiments to study the effect of these learning rate schedules.

Experimental results of different learning rate schedules with training steps ranging from 10k to 40k are depicted in Table 1. Compared to fixed policy and decaying policy, **cyclic policy and composite policy achieve superior pre-training performance**. These two types of learning rate schedules all have hyper-parameters associated with the period, which significantly impacts the pre-training performance of LLMs. In this case, LLMs of different versions require different decay periods. Hence, the training of current LLMs cannot reuse the checkpoints of previous LLMs, which results in high training cost.

Schedule	PPL			
	10K	20K	30K	40K
Fixed				
Const(1e-3)	42.97	39.41	38.05	37.27
Const(1e-4)	70.24	54.03	47.93	44.56
Decaying				
Inv-Sqrt	42.03	38.22	36.67	35.75
Cyclic				
Cos	39.85	35.93	34.35	33.42
Composite				
Knee	38.99	35.23	33.73	32.86
Multi	**38.81**	**35.14**	**33.63**	**32.74**

Table 1: PPLs of the most commonly used learning rate schedules on LLaMA-153M. Experiments on LLaMA-1.2B are also listed in Table 12 in Appendix.

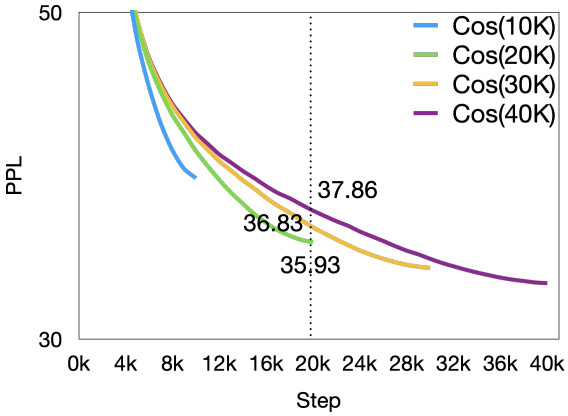

Figure 3: Comparison of cosine learning rate schedules with different decay periods.

Different Periods Decay period is one of the most important hyper-parameters for cyclic and composite learning rate policies. Without loss of generality, we study the impact of decay period based on cosine learning rate schedule. We compare the pre-training performance of LLMs trained with different cosine decay periods.

Figure 3 shows the PPL curves for different decay periods. For LLMs trained with 20K steps, a shorter decay period leads to better pre-training performance(20K: 35.93 vs. 30K: 36.83 vs. 40K: 37.86). Experimental results illustrate that **a complete decay period can lead to improved pre-training performance in LLMs**. A complete decay period indicates that the learning rate gradually decreases from its maximum to the minimum. It ensures the convergence of LLMs parameters to local optimum.

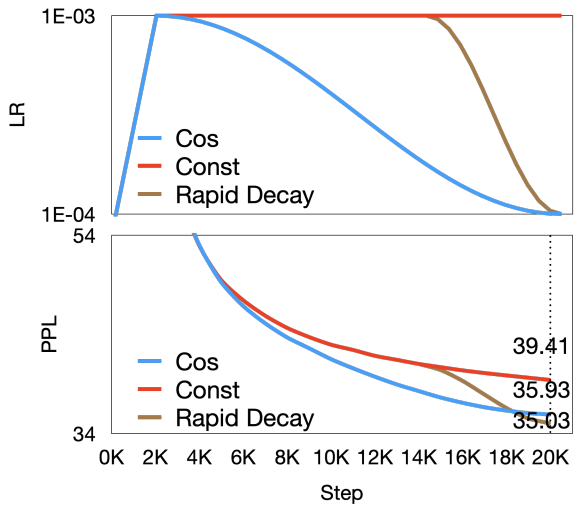

Figure 4: Comparison between cosine learning rate schedule and concatenated learning rate schedule.

Schedule	Type	PPL		
		20K	30K	40K
Cos	Rewarm	36.36	34.81	33.90
	Reset	**36.28**	**34.74**	**33.82**
	Gap	-0.08	-0.07	-0.08
Knee	Rewarm	35.89	34.50	33.66
	Reset	**35.67**	**34.27**	**33.44**
	Gap	-0.22	-0.23	-0.22
Multi	Rewarm	35.86	34.53	33.71
	Reset	**35.67**	**34.30**	**33.50**
	Gap	-0.19	-0.23	-0.21

Table 2: Comparison between rewarm and reset for CPT on LLaMA-153M.

Larger Average Learning Rate Another key to improve pre-training performance of LLMs is to find a better local optimum for LLMs. Existing researches indicate that a high learning rate can accelerate the convergence of model training (Smith and Topin, 2019; Smith, 2018). Review the experimental results in Table 1. Compared to the cosine learning rate schedule, LLMs trained with a constant learning rate schedule exhibit inferior pre-training performance, while LLMs trained with composite learning rate schedules (such as knee and multi-step) exhibit superior pre-training performance. We hypothesize that the differences stem from the completeness of the decay period. To further validate this hypothesis, we concatenate constant and cosine learning rate schedules.

The specific learning rate and PPL curves are plotted in Figure 4. Both cosine learning rate schedule and the concatenated learning rate schedule have a complete decay period, but the concatenated learning rate schedule has a larger average learning rate. As the results shown in Figure 4, although the pre-training performance of the constant learning rate schedule is inferior than that of the cosine learning rate schedule, the concatenated learning rate schedule exhibits superior pre-training performance compared to that of the cosine learning rate schedule. It indicates that the constant learning rate schedule can find better local optimum compared to the cosine learning rate schedule. In other words, **a larger average learning rate is beneficial for LLMs to find a better local optimum**.

2.3 Continual Pre-Training

Rewarm or Reset One key factor for CPT is *how to start the new learning rate schedule for LLMs of new version?* There are two methods for this problem, including rewarm and reset. Rewarm means progressively increasing the learning rate from minimum to maximum, while reset means reseting the learning rate as maximum directly.

To answer this question, we conduct comparison experiments in Table 2. Different from previous study (Gupta et al., 2023), **reset is stably better than rewarm**. Compared with rewarm, reset has a larger average learning rate, which is beneficial for LLMs to find a better optimum. Hence, we choose reset to start the new learning rate schedule for LLMs of new version.

Combination Another key factor for CPT is *how to choose the combination of old and new learning rate schedules?* For general LLMs, the datasets of different versions are similar. If the best combination consists of more than two types of learning rate schedules, we can replace one of them to achieve better performance. Hence, without loss of generality, we hypothesize that at most two different learning rate schedules are used. There are three methods to choose learning rate schedule for CPT, including extending, cycling and switching.

- **Extending**: The LLMs of new versions continual to be trained with the original learning rate schedule. This method is only applicable to the fixed and the decaying learning rate policies, which have no period-related hyper-parameters.

Schedule	PPL		
	20K	30K	40K
Extending			
Const(1e-3)	39.41	38.05	37.27
Const(1e-4)	54.03	47.93	44.56
Inv-Sqrt	38.22	36.67	35.75
Switching			
Cos	38.23	37.30	36.64
Knee	37.37	36.59	36.03
Multi	37.12	36.34	35.79
Cycling			
Cos	36.28	34.74	33.82
Knee	**35.67**	**34.27**	**33.44**
Multi	**35.67**	34.30	33.50

Table 3: Comparison of different methods for the combination of old and new learning rate schedules.

- **Cycling**: The LLMs of new versions are trained with the same learning rate schedule, which may has different parameters with the original one, such as the decay period. This method is applicable to the cyclic and composite learning rate policies, such as cosine learning rate schedule.

- **Switching**: The LLMs of new versions are trained with two different learning rate schedules. In order to distinguish from the method of cycling, we choose a constant learning rate schedule for new version. In order to ensure the convergence of LLMs, we set a small learning rate for switching.

We compare these methods in Table 3. Experimental results show that LLMs trained with cycling is significantly better than extending and switching. It means that **cycling is the best choice for CPT**.

Comparison between PTFS and CPT After determining the optimal learning rate setting for LLMs, we compare PTFS and CPT on cosine, knee and multi-step learning rate schedules in Table 4. Compared to PTFS, CPT incurs lower training cost, but also results in inferior performance for LLMs. **The performance gap between PTFS and CPT increases with the number of versions**.

2.4 Definition of Our Strategy

Based on the above experiments, we believe that the pre-training of LLMs carry out two tasks simultaneously: 1) Searching for a better local optimum;

Sch.	Strategy	Cost	PPL		
			20K	30K	40K
Cos	PTFS	1.00×	**35.95**	**34.35**	**33.42**
	CPT	**0.44**×	36.28	34.74	33.82
	Gap	-	-0.33	-0.39	-0.40
Knee	PTFS	1.00×	**35.23**	**33.73**	**32.86**
	CPT	**0.44**×	35.67	34.27	33.44
	Gap	-	-0.44	-0.54	-0.58
Multi	PTFS	1.00×	**35.14**	**33.63**	**32.74**
	CPT	**0.44**×	35.67	34.30	33.50
	Gap	-	-0.53	-0.67	-0.76

Table 4: Performance gaps between PTFS and CPT on LLaMA-153M.

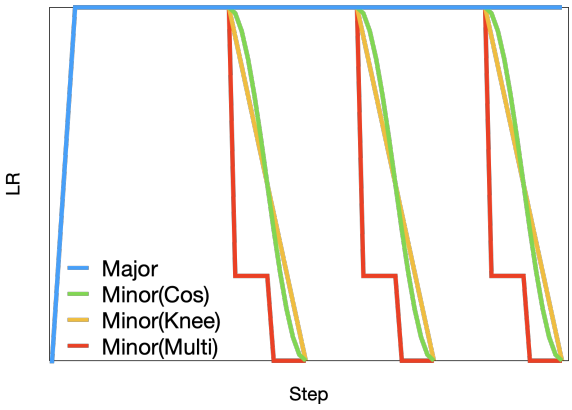

Figure 5: Implementations for the the proposed BSPT, which are based on cosine, knee and multi-step learning rate schedules.

2) Converging to the nearest local optimum. When the learning rate is large, the training of LLMs focuses more on searching for a better local optimum; as the learning rate gradually decays, the focus shifts to converging parameters of LLMs to the nearest local optimum. For the training of LLMs of different versions, a complete decay period ensures the pre-training performance of current LLMs, but is harmful for new LLMs to find a better optimum. When the learning rate decays slowly, LLMs gradually converge to different local optima before finding the final local optimum. It doesn't bring any benefit to the pre-training performance of LLMs.

To address the above issues, we propose **Branches Switching based Pre-Training (BSPT)** strategy for version iteration of LLMs. Our strategy is applicable to different learning rate schedules, including cosine, knee, and multi-step schedules. The specific learning rate curves are plotted in Fig-

Sch.	Strategy	Cost	PPL		
			20K	30K	40K
Cos	PTFS	1.00×	35.95	34.35	33.42
	CPT	**0.44**×	35.67	34.30	33.50
	BSPT	0.58×	**35.03**	**33.59**	**32.78**
Knee	PTFS	1.00×	35.23	**33.73**	**32.86**
	CPT	**0.44**×	35.67	34.27	33.44
	BSPT	0.58×	**35.22**	33.78	32.95
Multi	PTFS	1.00×	**35.14**	**33.63**	**32.74**
	CPT	**0.44**×	35.67	34.30	33.50
	BSPT	0.58×	35.43	33.76	32.86

Table 5: Comparison of different strategies for training LLaMA-153M of different versions. The experiments on LLaMA-1.2B are presented in Table 13 in Appendix.

Schedule	PPL		
	20K	30K	40K
Const	39.41	38.05	37.27
+ RD(10%)	35.33	33.97	32.93
+ RD(20%)	35.07	**33.59**	**32.71**
+ RD(30%)	**35.03**	33.69	32.81
+ RD(40%)	35.07	33.73	32.85
+ RD(2K)	35.33	34.02	33.26
+ RD(4K)	35.07	33.71	32.93
+ RD(6K)	**35.03**	33.59	32.78
+ RD(8K)	35.07	**33.55**	**32.71**

Table 6: PPL of constant schedule and our strategy with different rapid decay periods. "RD" is rapid decay, and values in parentheses indicate the rapid decay periods. Experiments are conducted on LLaMA-153M.

ure 5. Our strategy comprises one major learning rate branch and multiple minor learning rate branches. The number of minor branches matches the number of versions. The major branch maintains a higher learning rate, which facilitates the discovery of better local optimum. And the minor branches employ rapid decay learning rate schedules, which facilitate LLMs to converge to the nearest local optimum. Besides, LLMs of different versions share the same major branch learning rate. As a result, the training of current LLMs can reuse the checkpoints of previous versions, which reduces the total training cost. Compared with pre-training LLMs of different versions from scratch, our strategy reduces the total training cost to 58% while maintaining optimal pre-training performance.

3 Experiment

In this section, we focus on the characteristics of our strategy, including: 1) Comparison of different pre-training strategies in terms of performance and cost; 2) How to determine the value of rapid decay period; 3) Generalization in terms of model scaling, data scaling and maximum learning rate.

3.1 Version Iteration of LLMs

Table 5 lists the experimental results of PTFS, CPT and BSPT in terms of the training cost and pre-training performance. Compared with PTFS and CPT, **BSPT reduces the total training cost to 58% while maintaining optimal pre-training performance**. A more intuitive comparison of these methods can be seen in Figure 1 and Figure 2.

3.2 Rapid Decay Period

The rapid decay period of BSPT significantly impact the pre-training performance and training cost. On the one hand, the additional training cost of BSPT depends on the decay periods of minor branches. On the other hand, an excessively long period may result in a sub-optimal local optimum, while an excessively short period may prevent LLMs from converging sufficiently to local optimum. Therefore, how to appropriately configure the rapid decay period of BSPT is crucial in balancing training cost and pre-training performance of LLMs across different versions. To provide empirical answers to this question, we conduct experiments of minor branches trained with different absolute and relative periods. Additionally, we also provide another method for determining the value of rapid decay period in section 3.3, which is more accurate but has a higher cost.

Relative Periods We conduct experiments by setting the rapid decay periods as 10%, 20%, 30%, and 40% of the total steps, respectively. Based on experimental results in Table 6, we find that the optimal pre-training performance of LLMs is achieved when the rapid decay period is set to 20%-30% of the total steps. **Both excessively long and excessively short rapid decay periods can result in suboptimal pre-training performance**.

Absolute Periods We also conduct experiments by setting the rapid decay periods as 2K, 4K, 6K, and 8K steps, respectively. The experimental results of absolute periods are also indicated in Ta-

Schedule	PPL		
	20K	30K	40K
203M			
PTFS	30.97	29.50	28.65
CPT	31.31	29.90	29.07
BSPT	**30.25**	**28.94**	**28.19**
406M			
PTFS	26.58	25.06	24.19
CPT	26.89	25.49	24.67
BSPT	**25.85**	**24.52**	**23.79**
608M			
PTFS	23.12	21.75	20.93
CPT	23.50	22.26	21.52
BSPT	**22.59**	**21.43**	**20.77**
1.2B			
PTFS	20.84	19.28	18.36
CPT	21.22	19.79	18.97
BSPT	**20.13**	**18.81**	**18.09**
2.1B			
PTFS	18.33	16.88	16.04
CPT	18.76	17.47	16.72
BSPT	**17.82**	**16.63**	**15.97**
3.1B			
PTFS	17.22	15.87	**15.07**
CPT	17.67	16.48	15.77
BSPT	**16.84**	**15.72**	15.09

Table 7: The generalization of BSPT in terms of model scaling. The model sizes range from 203M to 3.1B.

ble 6. We can draw the following conclusions: 1) The performance of absolute periods is better than that of relative periods at a lower cost. 2) **The optimal pre-training performance is achieved with an absolute period of 6K-8K steps**. Taking into account both performance and cost, we set the rapid decay period as 6K in subsequent experiments.

3.3 Generalization

Model Scaling The effectiveness of our strategy has only been verified on LLaMA-153M, but not on larger model sizes. To demonstrate the generalization of model scaling, we conduct evaluations on other 6 model sizes ranging from 203M to 3.1B. Table 11 in Appendix presents the essential hyperparameters for these model sizes, while the pre-training performance of these models are depicted in Table 7.

Schedule	PPL		
	160K	240K	320K
Cos	30.59	30.17	29.87
Const	34.61	34.10	33.78
+ RD(16K)	29.98	29.53	29.26
+ RD(32K)	**29.93**	**29.46**	**29.18**
+ RD(48K)	29.96	29.47	**29.18**
+ RD(64K)	30.02	29.50	29.35

Table 8: The generalization of BSPT in terms of data scaling. The training steps of LLMs across different versions are 160K, 240K and 320K steps. We also provide experimental results on LLaMA-1.2B in Table 14 in Appendix, which are trained for 80K, 120K, 160K steps.

Experimental results demonstrate that **our strategy consistently enhances the pre-training performance of LLMs across different model sizes**. The notable consistency provides new insights into the question of "How to configure the rapid decay period". The optimal value for the rapid decay period can be determined through systematic enumerations on smaller models.

Data Scaling To investigate the generalization in terms of data scaling, we conduct experiments on LLaMA-153M trained for 320K steps (about 336B tokens). To ensure the adequate convergence of LLMs across different versions, we also scale the rapid decay periods to 16K, 32K, 48K, and 64K steps, respectively. The experimental results are shown in Table 8, and the experimental results on LLaMA-1.2B trained for 160K steps (about 178B tokens) are listed in Table 14 in Appendix. Compared to the experimental results in Table 6, we observe similar behavior of the LLaMA-153M model when trained for 40K steps and 320K steps, respectively. Furthermore, for LLaMA-153M trained for 320K steps, the optimal rapid decay is determined to be 32K steps. This implies a lower total training cost. It indicates that **our strategy also demonstrates generalization in terms of data scaling**.

Maximum Learning Rate Compared to the cosine learning rate schedule, our learning rate schedules based on BSPT have a larger average learning rate. It motivates us to explore the following two questions: **Q1**. *Can a larger maximum learning rate improve the pre-training performance of LLMs?* **Q2**. *Is our strategy still effective when the maximum learning rate is large enough?*

Max LR	10K	20K
5e-4	44.73	38.58
1e-3	39.85	35.93
2e-3	37.67	34.71
5e-3	36.73	**34.38**
1e-2	**36.67**	34.80

Table 9: The impact of maximum learning rate for our strategy. Experiments are conducted on LLaMA-153M.

Max LR	Strategy	10k	20k
5e-3	PTFS	36.56	34.38
	CPT	36.56	34.40
	BSPT	**36.29**	**33.83**
1e-2	PTFS	36.63	34.80
	CPT	36.63	34.68
	BSPT	**36.36**	**34.26**

Table 10: The generalization of BSPT in terms of optimal maximum learning rates. Experiments are conducted on LLaMA-153M.

To provide empirical answers to these questions, we conduct experiments based on cosine leaning rate schedule. The experimental results about different maximum learning rates are presented in Table 9. For LLMs trained with 10K steps, the optimal maximum learning rate is 1e-2; while for LLMs trained with 20K steps, the optimal maximum learning rate is 5e-3. The experimental results demonstrate that, when the maximum learning rate is smaller than the optimum, **the pre-training performance of LLMs increases as the maximum learning rate is increased**.

To answer the question of Q2, we also conduct experiments of BSPT based on optimal maximum learning rates. The experimental results in Table 10 demonstrate that **BSPT still outperforms PTFS and CPT even when setting the maximum learning rate as the optimal value**. In other words, the generalization of our strategy in terms of maximum learning rate has been verified.

4 Related Work

Learning Rate Policy The learning rate is one of the most important hyper-parameters in LLMs training. Existing learning rate schedules can be broadly categorized into the following four types (Wu et al., 2019; Wu and Liu, 2023; Jin et al., 2024): 1) Fixed learning rate policy, such as constant learning rate schedule; 2) Decaying learning rate policy, such as inverse square root learning rate schedule; 3) Cyclic learning rate policy, such as cosine learning rate schedule; 4) Composite learning rate policy, such as knee and multi-step learning rate schedules. Among these policies, the cosine learning rate schedule is the most commonly used for LLMs training (Zhao et al., 2023). However, it performs poorly in terms of version iteration in LLMs. Hence, we conduct a systematic investigation into the impact of various learning rate schedules, and design a novel pre-training strategy for version iteration of LLMs.

Continual Training Continual training is a straightforward approach to address the version iteration problem in LLMs. Research related to continual training of language models can be broadly categorized into the following types: 1) New challenges in the era of LLMs (Jang et al., 2021; Cossu et al., 2022; Wang et al., 2023b); 2) Methods based on a small number of additional parameters (Song et al., 2023; PENG et al., 2024; Ke et al., 2022, 2023); 3) Prompt-based approaches (Razdaibiedina et al., 2023; Wang et al., 2022b,a); 4) Methods tailored to specific scenarios (Peng et al., 2023; Gogoulou et al., 2023a,a). These methods are often applied in constrained scenarios, such as limited computational resources or unavailability of complete data. They trade-off performance for lower training cost and are not suitable for version iteration in LLMs.

5 Conclusion

In this study, we systematically explore the optimal learning rate settings for PTFS and CPT. These approaches focus on the performance of LLMs of current version, but overlook the mutual influence of training processes across different versions. To achieve a better balance between pre-training performance and training cost, we design a new pre-training strategy for the training of LLMs of different versions. Compared with PTFS, our strategy reduces the total training cost to 58% while maintaining optimal pre-training performanc. Besides, the generalization of our strategy in model scaling, data scaling, and maximum learning rate has been verified.

Limitations

We list the main limitations of this paper as follows:

Insufficient Experiments 1. Due to limited computing resources, we don't further verify the generalization of our method on larger models and more data. 2. Limited by the length of paper, We do not provide more detailed analysis experiments, including the impact of minimum learning rate, comparation with other continual training methods and, etc.

Additional Training Cost Despite our strategy shows superiority in version iteration of LLMs, it still incurs about 30% additional training cost compared to pre-training from scratch, which can be further reduced. We will further investigate this problem in future work.

References

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. 2024. Deepseek llm: Scaling open-source language models with longtermism. arXiv.

Andrea Cossu, Tinne Tuytelaars, Antonio Carta, Lucia Passaro, Vincenzo Lomonaco, and Davide Bacciu. 2022. Continual pre-training mitigates forgetting in language and vision. arXiv.

Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, Yang Zhou, Kaizhao Liang, Jintai Chen, Juanwu Lu, Zichong Yang, Kuei-Da Liao, et al. 2023. A survey on multimodal large language models for autonomous driving. arXiv.

Evangelia Gogoulou, Timothée Lesort, Magnus Boman, and Joakim Nivre. 2023a. A study of continual learning under language shift. arXiv.

Evangelia Gogoulou, Timothée Lesort, Magnus Boman, and Joakim Nivre. 2023b. A study of continual learning under language shift. arXiv.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. arXiv.

Kshitij Gupta, Benjamin Thérien, Adam Ibrahim, Mats L. Richter, Quentin Anthony, Eugene Belilovsky, Irina Rish, and Lesort Timothée. 2023. Continual pre-training of large language models: How to (re)warm your model? In *ICML Workshop*.

Nikhil Iyer, V Thejas, Nipun Kwatra, Ramachandran Ramjee, and Muthian Sivathanu. 2023. Wide-minima density hypothesis and the explore-exploit learning rate schedule. *Journal of Machine Learning Research*.

Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, Stanley Jungkyu Choi, and Minjoon Seo. 2021. Towards continual knowledge learning of language models. arXiv.

Hongpeng Jin, Wenqi Wei, Xuyu Wang, Wenbin Zhang, Hongpeng Wu, YanzhaoJin, Wenqi Wei, Xuyu Wang, Wenbin Zhang, and Yanzhao Wu. 2024. Rethinking learning rate tuning in the era of large language models. arXiv.

Zixuan Ke, Haowei Lin, Yijia Shao, Hu Xu, Lei Shu, and Bing Liu. 2022. Continual training of language models for few-shot learning. In *EMNLP*.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pre-training of language models. In *ICLR*.

Bohao PENG, Zhuotao Tian, Shu Liu, Ming-Chang Yang, and Jiaya Jia. 2024. Scalable language model with generalized continual learning. In *ICLR*.

Guangyue Peng, Tao Ge, Si-Qing Chen, Furu Wei, and Houfeng Wang. 2023. Semiparametric language models are scalable continual learners. arXiv.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2023. Progressive prompts: Continual learning for language models. In *ICLR*.

Leslie N Smith. 2018. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. arXiv.

Leslie N Smith and Nicholay Topin. 2019. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*.

Chenyang Song, Xu Han, Zheni Zeng, Kuai Li, Chen Chen, Zhiyuan Liu, Maosong Sun, and Tao Yang. 2023. Conpet: Continual parameter-efficient tuning for large language models.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. arXiv.

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

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2023a. A survey on large language model based autonomous agents. arXiv.

Xiao Wang, Yuansen Zhang, Tianze Chen, Songyang Gao, Senjie Jin, Xianjun Yang, Zhiheng Xi, Rui Zheng, Yicheng Zou, Tao Gui, et al. 2023b. Trace: A comprehensive benchmark for continual learning in large language models. arXiv.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. 2022a. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *ECCV*.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022b. Learning to prompt for continual learning. In *CVPR*.

Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and Philip S. Yu. 2023. Multimodal large language models: A survey. arXiv.

Yanzhao Wu and Ling Liu. 2023. Selecting and composing learning rate policies for deep neural networks. *ACM Transactions on Intelligent Systems and Technology*.

Yanzhao Wu, Ling Liu, Juhyun Bae, Ka-Ho Chow, Arun Iyengar, Calton Pu, Wenqi Wei, Lei Yu, and Qi Zhang. 2019. Demystifying learning rate policies for high accuracy training of deep neural networks. In *IEEE International conference on big data*.

Yong Xie, Karan Aggarwal, and Aitzaz Ahmad. 2023. Efficient continual pre-training for building domain specific large language models. arXiv.

Aiyuan Yang, Bin Xiao, Bingning Xiao, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023. Baichuan 2: Open large-scale language models. arXiv.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Zheng, et al. 2023. Glm-130b: An open bilingual pre-trained model. In *ICLR*.

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.

A Appendix

A.1 Different Learning Rate Schedules

We train LLMs with commonly used learning rate schedules, including constant, inverse square root, cosine, knee and multi-step learning rate schedules. LLMs of different sizes have varying maximum learning rates, which are listed in Table 11. The minimum learning rate is set to 0.1 times the maximum learning rate. We also plot the specific learning rate curves of these schedules in Figure 6.

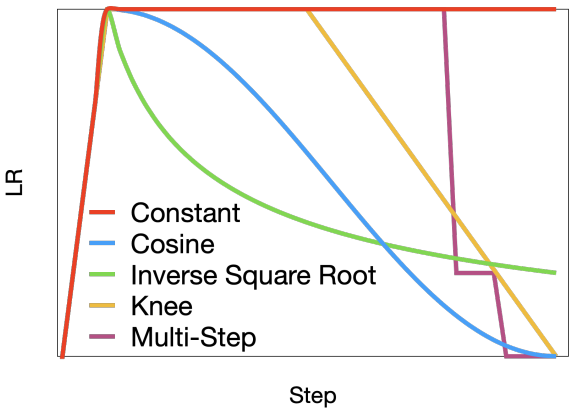

Figure 6: Learning rate curves of different schedules.

Size	LR	Hidden	Heads	Layers
153M	1e-3	512	8	12
203M	1e-3	512	8	24
406M	6e-4	1024	16	12
608M	6e-4	1024	16	24
1.2B	3e-4	1536	16	24
2.1B	3e-4	1536	16	48
3.1B	3e-4	8192	32	40

Table 11: Detailed Hyper-parameters of LLMs with different sizes.

A.2 Hyper-Parameters of LLMs

In this paper, we conduct experiments on LLMs with 7 different sizes, including LLaMA-153M, LLaMA-206M, LLaMA-406M, LLaMA-608M, LLaMA-1.2B, LLaMA-2.1B and LLaMA-3.1B. The detailed hyper-parameters are listed in Table 11.

A.3 Experiments on LLaMA-1.2B

In this section, we present a list of significant experiments conducted based on LLaMA-1.2B, including: 1. Comparison of different learning rate schedules; 2. Comparison of different strategies for training LLMs of different versions; 3. The generalization of BSPT in terms of data scaling.

Different Learning Rate Schedules Based on LLaMa-1.2B, we also conduct systematic investigation of various learning rate schedules. The experimental results are listed in Table 12, which exhibit similarities to those presented in Table 1. Compared to fixed policy and decaying policy, cyclic policy and composite policy achieve superior pre-training performance.

Schedule	PPL			
	10K	20K	30K	40K
Fixed				
Const(3e-4)	25.67	22.22	20.86	20.08
Const(3e-5)	53.19	37.13	31.36	28.23
Decaying				
Inv-Sqrt	25.62	22.15	20.71	19.84
Cyclic				
Cos	24.66	20.84	19.28	18.36
Composite				
Knee	23.79	**20.22**	**18.80**	**17.98**
Multi-Step	**23.76**	20.28	18.88	18.06

Table 12: PPLs of the most commonly used learning rate schedules on LLaMA-1.2B.

Sch.	Strategy	Cost	PPL		
			20K	30K	40K
Cos	FSPT	1.00×	20.84	19.28	18.36
	CPT	**0.44**×	21.22	19.79	18.97
	BSPT	0.58×	**20.13**	**18.81**	**18.09**
Knee	FSPT	1.00×	20.22	**18.80**	**17.98**
	CPT	**0.44**×	20.56	19.27	18.52
	BSPT	0.58×	**20.12**	18.81	18.08
Multi	FSPT	1.00×	**20.28**	**18.88**	**18.06**
	CPT	**0.44**×	20.62	19.37	18.65
	BSPT	0.58×	20.40	**18.88**	18.09

Table 13: Comparison of different strategies for training LLaMA-1.2B of different versions.

Schedule	PPL		
	80K	120K	160K
Cos	16.70	15.97	15.54
Const	18.78	18.20	17.86
+ RD(8K)	16.73	16.18	15.85
+ RD(16K)	16.53	15.96	15.63
+ RD(24K)	16.47	15.86	15.51
+ RD(32K)	**16.43**	**15.79**	**15.44**

Table 14: The generalization of BSPT in terms of data scaling. The training steps of LLMs across different versions are 80K, 120K and 160K steps.

Different Strategies Table 13 lists the experimental results of PTFS, CPT and BSPT, which is similar to that in Table 5. Compared with PTFS, our strategy reduces the total training cost to 58% while maintaining pre-training performanc.

Data Scaling To further verify the generalization of our strategy, we conduct experiments by training LLaMA-1.2B for 160K steps (178B tokens). The experimental results are listed in Table 14, which exhibit similarities to those presented in Table 8.