How to Set the Learning Rate for Large-Scale Pre-training?

Yunhua Zhou[1*] **Shuhao Xing**[2,1 *] **Junhao Huang**[3,1] **Xipeng Qiu**[2†] **Qipeng Guo**[1†]
[1]Shanghai AI Laboratory, China
[2]Fudan University, China
[3]Shanghai JiaoTong University, China
{zhouyunhua, xingshuhao.dispatch}@pjlab.org.cn

Abstract

Optimal configuration of the learning rate (LR) is a fundamental yet formidable challenge in large-scale pre-training. Given the stringent trade-off between training costs and model performance, the pivotal question is whether the optimal LR can be accurately extrapolated from low-cost experiments. In this paper, we formalize this investigation into two distinct research paradigms: Fitting and Transfer. Within the Fitting Paradigm, we innovatively introduce a Scaling Law for search factor, effectively reducing the search complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n \cdot C_D \cdot C_\eta)$ via predictive modeling. Within the Transfer Paradigm, we extend the principles of μTransfer to the Mixture of Experts (MoE) architecture, broadening its applicability to encompass model depth, weight decay, and token horizons.

By pushing the boundaries of existing hyperparameter research in terms of scale, we conduct a comprehensive comparison between these two paradigms. Our empirical results challenge the scalability of the widely adopted μTransfer in large-scale pre-training scenarios. Furthermore, we provide a rigorous analysis through the dual lenses of training stability and feature learning to elucidate the underlying reasons why module-wise parameter tuning underperforms in large-scale settings. This work offers systematic practical guidelines and a fresh theoretical perspective for optimizing industrial-level pre-training.

1 Introduction

The rapid evolution of Large Language Models (LLMs) (OpenAI et al., 2024c,a,b, 2025; DeepSeek-AI et al., 2024a,b, 2025b,a,c) is continuously pushing the cognitive boundaries of artificial intelligence, driven fundamentally by the Scaling Laws (Kaplan et al., 2020) arising from large-scale pre-training. However, executing such large-scale pre-training remains formidable, A fundamental challenge is selecting an appropriate/optimal learning rate (LR). On one hand, large-scale pre-training involves massive computational loads and prolonged training cycles, requiring a precise LR to ensure both stability and convergence efficiency. On any other hand, the vast consumption of computational resources makes the cost of trial-and-error unacceptable. Consequently, **the crux of learning rate for large-scale pre-training lies in accurately characterizing the relationship between the optimal LR in "cheaper-to-train" small-scale experiments and that of the target scale.**

This paper establishes two fundamental research paradigms for setting the learning rate in large-scale pre-training: **Fitting** and **Transfer**. The Fitting Paradigm involves directly modeling the relationship between the optimal learning rate, model size, and training data under standard initialization conditions, thereby extrapolating the learning rate for the target training scale (DeepSeek-AI et al., 2024a; Li et al., 2025). To overcome the bottlenecks of combinatorial explosion and prohibitive training costs inherent in prior research within the fitting paradigm, this work innovatively introduces a scaling Law for search factor. By leveraging performance prediction, we effectively reduce the search complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n \cdot C_D \cdot C_\eta)$.

The Transfer Paradigm, on the other hand, conducts hyperparameter optimization (including learning rate) on selected proxy models and subsequently transfers these hyperparameters to the target model according to established rules. In this study, we adopt the fundamental principles of μTransfer (Yang et al., 2022). However, to better align with contemporary large-scale pre-training scenarios, we implement a critical extension by selecting the Mixture of Experts (MoE) as our research architecture. Building upon existing litera-

[*]Equal contribution. Orders are determined randomly.
[†]Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 37344–37361

ture, we expand the transfer dimensions to encompass model widths and depths, while simultaneously incorporating the influences of weight decay and token horizon on μTransfer. These enhancements substantially push the boundary of applicability for μTransfer.

To facilitate a comprehensive comparison between the two paradigms, this study extends the target prediction scale for learning rates by more than tenfold (x10). Our target configuration is set as a MoE model with 12B total parameters with 1.3B activated for each token, trained on 500B tokens—a scale that significantly surpasses existing hyperparameter research. The primary contributions of this paper can be summarized in the following three aspects:

Paradigm and Theoretical Innovation: We systematically formalize the two research paradigms and innovatively integrate Scaling Laws for performance prediction. This approach effectively reduces modeling costs while substantially enhancing both prediction efficiency and the range of parameter coverage.

Ultra-Large-Scale Empirical Comparison: Breaking through the scale limitations of prior hyperparameter studies, this work provides the first comprehensive comparison of the two paradigms within a real-world, large-scale pre-training environment, offering systematic practical guidelines for large-model engineering.

Multidimensional Mechanistic Insights: We provide an in-depth analysis of the dynamical characteristics of both paradigms during pre-training, focusing on two core dimensions: Training Stability and Feature Learning. This offers a novel perspective for research into large-scale pre-training.

2 Related Works

2.1 Learning Rate Schedule

Prior to the advent of large-scale language model (LLM) pre-training, the cosine annealing schedule (Loshchilov and Hutter, 2017) served as the predominant standard. However, the cosine schedule mandates a predetermined number of total training steps, rendering it insufficiently flexible amidst the backdrop of continuously expanding pre-training scales. Consequently, the Warmup-Stable-Decay (WSD) schedule (Hu et al., 2024) has emerged. This schedule is characterized by a stable phase where the learning rate remains constant following the warmup period, eventually decaying to a specific terminal value. Since the decay phase can be initiated at any point during the stable phase to conclude training, WSD is regarded as highly adaptable to the dynamic requirements of large-scale pre-training. Reflecting this advantage, the WSD scheduler has recently been adopted by mainstream large-scale pre-training projects(DeepSeek-AI et al., 2025b; Team et al., 2025; Bai et al., 2025).

Propelled by scaling laws, the magnitude of pre-training continues to escalate. The stable phase frequently spans weeks or even months (DeepSeek-AI et al., 2025b; Bai et al., 2025; Yang et al., 2025), making the precise configuration of the learning rate critically important. However, existing research on learning rate configuration has predominantly focused on the cosine annealing schedule. Under the cosine regime, Kaplan et al. (2020) elucidated the relationship between the learning rate and model parameters, while Bjorck et al. (2025) and Li et al. (2025) empirically derived power-law formulations correlating the learning rate with model size N and training data size D. Diverging from existing literature, our work investigates the relationship between the optimal learning rate, model size, and training data size specifically within the stable phase of a constant learning rate schedule.

2.2 Maximal Update Parametrization

Maximal Update Parametrization (μParametrization or μP, Yang et al. (2022)) is a widely investigated framework for hyperparameter configuration. The fundamental premise of μP is to guarantee training stability and ensure that weights across different modules are adequately trained(i.e. maximal feature learning) even as model width approaches infinity.

By virtue of maintaining these properties in the infinite-width limit, μP possesses inherent capabilities for hyperparameter transfer. This gives rise to a derivative method known as μTransfer, wherein the optimal learning rate for a target model can be directly calculated based on the optimum identified via search on a smaller proxy model. While the initial formulation of μTransfer was limited to extrapolating model width, subsequent studies by Yang et al. (2023) and Dey et al. (2025) have investigated extensions for scaling model depth. Beyond its extensive application in dense architectures(Lingle, 2025), μTransfer has also been experimentally applied to Mixture-of-Experts

(MoE) structures(Małaśnicki et al., 2025). Furthermore, recent research indicates that the efficacy of μTransfer is primarily manifested during the early stages of training; to extend the effective transfer horizon, adjustments to weight decay are required(Wang and Aitchison, 2025; Małaśnicki et al., 2025; Fan et al., 2025).

Building upon existing research of μTransfer and integrating current methodologies for pre-training hyperparameter configuration, our work conducts a granular investigation into the impact of μTransfer on the performance of large-scale pre-training.

3 Approach

This section delineates the specific methodologies for the configuration of the learning rate under two distinct paradigms. Section 3.1 introduces the Fitting Paradigm, which leverages scaling laws to enhance the efficiency and scope of the fitting process. Section 3.2 focuses on the representative transfer paradigm μTransfer method and elucidating its practical implementation within large-scale pre-training contexts. Crucially, our study focuses on the stable training phase governed by the Warmup-Stable-Decay (WSD) learning rate schedule.

3.1 Scaling Laws for Learning Rate

For a given model size N and training data size D , the optimal learning rate η is formulated as:

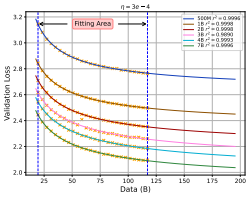
$$\eta^*_{ND} = \underset{\eta}{\operatorname{argmin}}\, L(\eta \mid N, D, \Theta), \tag{1}$$

where L is validation loss and Θ contains other hyperparameters involved in the pre-train process.

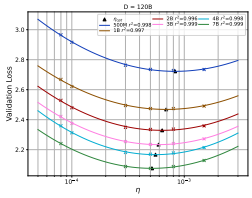
Characterizing the relationship between the optimal η, model size, and data size necessitates a grid search across the N, D, and η dimensions, resulting in a computational complexity of approximately $\mathcal{O}(n^3)$. Fortunately, inspired by prior work (Bjorck et al., 2025), we observe that for fixed N and D, the relationship between the validation loss L and the learning rate η approximates an invex profile. Consequently, we employ a quadratic polynomial to fit this relationship:

$$L(\eta \mid N, D, \Theta) = L_{min} + C \cdot (\log(\eta) - \eta_{min})^2, \tag{2}$$

where L_{min}, C, and η_{min} are the fitting coefficients.



(a) Fitting results for Equation 4. Data points to the left of the dashed line represent the empirical values used for fitting; The curves to the right side depicts predictions. See AppendixA.2 for the discussion on the accuracy of Equation 4.



(b) Results of fitting the validation loss against the learning rate (LR) using a **quadratic polynomial**. Different colored curves correspond to models of varying sizes, while the triangle indicate the optimal LR.

Figure 1: Results of Equation 4 and 2. These approaches allow for a substantial reduction in the time and storage cost of the search process.

Consequently, for a given N and D, the optimal learning rate η^* can be directly derived via fitting on a limited set of learning rates:

$$\log(\eta^*) = \eta_{min} = \underset{\eta}{\operatorname{argmin}}\{L(\eta \mid N, D, \Theta)\}, \tag{3}$$

Figure 1(b) shows the fitted curves of Equation 2. The search complexity is reduced from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2 * C_\eta)$

Furthermore, inspired by contemporary research on scaling laws(Hoffmann et al., 2022; Tissue et al., 2024), we observe that under the WSD schedule, the validation loss exhibits a power-law relationship with the training data size D for a fixed model configuration:

$$L(D) = L_0 + A \cdot D^{-\gamma}, \tag{4}$$

Where L_0, A, γ are parameters to fit. This implies that the search space along the dimension of data size D can be significantly compressed, enabling the extrapolation of results to larger data regimes via a limited number of search points. The specific fitting procedure is illustrated in Figure 1(a). This methodology effectively improves the trade-off between search cost and fitting precision, reducing the computational complexity of the search from $\mathcal{O}(n^3)$ to $\mathcal{O}(n \cdot C_D \cdot C_\eta)$.

Based on Equation 2 and 3, by conducting a search within the N dimension, we can efficiently derive a comprehensive set of optimal LR $\{\eta^*_{ND}\}_{N,D}$, corresponding to varying model sizes N and data sizes D. This facilitates the fitting of the functional relationship between the optimal LR and the variables N and D:

$$Lr(N, D) = \underset{Lr \in \mathcal{F}}{\operatorname{argmin}} L(Lr(N, D), \eta^*_{ND} \mid \Theta), \tag{5}$$

where $\mathcal{F}$ represents the candidate function space, and L denotes the metric function, which is Root Mean Squared Error (RMSE) in our work.

The final fitted relationship governing the optimal learning rate with respect to model size N and data size D is given by:

$$Lr(N, D) = 38.4588 \cdot N^{-0.2219} \cdot D^{-0.3509}. \tag{6}$$

We observe a good fit with $R^2 \approx 0.9622$(See Appendix A.3.1 for details of fitting process). The overall fitting results are shown in Figure 2.

Extending this approach, we further conduct a fine-grained investigation into the learning rate configurations for distinct model modules in Section 6.1.

3.2 Scaling μTransfer for Pre-training

As the Mixture-of-Experts (MoE) architecture increasingly serves as the foundational backbone for large-scale pre-training (DeepSeek-AI et al., 2024b, 2025b,a,c; Yang et al., 2024; Qwen et al., 2025; Yang et al., 2025; Bai et al., 2025), we adopt the MoE architecture as our proxy model for μP. Regarding the target model, we adhere to the settings proposed by Małaśnicki et al. (2025) for initialization along the width dimension. For the depth dimension, we draw upon the methodologies of Depth-up (Yang et al., 2023) and Complete-μP (Dey et al., 2025; Mlodozeniec et al., 2025). The central mechanism involves applying a depth-dependent scaling factor to the residual branch:

$$H^{i+1} = H^i + m_L^{-\alpha} \mathcal{F}(H^i), \quad i \in \{1, \ldots, L\}, \tag{7}$$

where H^i denotes the output of the i-th layer, and $\mathcal{F}$ represents either the Attention or Feed-Forward Network (FFN) layer. Following the recommendations of Complete-μP, we set $\alpha = 1$ to enhance the transferability of μTransfer.

Wang and Aitchison (2025) and Fan et al. (2025) have identified weight decay λ as a critical determinant of μTransfer efficacy. Consequently, we incorporate the influence of weight decay into the training process of the target model, maintaining the proportionality $\delta\lambda \propto \delta lr$. For given model size N and data volume D, we observe that the approximate invex relationship between validation loss L and learning rate persists within μP proxy models. This observation allows for a reduction in the search space along the learning rate dimension, thereby improving the efficiency of μTransfer. Regarding transfer along the token horizon dimension, we adopt the configuration from Mlodozeniec et al. (2025). The detailed initialization and transfer rules for the target model parameters are summarized in Table 2 and Table 10.

4 Experiments

4.1 Datasets

The pre-training corpus utilized in our work is derived from InternLM2.5 (Cai et al., 2024), including general text, source code, and long-context sequences. Specifically, the textual component spans web pages, academic papers, patents, and books. The code component is primarily sourced from GitHub, programming communities, and other public repositories, covering a diverse array of programming languages including C/C++, Java, and Python. All data underwent rigorous deduplication and safety filtering protocols.

To ensure distributional consistency, the validation set employed in our experiments was constructed via random sampling from the above corpus, while strictly maintaining disjointness from the training samples to prevent data leakage.

4.2 Experimental Settings

We adopt the Qwen3-MoE (Yang et al., 2025) architecture for our experimental models. For all model training, we utilize the AdamW optimizer (Loshchilov and Hutter, 2019) with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and $\epsilon = 10^{-8}$. The learning rate schedule consists of a linear warmup for 1,000 steps, followed by a constant learning rate strategy. The sequence length is fixed at 4,096, and the global batch size is set to 4M tokens.

For experiments in 3.1, we employ models of four distinct sizes (550M, 1B, 2B, and 3B) all adhering to the structural configuration of the Qwen3-30B-A3B model. Notably, the aspect ratio between model width and depth remains constant across these scales. We subsequently validate our experimental findings on target models with 4B, 12B total parameters. With the exception of normalization parameters, all model weights are initialized from a normal distribution with a standard devia-

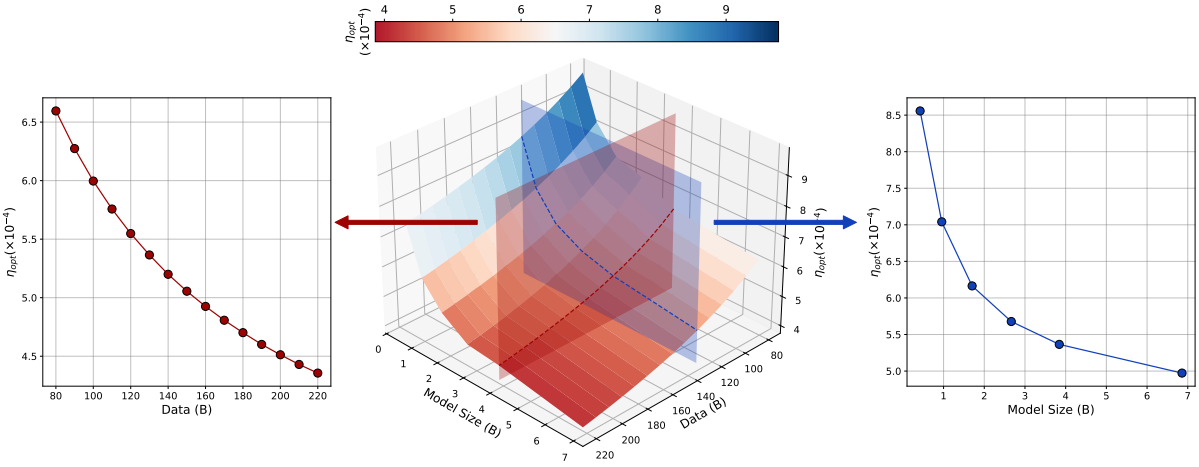


Figure 2: **Middle**: Visualization of the optimal learning rate relative to model size N and data size D. **Left**: The relationship between the optimal learning rate and data size D with model size fixed at $N = 4$B. **Right**: The relationship between the optimal learning rate and model size N with data size fixed at $D = 140$B.

tion of 0.02. The search space for the learning rate is defined as $\eta \in \{8e-5, 1e-4, 3e-4, 5e-4, 8e-4, 1.5e-3, 2e-3\}$. Each model is trained on approximately 120B tokens (30,000 steps), and we extrapolate the results to 500B tokens using Equation 4. Weight decay is set to 0.1.

For the μTransfer experiments, we conduct learning rate and initialization searches on a proxy model with 2B total parameters. The search space for the learning rate is defined as $\eta \in \{8e-5, 1e-4, 3e-4, 5e-4, 8e-4, 1.5e-3, 2e-3\}$; for initialization, we explore the range $\sigma \in \{0.0005, 0.001, 0.002, 0.005, 0.01, 0.015, 0.02\}$. The actual training data size for these experiments is approximately 200B tokens (50,000 steps), and we extrapolate the hyperparameters to a 500B token regime with Equation 4 and settings from Mlodozeniec et al. (2025).

4.3 Evaluation

To assess the downstream performance of the models developed during our validation experiments, we evaluate our models on MMLU (Hendrycks et al., 2021) and CMMLU (Li et al., 2024) benchmarks. MMLU serves as our primary English evaluation set, comprising four-choice multiple-choice questions across 57 distinct subjects, including anatomy, physics, genetics etc. Conversely, we employ CMMLU to evaluate Chinese language proficiency which covers 67 domains ranging from natural sciences and humanities to general knowledge.

For the implementation of these evaluations, we leverage the OpenCompass framework (Contributors, 2023b), a comprehensive Python toolkit designed to facilitate the batch evaluation of diverse foundation models across heterogeneous datasets. Furthermore, to expedite the evaluation pipeline, we utilize the LMDeploy framework (Contributors, 2023a) with Turbomind (Zhang et al., 2025) backend for efficient model loading and inference acceleration.

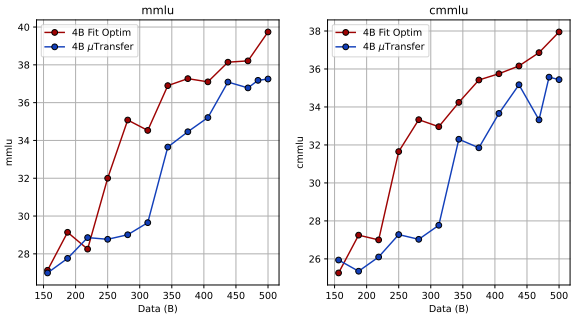


Figure 3: Downstream task performances of 4B model with global optimal LR and μP respectively.

5 Results

First, we extrapolate the proxy model solely by increasing its width, scaling it to 4B total parameters with 530M active parameters, and conducting from-scratch pre-training on 500B tokens. To rigorously assess the pre-training quality under both paradigms, we evaluate not only the final model performance but also the downstream task results throughout the training process. The performance trends are illustrated in Figure 3. As shown, the pre-training quality achieved by the Fitting Paradigm

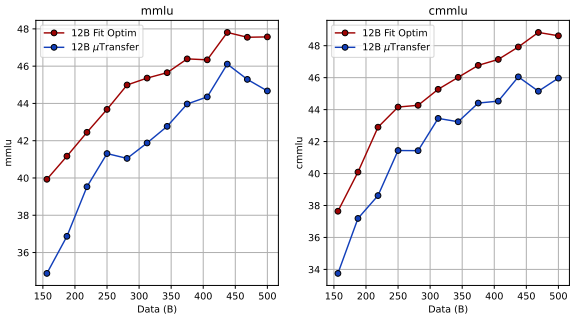


Figure 4: Downstream task performances of 12B model with global optimal LR and μP respectively.

is significantly higher than that of $\mu Transfer$.

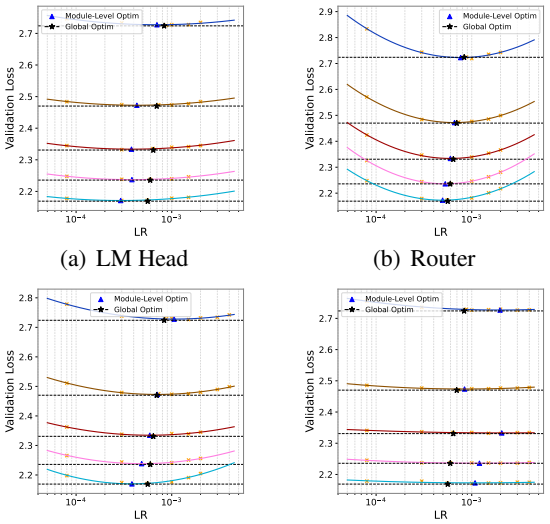
Furthermore, we extend the predictive training scale by more than an order of magnitude, scaling the model to 12B total parameters (1.3B active) and pre-training it from scratch on 500B tokens. We similarly evaluate the intermediate progress, with the overall performance trajectories presented in Figure 4. As demonstrated in Figure 4, the model trained via the Fitting Paradigm consistently and significantly outperforms the one using $\mu Transfer$.

6 Analyze

6.1 Module-Level Optimal Learning Rates

A fundamental motivation behind μP is the hypothesis that under Standard Parametrization (SP) and a uniform global learning rate, specific modules may suffer from insufficient training, thereby failing to satisfy the regime of maximal feature learning. To investigate this, building upon the global optimal learning rate derived from our fitting paradigm in Section 3.1, we employ a greedy search strategy to conduct a fine-grained learning rate search across four distinct parameter modules: Embeddings, LM Head, Router, and Hidden parameters.We observe that fine-grained tuning of individual modules yields no significant performance improvement compared to the global optimal learning rate configuration.

The optimal learning rates identified for specific modules align closely with the global optimum, and the minimum loss achieved through module-specific search exhibits negligible deviation from the loss achieved with global optimal learning rate from Equation 6 (as illustrated in Figure 5). Consequently, assigning distinct optimal learning rates to specific modules does not appear to materially enhance model performance.



(a) LM Head (b) Router

(c) Hidden (d) Embeddings

Figure 5: The relationship between loss and learning rate for **(a)LM Head**, **(b)Router**, **(c)Hidden** and **(d)Embedding** parameters during the module-level learning rate search. Each curve corresponds to a model of a specific size. The dashed lines indicate the loss achieved by the corresponding model size under the global optimal learning rate setting. Triangle markers denote the optimal learning rate for the current module, while star markers represent the global optimal learning rate.

To further validate the effect of module-level optimal LR, we trained the target 4B model for 120B tokens using both the derived module-specific optimal LR and the calculated global optimal LR. As depicted in Figure 6, the comparison of the validation losses reveals that the loss curves for both settings are virtually indistinguishable. See Table 3 and Appendix A.3.2 for detailed settings.

6.2 A Closer Look at Feature Learning

In the previous subsection, we observed that fine-grained learning rate tuning across distinct model modules yielded no substantial performance gains, indicating that a global learning rate configuration does not induce training imbalances among components. In this subsection, we further investigate the feature learning dynamics of these modules by analyzing the optimization trajectory, specifically monitoring the evolution of parameter update magnitudes throughout the training process.

As illustrated in the Figure7, training with the AdamW optimizer results in parameter update magnitudes that remain stable over extended periods and exhibit relative uniformity across layers. The update magnitudes consistently approximate 0.2, a finding consistent with recent theoretical studies

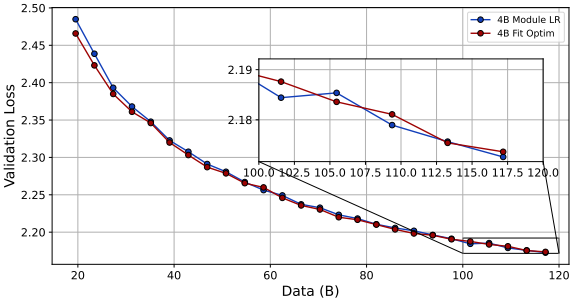


Figure 6: Performance comparison between the global optimal LR (red line) and module-wise optimal LR (blue line) on a 4B model trained for 120B tokens. The two loss curves are virtually indistinguishable during the mid-to-late stages of training ($\Delta L \leq 0.01$), indicating that module-specific learning rate optimization does not yield significant performance improvements.

(Liu et al., 2025; Kosson et al., 2024). This evidence further corroborates that distinct modules maintain comparable feature learning capabilities at any given stage of training, thereby negating the necessity for module-specific learning rates to balance feature learning efficiency.

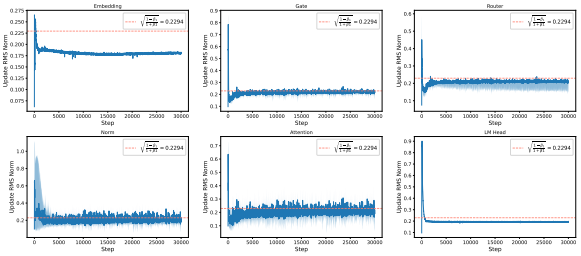
6.3 Does Standard Parametrization Scale?

Training stability is widely regarded as another distinct advantage of μP. Yang et al. (2022) argues that under standard parametrization, the internal training states of certain modules tend to "blow up" as model scale increases, thus the adjustment of learning rates on different modules is necessary.

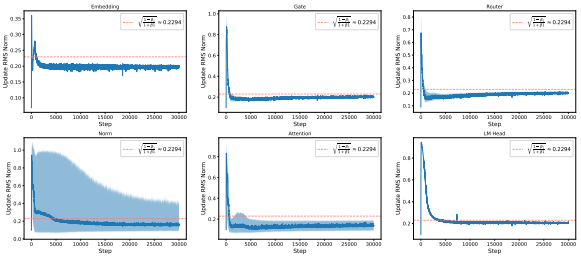
We replicated the methodology of μTransfer to analyze the model derived from our experiments. Contrary to expectations, under standard parametrization, the internal states of our model did not exhibit blow up(Figure 8(a)); rather, they displayed trends remarkably similar to those of models initialized via μP. To investigate further, we conducted an ablation where the QK-Norm modules were removed during the computation of attention logits (Figure 9). Under this condition, we successfully reproduced the instability trends described in μTransfer. Consequently, we posit that recent advancements in model architecture—such as the incorporation of QK-Norm—have rendered layer-wise training more balanced and significantly enhanced robustness to hyperparameter variations.

6.4 Impact of Data Size on Training Stability

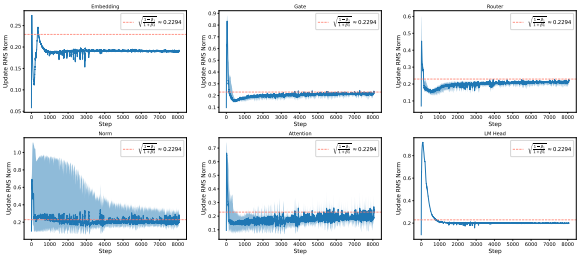
Existing research on training stability, most notably the work on μP, has predominantly focused on model scale while neglecting the influence of



(a) 4B Model

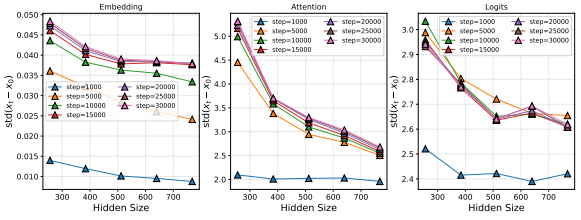


(b) 500M Model without QK-Norm

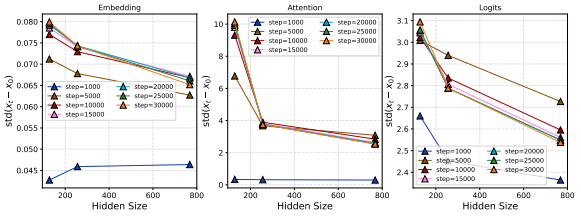


(c) 4B μP Model

Figure 7: Update RMS norm during the training process of: (a) 4B model (b) 500M model with QK-Norm removed (c) 4B model with μP. Update RMS norm maintains approximately 0.2 in all the models we observed.

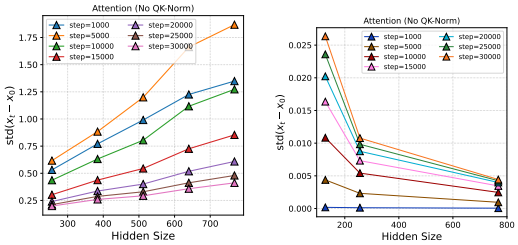


(a) 4B Model



(b) 4B μP Model

Figure 8: Variation of word embeddings, attention logits, and logits compared to initial states at certain training steps as width increases. With reference to Yang et al. (2022), we plot the standard deviation of the coordinates of $x_t - x_0$, $x \in \{word\ embeddings, attention\ logits, logits\}$. In our experiments, logits and attention logits of models with standard parametrization do not exhibit the "blow-up" tendency.



(a) 4B Model (b) 4B μP Model

Figure 9: Variation of attention logits at certain training steps as width increases. We ignored QK-Norm parameters when compute attention logits. Attention logits started to blow up with width in SP model.

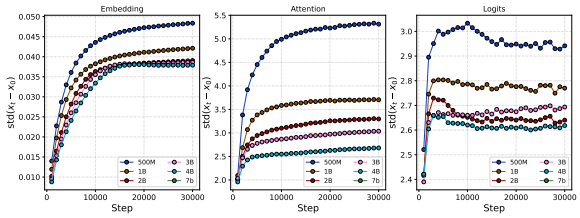


Figure 10: Variation of word embeddings, attention logits, and logits compared to initial states as training proceeded.

training data size. Employing the analytical framework established in Section 6.3, we investigate the evolution of the model's internal states as the amount of training data increases under standard parametrization.

As illustrated in the Figure 10, while the model's internal states eventually converge to a relatively stable regime as the amount of training data increases, the internal variations are significantly more drastic with respect to data scaling than to model scaling. This phenomenon is particularly evident in the attention logits. This observation offers a potential explanation for the scaling coefficients in Equation 6, where the exponent for model parameter count N (-0.22) is algebraically greater than that for data volume D (-0.35). As the size of training data expands, the magnitude of parameter updates across different modules exhibits a more pronounced increase; consequently, the optimal learning rate requires more substantial adjustment to maintain training stability.

6.5 Decay Training

A key characteristic of WSD schedule is the utilization of higher-quality training data during the decay phase after the constant-learning-rate stable phase to maximize the model's feature learning.

Building upon the experiments described in Section 5 we extended the training of both model variants after the end of the stable phase using a distinct corpus of high-quality data. We annealed the learning rate to 10% of its value during the stable phase and continued training for an additional 100B tokens. We then evaluated the downstream task performance of the models that had completed the full WSD training.

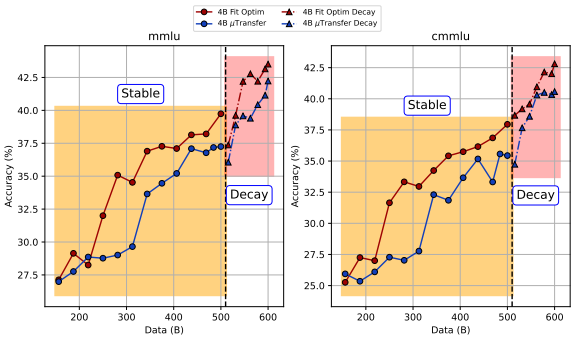


Figure 11: Downstream task performances of 4B model after the end of decay phase. The left area depicts the stable phase of training, while the right area corresponds to the decay phase.

As shown in Figure 11, the model trained with the global optimal learning rate outperformed the model derived from μ Transfer on both the MMLU and CMMLU benchmarks, achieving accuracy improvements of 1.28% (42.23% $\rightarrow$ 43.51%) and 2.23% (40.58% $\rightarrow$ 42.81%), respectively. These results demonstrate that the global optimal learning rate yields superior performance in realistic pre-training scenarios.

7 Conclusion

This paper systematically establishes two fundamental research paradigms—Fitting and Transfer—to address the critical challenge of learning rate configuration in large-scale pre-training. At the methodological level, we introduce scaling Laws to reduce the complexity of the Fitting Paradigm, and provide a comprehensive extension of μTransfer across model architectures, depths, weight decay, and token horizons. Through extensive experimentation, we challenge the scalability of the widely adopted μTransfer in large-scale pre-training scenarios and provide an in-depth analysis of the underlying mechanisms that limit the performance of module-wise parameter tuning at scale. This research offers both systematic practical guidance and a novel theoretical perspective for optimizing industrial-level pre-training.

Limitations

To inform and inspire future research, we summarize the limitations of our work as follows:

Learning Rate Schedules: This study focuses on large-scale pre-training, where the Warm-Stable-Decay (WSD) scheduler is currently the industry standard. Consequently, our analysis is centered on this specific schedule and does not explore the dynamics of other learning rate schedulers.

Model Architectures: Given that the Mixture of Experts (MoE) architecture has become the foundational backbone for modern large-scale pre-training, it served as the primary subject of our investigation. The generalizability of our findings to Dense architectures remains to be verified in future work.

Extrapolation Limits: Due to computational resource constraints, this study did not investigate the ultimate extrapolation boundaries (i.e., the maximum scale at which these predictions remain accurate) for both the Fitting and Transfer paradigms.

Use of AI Assistants

We primarily use AI assistants to improve and enrich our writing.

References

Lei Bai, Zhongrui Cai, Yuhang Cao, Maosong Cao, Weihan Cao, Chiyu Chen, Haojiong Chen, Kai Chen, Pengcheng Chen, Ying Chen, Yongkang Chen, Yu Cheng, Pei Chu, Tao Chu, Erfei Cui, Ganqu Cui, Long Cui, Ziyun Cui, Nianchen Deng, and 158 others. 2025. Intern-s1: A scientific multimodal foundation model. *Preprint*, arXiv:2508.15763.

Johan Bjorck, Alon Benhaim, Vishrav Chaudhary, Furu Wei, and Xia Song. 2025. Scaling optimal lr across token horizons. *Preprint*, arXiv:2409.19913.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, and 81 others. 2024. Internlm2 technical report. *Preprint*, arXiv:2403.17297.

LMDeploy Contributors. 2023a. Lmdeploy: A toolkit for compressing, deploying, and serving llm. https://github.com/InternLM/lmdeploy.

OpenCompass Contributors. 2023b. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

DeepSeek-AI, :, Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, and 69 others. 2024a. Deepseek llm: Scaling open-source language models with longtermism. *Preprint*, arXiv:2401.02954.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, and 138 others. 2024b. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *Preprint*, arXiv:2405.04434.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025b. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, and 245 others. 2025c. Deepseek-v3.2: Pushing the frontier of open large language models. *Preprint*, arXiv:2512.02556.

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

Zhiyuan Fan, Yifeng Liu, Qingyue Zhao, Angela Yuan, and Quanquan Gu. 2025. Robust layerwise scaling rules by proper weight decay tuning. *Preprint*, arXiv:2510.15262.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

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, and 3 others. 2022. Training compute-optimal large language models. *Preprint*, arXiv:2203.15556.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zheng Leng Thai, Kaihuo Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, and 6 others. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *Preprint*, arXiv:2404.06395.

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

Atli Kosson, Bettina Messmer, and Martin Jaggi. 2024. Rotational equilibrium: how weight decay balances learning across neural networks. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024. CMMLU: Measuring massive multitask language understanding in Chinese. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11260–11285, Bangkok, Thailand. Association for Computational Linguistics.

Houyi Li, Wenzhen Zheng, Qiufeng Wang, Hanshan Zhang, Zili Wang, Shijie Xuyang, Yuantao Fan, Zhenyu Ding, Haoying Wang, Ning Ding, Shuigeng Zhou, Xiangyu Zhang, and Daxin Jiang. 2025. Predictable scale: Part i, step law – optimal hyperparameter scaling law in large language model pretraining. *Preprint*, arXiv:2503.04715.

Lucas Lingle. 2025. An empirical study of μp learning rate transfer. *Preprint*, arXiv:2404.05728.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, and 9 others. 2025. Muon is scalable for llm training. *Preprint*, arXiv:2502.16982.

Ilya Loshchilov and Frank Hutter. 2017. Sgdr: Stochastic gradient descent with warm restarts. *Preprint*, arXiv:1608.03983.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Jan Małaśnicki, Kamil Ciebiera, Mateusz Boruń, Maciej Pióro, Jan Ludziejewski, Maciej Stefaniak, Michał Krutul, Sebastian Jaszczur, Marek Cygan, Kamil Adamczewski, and Jakub Krajewski. 2025. μ-parametrization for mixture of experts. *Preprint*, arXiv:2508.09752.

Bruno Mlodozeniec, Pierre Ablin, Louis Béthune, Dan Busbridge, Michal Klein, Jason Ramapuram, and Marco Cuturi. 2025. Completed hyperparameter transfer across modules, width, depth, batch and duration. *Preprint*, arXiv:2512.22382.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, and 108 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024a. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, and 244 others. 2024b. Openai o1 system card. *Preprint*, arXiv:2412.16720.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024c. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, and 150 others. 2025. Kimi k2: Open agentic intelligence. *Preprint*, arXiv:2507.20534.

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

Xi Wang and Laurence Aitchison. 2025. How to set adamw's weight decay as you scale model and dataset size. *Preprint*, arXiv:2405.13698.

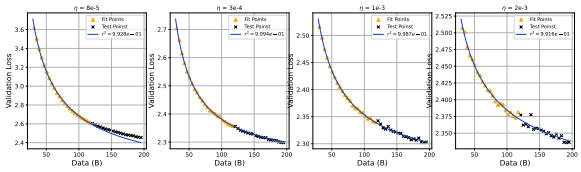


Figure 12: The precision of Equation 4.

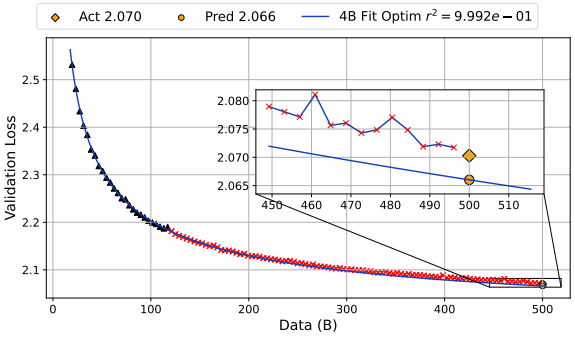


Figure 13: The precision of Equation 4 on 4B Fit Optim model.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

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

Greg Yang, Dingli Yu, Chen Zhu, and Soufiane Hayou. 2023. Tensor programs vi: Feature learning in infinite-depth neural networks. *Preprint*, arXiv:2310.02244.

Li Zhang, Youhe Jiang, Guoliang He, Xin Chen, Han Lv, Qian Yao, Fangcheng Fu, and Kai Chen. 2025. Efficient mixed-precision large language model inference with turbomind. *arXiv preprint arXiv:2508.15601*.

A Appendix

A.1 Model Architectures and LR Settings

Table 1, 2 and 3 shows the detailed parameters of models' architectures and learning rate settings.

A.2 Extrapolation of L(D)

In the experiments presented in our works, $L(D) = L_0 + A \cdot D^{-\gamma}$(Equation 4) is repeatedly employed for curve fitting and data augment. To validate the effectiveness of this approach, we continue training the 2B proxy model of μTransfer from 120B to approximately 200B tokens (50,000 steps). Points corresponding to $D \leq 120B$ are used as the fitting set, while the remaining data serve as the test set. The fitted curve is illustrated in Figure 12.

Among the four settings, the extrapolation accuracy is notably poorer under the learning rate of 8e-5, while the predicted values for the other learning rates at 200B tokens align closely with the ground truth. We attribute this discrepancy to the fact that the learning rate of 8e-5 is excessively small for the current model, causing the model state to evolve too gradually as data volume increases. By 120B tokens, the model has yet to exhibit a clear trend toward convergence. Consequently, the curve fitted by Equation 4 declines sharply rather than gradually flattening, leading to a misinterpretation of the model's future trajectory. In contrast, the other tested learning rates are relatively larger and closer to the model's optimal learning rate, enabling the validation loss curve to enter the convergence phase more rapidly and thus yielding more accurate predictions from the L(D) curve.

For experiments conducted in Section 5, we also conduct the same experiment on 4B Model, fitting with points that corresponding to $D \leq 120B$. Figure 13 indicates that despite only approximately one-quarter of the data is used for fitting, the predicted value at $D = 500B(2.066)$ exhibits a negligible discrepancy from the actual value (2.070). Therefore, we consider the method of data extrapolation via Equation 4 to be reasonable within the data range discussed in this paper.

A.3 Details of Fitting Experiments

A.3.1 Global Optimal Learning Rate

This subsection contains the whole fitting process of Section 3.1.

The validation loss curves for various models under different learning rates, derived from the global optimal learning rate search experiments, are illustrated in Figure 14. We utilize the smoothed

Table 1: Overview of Qwen3-MoE Model Architectures and Hyperparameters in Section 3.1

Models	Total Params	Activate Params	Hidden Size	Num Layers	Attn Heads	KV Heads	Interm. Size	Learning Rate
				Training Set				
Qwen3-MoE-0.5B-A0.1B	550M	100M	256	3	32	4	768	8e-5, 1e-4, 3e-4, 5e-4, 8e-4, 1.5e-3, 2e-3
Qwen3-MoE-1B-A0.2B	1B	190M	384	9	32	4	768	8e-5, 1e-4, 3e-4, 5e-4, 8e-4, 1.5e-3, 2e-3
Qwen3-MoE-2B-A0.3B	2B	280M	512	12	32	4	768	8e-5, 1e-4, 3e-4, 5e-4, 8e-4, 1.5e-3
Qwen3-MoE-3B-A0.4B	3B	400M	640	15	32	4	768	8e-5, 1e-4, 3e-4, 5e-4, 8e-4, 1.5e-3
				Test Set				
Qwen3-MoE-4B-A0.5B	4B	530M	768	18	32	4	768	8e-5, 3e-4, 5e-4, 8e-4, 1.5e-3
Qwen3-MoE-12B-A1.3B	12B	1.3B	1280	30	32	4	768	-

Table 2: Overview of Qwen3-MoE Model Architectures and Hyperparameters in Section 3.2

Models	Total Params	Activate Params	Hidden Size	Num Layers	Attn Heads	KV Heads	Interm. Size	Learning Rate	std
Qwen3-MoE-2B-A0.3B-proxy	2B	290M	512	18	32	4	512	8e-5, 1e-4, 3e-4, 5e-4, 1e-3, 2e-3	0.01, 0.015, 0.02, 0.03, 0.04
Qwen3-MoE-4B-A0.5B	4B	530M	768	18	32	4	768	-	-
Qwen3-MoE-2B-A0.3B-proxy	2B	290M	640	18	32	4	384	1e-4, 3e-4, 5e-4, 1e-3, 2e-3	0.0005, 0.001, 0.002, 0.005, 0.01, 0.02
Qwen3-MoE-12B-A1.3B	12B	1.3B	1280	30	32	4	768	-	-

Table 3: Overview of Qwen3-MoE Model Architectures and Hyperparameters in Section 6.1

Models	Total Params	Activate Params	Hidden Size	Num Layers	Attn Heads	KV Heads	Interm. Size	Learning Rate
				Training Set				
Qwen3-MoE-0.5B-A0.1B	550M	100M	256	3	32	4	768	8e-5, 3e-4, 8.75e-4, 1e-3, 1.5e-3, 2e-3, 3e-3, 4e-3
Qwen3-MoE-1B-A0.2B	1B	190M	384	9	32	4	768	8e-5, 3e-4, 7.24e-4, 1e-3, 1.5e-3, 2e-3, 3e-3, 4e-3
Qwen3-MoE-2B-A0.3B	2B	280M	512	12	32	4	768	8e-5, 3e-4, 6.36e-4, 1e-3, 1.5e-3, 2e-3, 3e-3, 4e-3
Qwen3-MoE-3B-A0.4B	3B	400M	640	15	32	4	768	8e-5, 3e-4, 5.90e-4, 1e-3, 1.5e-3, 2e-3, 3e-3, 4e-3
Qwen3-MoE-4B-A0.5B	4B	530M	768	18	32	4	768	8e-5, 3e-4, 5.55e-4, 1e-3, 1.5e-3, 2e-3, 3e-3, 4e-3

data via Equation 4 as the input for subsequent fitting stages. Furthermore, we employ this equation to extrapolate the validation loss for each model at a training volume of 200B tokens across distinct learning rates, thereby augmenting the dataset available for fitting.

We sample validation loss data points at 10B-token intervals, ranging from 80B to 220B tokens, to facilitate subsequent analysis and curve fitting. Figure 15 illustrates the variation of validation loss as a function of the learning rate η with different model size and data size. Figure 16 presents a 3-D visualization of the relationship among loss, learning rate, and training data size. Upon observing a distinct local minimum, we employ the quadratic polynomial defined in Equation 2 to fit the data (as shown in Figure 17). The coefficients of determination (R^2) consistently exceed 0.995, enabling a precise estimation of the optimal learning rate based on these curves.

As shown in Figure 18(a) and 18(b), the global optimal learning rate exhibits a power-law relationship with both the model parameter count N and training data size D. With reference to the studies of Bjorck et al. (2025), we decide to use the following functional form:

$$\eta_{opt}(N, D) = C_\eta \cdot N^{-\alpha} \cdot D^{-\beta}, \tag{8}$$

where C_η, α, β are positive constants. After employing non-linear least squares to fit the curve, we finally get the parameters of Equation 6:

$$C_\eta \sim 38.4588, \alpha \sim 0.2219, \beta \sim 0.3509. \tag{9}$$

A.3.2 Module-Level Optimal Learning Rates

This subsection details the step-by-step process of searching Module-Level Optimal LR.

We split the model into the following four groups of parameters:

- **Embedding Parameters**, which is the word embedding layer of a model,
- **Hidden Parameters**, mainly composed of self-attention and layer norm modules,
- **Router**, which contains the router matrix and experts,
- **LM Head Parameters**, which is the unembedding output layer.

Similar to our experiments in Section A.3.2, while searching optimal LR across different module groups, the training data size in set to approximately 120B tokens. According to the results above, we can derive the global optimal LR η_{opt} of every size of model in the experiment via Equation 2:

In the following stages, we sequentially conducting experiments in the order of LM Head, Router, Hidden, and Embedding parameters with greedy search strategy.

LM Head. First, we begin with the LM Head module. By varying the learning rate η^{out} of the LM Head weights within a specified range while fixing the learning rates of all other weights to the current model's global optimal learning rate(i.e. $\eta^{emb} = \eta^{hidden} = \eta^{router} = \eta_{opt}$), we conduct the search following the method described in Section 3.1. The curve fitted using Equation 2 is shown in Figure 20, where the fitted minimum is taken as the module-level learning rate η^{out}_{opt} for LM Head(Table 5).

Router. In the second searching stage, we set $\eta^{emb} = \eta^{hidden} = \eta_{opt}, \eta^{out} = \eta^{out}_{opt}$ and search learning rate on Router layers. The results are illustrated in Figure 21 and Table 6

Hidden. Next, set $\eta^{emb} = \eta_{opt}, \eta^{router} = \eta^{router}_{opt}, \eta^{out} = \eta^{out}_{opt}$ while searching optimal learning rate on Hidden parameters to obtain η^{hidden}_{opt}. The results are illustrated in Figure 22 and Table 7

Embedding. Finally, we set $\eta^{router} = \eta^{router}_{opt}, \eta^{hidden} = \eta^{hidden}_{opt}, \eta^{out} = \eta^{out}_{opt}$ and conduct learning rate searching on Embedding layer and get its optimal learning rate η^{emb}_{opt}. The results are illustrated in Figure 23 and Table 8

Overall Results. The overall results of module-level optimal learning rate are shown in Table 9

A.3.3 μTransfer

We refer to Mlodozeniec et al. (2025) to conduct our μTransfer experiments. The transfer method is shown in Table 10. As we maintain an invariant batch size across all experimental configurations, we only consider the influence of training token counts alongside model width and depth when employ μTransfer.

Table 4: Global Optimal LR at 120B

N	**500M**	**1B**	**2B**	**3B**	**4B**
η_{opt}	8.75e-4	7.24e-4	6.36e-4	5.90e-4	5.55e-4

Table 5: LM Head Optim LR at 120B

N	**500M**	**1B**	**2B**	**3B**	**4B**
η_{opt}^{out}	6.92e-4	425e-4	3.72e-4	3.81e-4	2.86e-4

Table 6: Router Optim LR at 120B

N	**500M**	**1B**	**2B**	**3B**	**4B**
η_{opt}^{router}	7.62e-4	6.55e-4	5.93e-4	5.23e-4	4.89e-4

Table 7: Hidden Optim LR at 120B

N	**500M**	**1B**	**2B**	**3B**	**4B**
η_{opt}^{hidden}	1.05e-3	6.99e-4	5.80e-4	4.79e-4	3.78e-4

Table 8: Embedding Optim LR at 120B

N	**500M**	**1B**	**2B**	**3B**	**4B**
η_{opt}^{out}	1.95e-3	8.38e-4	2.00e-3	1.15e-3	1.05e-3

Table 9: Module-Level Optim LR at 120B

N	**500M**	**1B**	**2B**	**3B**	**4B**
η_{opt}^{out}	6.92e-4	425e-4	3.72e-4	3.81e-4	2.86e-4
η_{opt}^{router}	7.62e-4	6.55e-4	5.93e-4	5.23e-4	4.89e-4
η_{opt}^{hidden}	1.05e-3	6.99e-4	5.80e-4	4.79e-4	3.78e-4
η_{opt}^{out}	1.95e-3	8.38e-4	2.00e-3	1.15e-3	1.05e-3

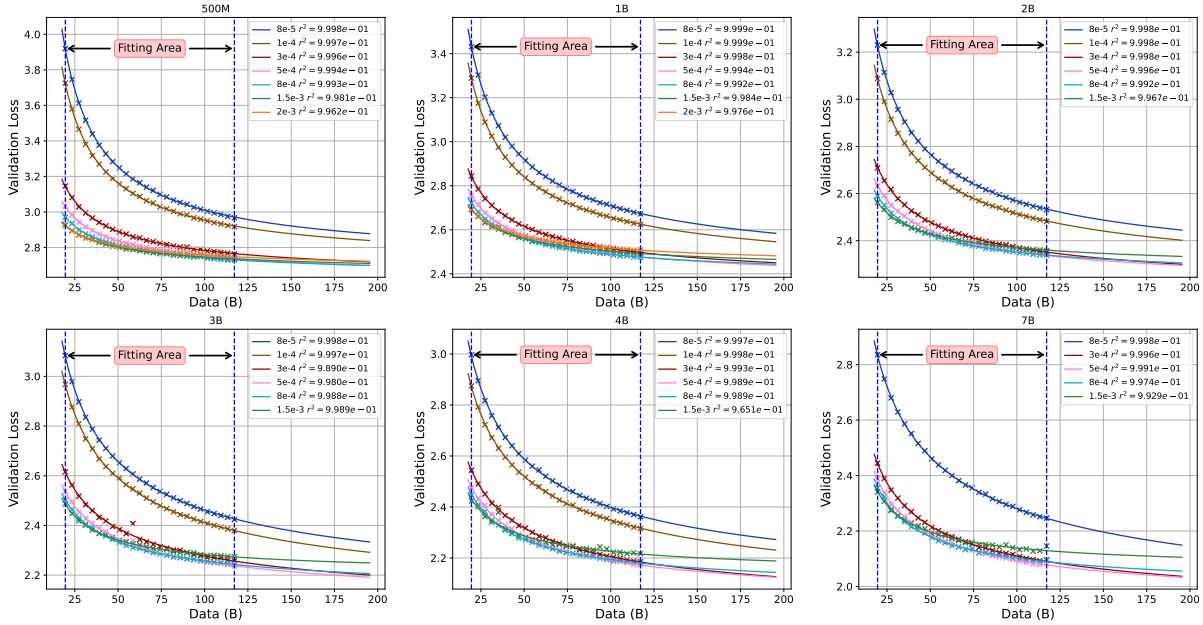


Figure 14: Results of fitting via $L(D) = L_0 + A \cdot D^{-\gamma}$ for each group of experiments.

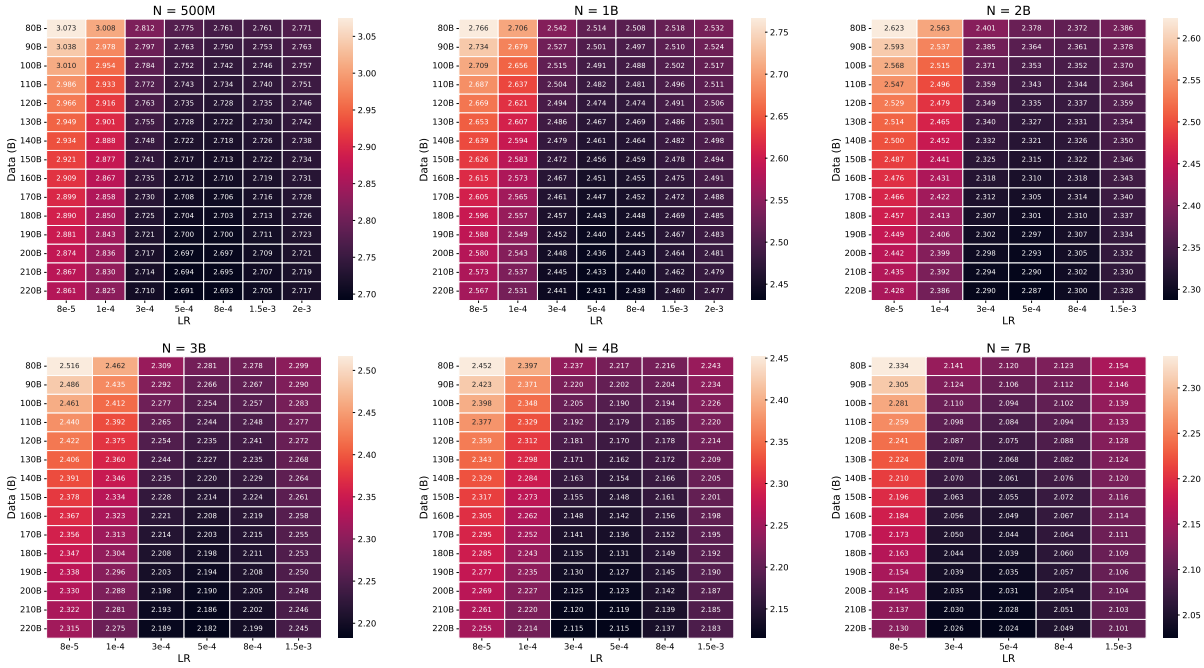


Figure 15: Relationship among loss, learning rate, and training data size of various model.

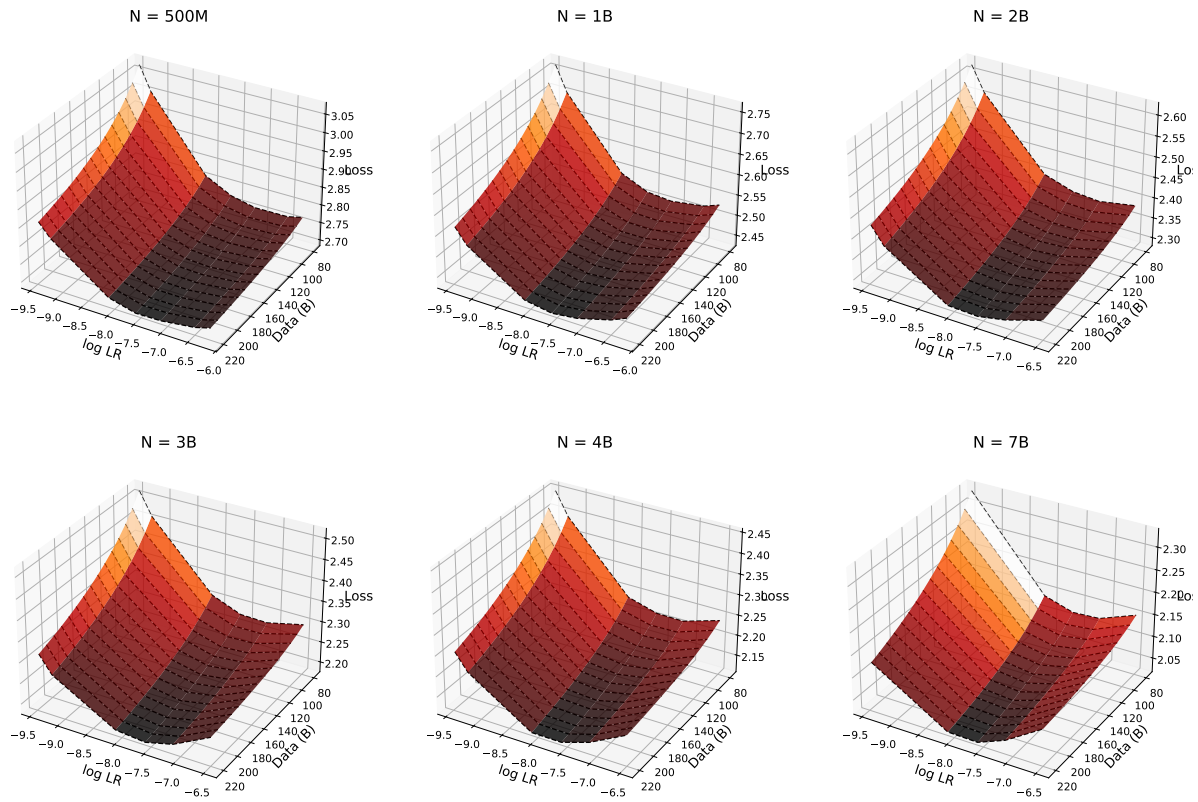


Figure 16: 3D visualization of 15

.

Table 10: Hyperparameters' transfer rule of μTransfer

	Parameterisation:		μ**P**	**Complete**$^{(d)}$**P**	
Multipliers	MHA Residual		$\mathbf{x} + \text{MHABlock}(\mathbf{x})$	$\mathbf{x} + m_L^{-\alpha}\text{MHABlock}(\mathbf{x})$	
	MLP Residual		$\mathbf{x} + \text{MLPBlock}(\mathbf{x})$	$\mathbf{x} + m_L^{-\alpha}\text{MLPBlock}(\mathbf{x})$	
	Unemb. Fwd		Unaugmented	Unaugmented	
Init Variances	Input Emb.	σ_b^2	$\times m_N^{-1}$	$\times m_N^{-1}$	
	Hidden weights				
	Hidden biases/norms				
	Unemb. LN				
	Unemb. Weights		$\times m_N^{-2}$	$\times m_N^{-2}$	
Learning Rates	Input Emb.	η_b	$\times m_N^{-1}$	$\times m_N^{-1} \times m_L^{\alpha-1}$	$\times\sqrt{\frac{1}{m_D}}$
	Hidden weights			$\times m_L^{\alpha-1}$	
	Hidden biases/norm				
	Unemb. LN				
	Unemb. weights		$\times m_N^{-1}$	$\times m_N^{-1}$	
AdamW ϵ	Hidden weights/biases/norms	ϵ_b	$\times m_N^{-1}$	$\times m_N^{-1} \times m_L^{-\alpha}$	$\times\sqrt{m_D}$
	QK norms		NA	$\times m_L^{-\alpha}$	
	Input Emb.		$\times m_N^{-1}$	$\times m_N^{-1}$	
	Output weights/biases/norms				
Weight decay	Hidden weights	λ_b	$\times m_N$	$\times m_N$	$\times\sqrt{\frac{1}{m_D}}$
	Unemb. weights		$\times m_N$	$\times m_N$	
	Rest		$\times 1$	$\times 1$	

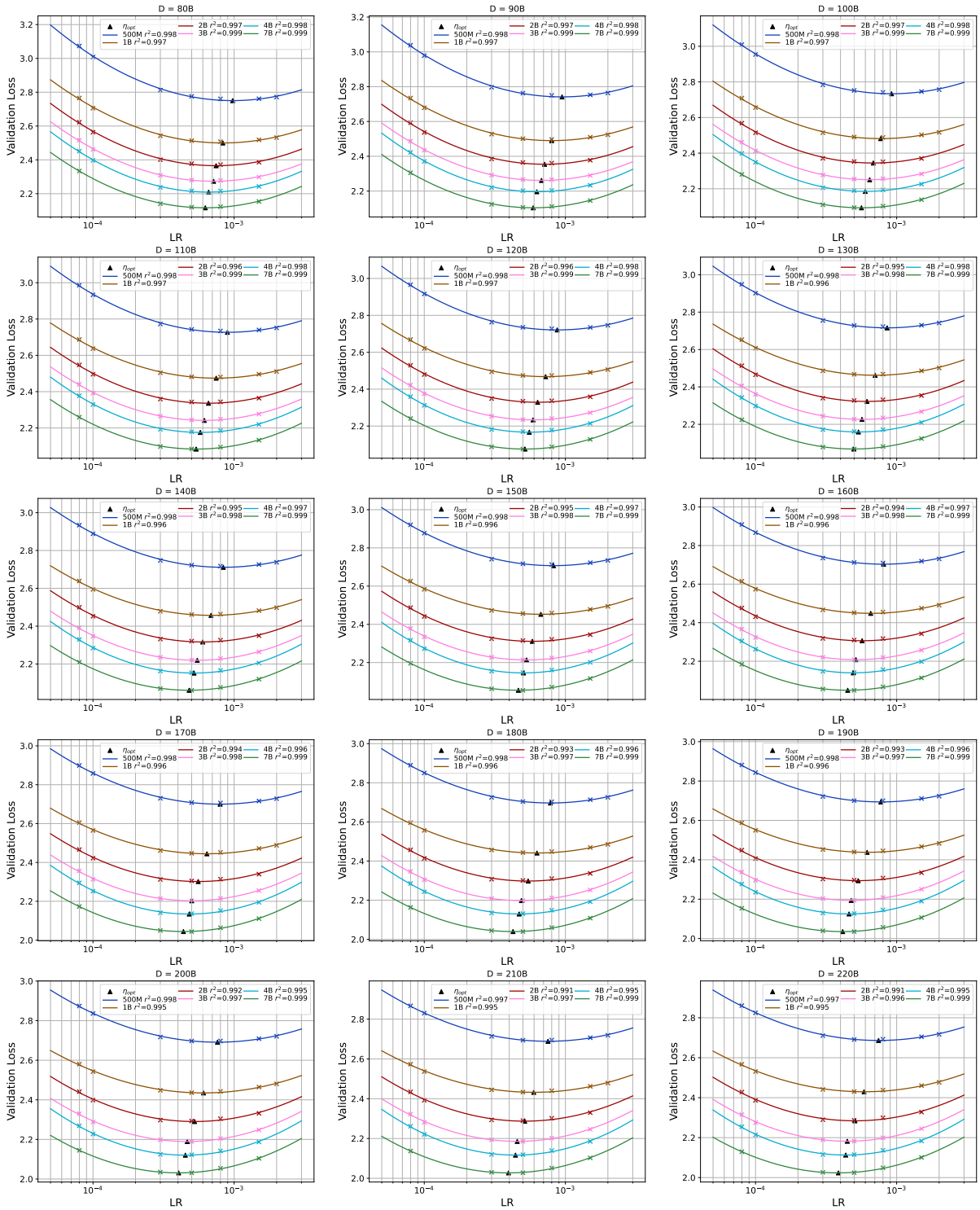
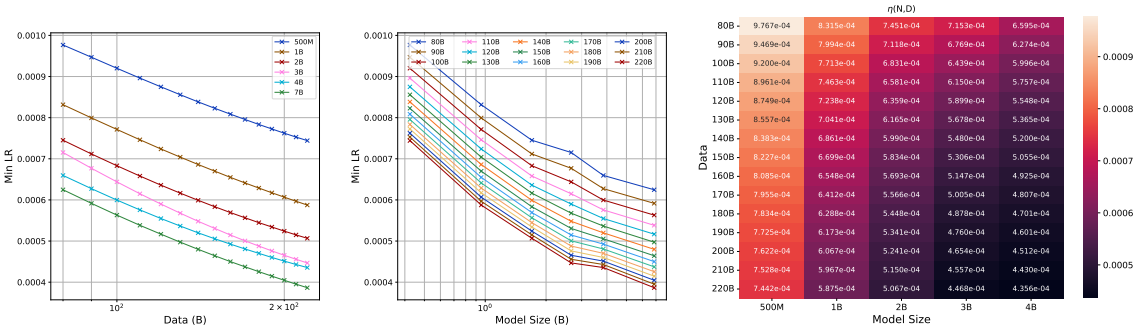


Figure 17: All results of fitting with Equation 2 across different amount of data.



(a) η_{opt}(D) (b) η_{opt}(N) (c) η_{opt}(N, D)

Figure 18: Relationship ammog learning rate η, model size N and training data size D.

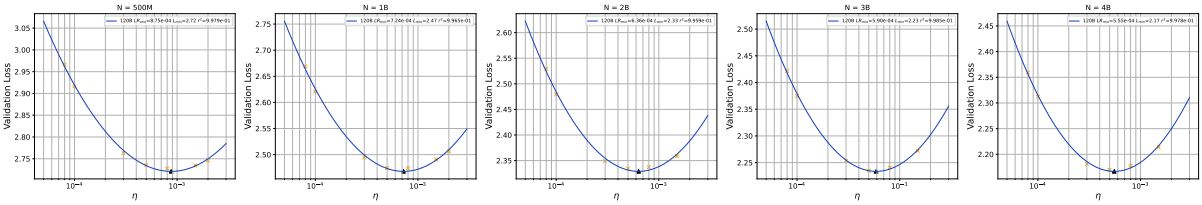


Figure 19: Initial stage of module-level learning rate searching.

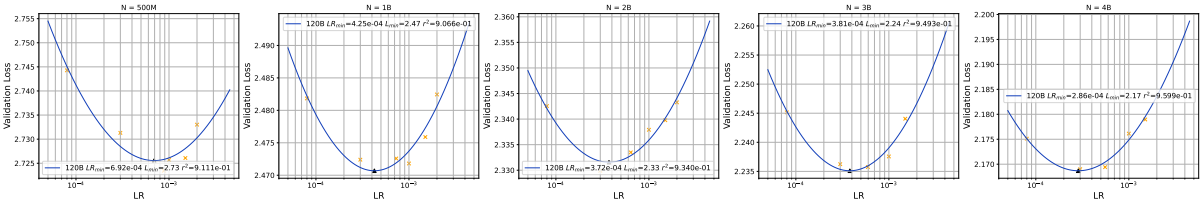


Figure 20: LM Head stage of module-level learning rate searching.

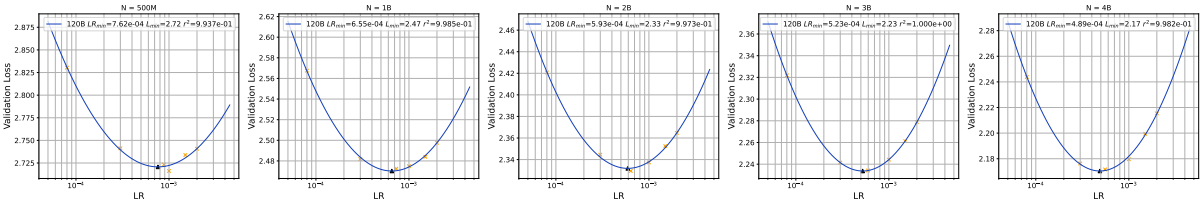


Figure 21: Router stage of module-level learning rate searching.

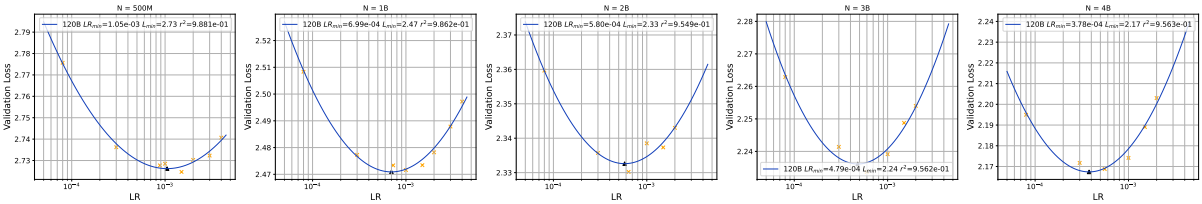


Figure 22: Hidden stage of module-level learning rate searching.

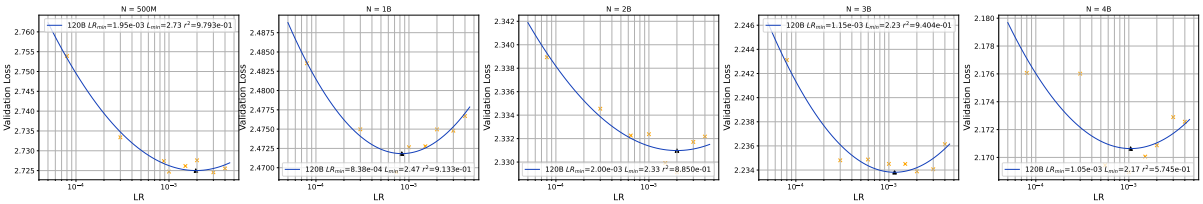


Figure 23: Embedding stage of module-level learning rate searching.