AdaLRS: Loss-Guided Adaptive Learning Rate Search for Efficient Foundation Model Pretraining

Hongyuan Dong, Dingkang Yang, Xiao Liang, Chao Feng†, Jiao Ran
ByteDance Inc.
d_ousia@icloud.com, yangdingkang@bytedance.com, liangxiao.ilx@bytedance.com
chaofeng.zz@bytedance.com, ranjiao@bytedance.com

Abstract

Learning rate is widely regarded as crucial for effective foundation model pretraining. Recent research explores and demonstrates the transferability of learning rate configurations across varying model and dataset sizes, etc. Nevertheless, these approaches are constrained to specific training scenarios and typically necessitate extensive hyperparameter tuning on proxy models. In this work, we propose **AdaLRS**, a plug-in-and-play adaptive learning rate search algorithm that conducts online optimal learning rate search via optimizing loss descent velocities. We provide theoretical and experimental analyzes to show that foundation model pretraining loss and its descent velocity are both convex and share the same optimal learning rate. Relying solely on training loss dynamics, AdaLRS involves few extra computations to guide the search process, and its convergence is guaranteed via theoretical analysis. Experiments on both LLM and VLM pretraining show that AdaLRS adjusts suboptimal learning rates to the neighborhood of optimum with marked efficiency and effectiveness, with model performance improved accordingly. We also show the robust generalizability of AdaLRS across varying training scenarios, such as different model sizes, training paradigms, base learning rate scheduler choices, and hyperparameter settings.

1 Introduction

Learning rate (LR) is regarded as a critical hyperparameter for foundation model pretraining, *e.g.* Large Language Model (LLM), Vision Language Model (VLM), etc. With the development of foundation model pretraining techniques [1, 29, 46, 50, 12, 11, 16, 25], a series of studies seek to find the optimal learning rate for effective pretraining [39, 49, 26, 35]. Although reducing the costs of hyperparameter searches to some extent, these methods are bound to specific pretraining scenarios or necessitate resource-consuming hyperparameter searches on proxy models to take effect.

One line of work predicts the optimal learning rate settings directly via summarizing model performance dynamics *w.r.t.* hyperparameter settings. Under certain model structure designs, the optimal learning rate can be expressed as a function of a series of training hyperparameters [24, 5, 26, 35], such as model size, batch size, training budget, etc. Nevertheless, the resulting correlation laws lack desired generalizability to new training scenarios, where large-scale learning rate search experiments are required to establish the model performance dynamic patterns.

Another line of research explores transferring hyperparameter search results obtained from proxy models to larger ones. Tensor Program series work [48, 49] proves the zero-shot transferability of hyperparameter settings across model sizes for transformer models. As long as the model design

†Email corresponding

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

 https://doi.org/10.52202/085713-3088

meets certain criteria, learning rate search on certain proxy models can be transferred to larger models directly, reducing the hyperparameter search cost significantly. However, the learning search process on proxy models is still time- and resource-consuming. Although auto hyperparameter learning techniques can be applied to accelerate the search process [4, 41, 28], a large number of independent runs are still required to approximate the optimum, resulting in additional training resource overhead.

In this work, we propose AdaLRS, a plug-in-and-play learning rate search algorithm that optimizes the loss descent velocity to find the optimal learning rate in a single run. We formulate the optimization of training loss and loss descent velocity w.r.t LR in foundation model pretraining tasks as convex problems, and show that they share the same optimum by experiment results. Based on these observations, AdaLRS adjusts the base learning rate sequentially according to training loss dynamics to approximate the optimum. We provide theoretical proof for the convergence and geometric error decay of AdaLRS, and show that AdaLRS effectively tackles various foundation model pretraining tasks. It adjusts inappropriate LRs effectively in single runs, and improves model performance significantly for different model sizes, training paradigms, and base scheduler choices.

To summarize, our contributions can be listed as follows:

- We provide theoretical and experimental analysis to demonstrate that the foundation model pretraining loss and its slop *w.r.t.* LR are convex and share the same optimum.
- We propose AdaLRS, an learning rate search algorithm which performs online LR adjustment to optimize loss descent velocity, approximating the optimal learning rate in a single run. We provide theoretical analysis for its convergence and the geometric error decay in the search process.
- We conduct experiments in both LLM and VLM pretraining tasks with varying base learning rates. AdaLRS not only demonstrates marked effectiveness in finding the optimal LR and improving model performance, but also exhibits desired generalizability in different pretraining tasks.

2 AdaLRS

In this section, we provide formulations for the proposed AdaLRS algorithm and present proof of its convergence and geometric error decay in the optimal LR search process.

2.1 Formulations of AdaLRS Algorithm

Previous work has discussed the convexity of training loss optimization *w.r.t.* learning rate settings [38]. In this work, we further hypothesize that the optimization of the training loss and loss curve slope in foundation model pretraining tasks are both near-convex and share the same optimal learning rate, which is supported by the experiment results shown in Section 2.2. In this way, we can approximate the optimal learning rate by optimizing the velocity of loss descent within a single run.

Formulation. Denoting the learning rate at training step t as η_t, and the LR upscaling and downscaling factors as α' and β', we formulate the workflow of AdaLRS as follows:

$$\eta_{t+k} = \begin{cases} \alpha'\eta_t & \text{if } v(\alpha'\eta_t) > v(\eta_t) + 2e \quad \text{(loss slope } \textit{increases} \uparrow\text{)}, \\ \beta'\eta_t & \text{if } v(\alpha'\eta_t) < v(\eta_t) - 2e \quad \text{(loss slope } \textit{decreases} \downarrow\text{)}, \\ \eta_t & \text{otherwise.} \end{cases} \tag{1}$$

$v(\cdot)$ indicates the estimated loss curve slope obtained from a k-step window, while e stands for the estimation error between v and the true loss descent velocity V. α' and β' are rectified LR scaling factors which satisfy $\alpha' = \max(\lambda^t\alpha, 1), \beta' = \frac{1}{\max(\lambda^t\beta, 1)}$, where $\alpha, \beta > 1$ are two multiplicatively independent real numbers and $\lambda \in (0, 1)$ is a decay factor. We validate the multiplicatively independent design of LR scaling factors (for all integers m, n, $\alpha^m = \beta^n \implies m = n = 0$) in Appendix B.

Workflow. During model training, AdaLRS monitors the loss curve slope v_t with the least squares method [7], and attempts to upscale the learning rate when the loss curve slope decays. After the upscaling adjustment, we compare the loss curve slope with that before upscaling. As shown in Equation 1, if the estimated loss slope increases more than $2e$ after the adjustment, the upscaling is regarded as valid and the adjustment will be retained. On the other hand, once the validation fails

under condition $V(\beta'\eta_t) < V(\eta_t) - 2e$, the upscaling adjustment is reverted, with a downscaling factor applied to the base learning rate. As a result, AdaLRS is able to conduct an online search for the optimal learning rate within a single run, performing marked efficiency and effectiveness.

We also introduce several tricks to improve the stability of the AdaLRS algorithm. In the LR upscaling process, we adopt an early stopping operation if the training loss rises over the largest loss value in the loss records, followed by several trial training steps performed after upscaling. This is to ensure that the loss slopes before and after the trial adjustment can be compared fairly at similar loss levels. Additionally, the learning rate search process is restricted to certain step ratios in the early training stage, ensuring a stable LR decay process for pretraining. To tackle extremely high learning rate settings, we also introduce a boundary condition that lowers the LR if the loss increases for two consecutive windows. We refer to Appendix A for a detailed formulation of AdaLRS in pseudocode.

2.2 Convergence Analysis

2.2.1 Theoretical Hypotheses

We make two hypotheses to prove the convergence of the proposed AdaLRS algorithm.

Hypothesis 1. *The optimization of the training loss w.r.t. learning rate in foundation model pretraining is convex, and share the same optimum with loss descent velocity optimization. Let $V(\eta)$ and $L(\eta)$ be the loss descent velocity and loss value function w.r.t. the learning rate η, this hypothesis means that there exists an optimal learning rate η^*, so that:*

$$\begin{cases} \frac{\partial V}{\partial \eta} > 0 \text{ and } \frac{\partial L}{\partial \eta} < 0, & \forall \eta < \eta^*, \\ \frac{\partial V}{\partial \eta} < 0 \text{ and } \frac{\partial L}{\partial \eta} > 0, & \forall \eta > \eta^*. \end{cases} \tag{2}$$

This is a relatively strong assumption, and therefore we conduct both theoretical and experimental analyses to support it. Consider a simplified foundation model pretraining task with a constant learning rate and SGD optimizer, we formulate the update rule for model parameter ψ as:

$$\psi_{t+1} = \psi_t - \eta \nabla L_t, \tag{3}$$

where L_t is the training loss at time step t. The expected loss descent velocity per step (using Taylor expansion) is:

$$\mathbb{E}\left[L_{t+1} - L_t\right] \approx -\eta \|\nabla L_t\|^2 + \frac{C_{Lip}}{2}\eta^2 \|\nabla L_t\|^2, \tag{4}$$

where C_{Lip} is the Lipschitz constant of the gradient. When the learning rate is small, the first term dominates the expectation, and a smaller η leads to a smaller decrease in loss. When the learning rate is large, on the other hand, the second term cannot be neglected. It contributes to suppress the expected loss descent value. Differentiating w.r.t. η and setting to zero for the extreme:

$$\frac{\partial}{\partial \eta}\left(-\eta + \frac{C_{Lip}}{2}\eta^2\right) = 0 \implies \eta^* = \frac{1}{C_{Lip}}. \tag{5}$$

This shows that the loss descent velocity w.r.t. η is a convex function. It is intuitive to infer that a higher loss descent velocity leads to a lower training loss, and vice versa, which proves that the optimization of training loss is also a convex problem.

Apart from theoretical analyses, we also conduct small-scale pretraining experiments in LLM and VLM pretraining to verify this hypothesis. For LLM pretraining, we use Qwen2.5-1.5B and Qwen2.5-7B [43] models with randomly initialized parameters, and train them from scratch on 8M samples from the SlimPajama [42] dataset. A series of exponentially larger learning rates in $[2e^{-6}, 8e^{-3}]$ are applied to investigate the dynamics of training loss and loss curve slope under varying LR settings. For VLM pretraining, we conduct experiments on a 2B model with the architecture described in SAIL-VL [16]. Approximately 8M data consisting of caption and OCR samples are used for model training, with learning rates configured in $[2e^{-5}, 2]$.

The resulted training loss curves are shown in Figures 1(a)(b)(c), with training loss and loss descent velocity dynamics *w.r.t.* varying LR settings illustrated in Figures 1(d)(e)(f) and (g)(h)(i), respectively. As Figures 1(d)(e)(f) shows, pretraining losses at different training steps exhibit strong correlations with learning rate settings, forming a series of convex curves with constant optima. To investigate

https://doi.org/10.52202/085713-3088

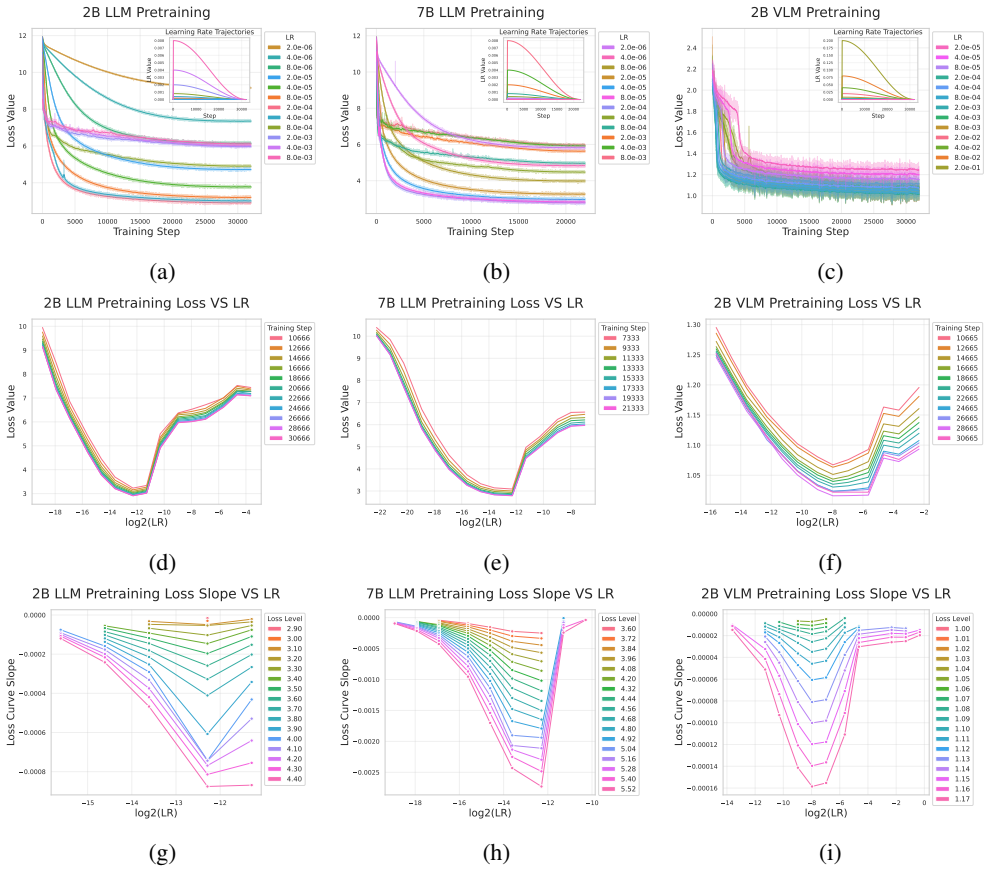


Figure 1: Training loss and loss descent velocity dynamics *w.r.t.* varying LR settings for LLM and VLM pretraining. Figures (a)&(b)&(c) show the training losses and LR trajectories with a cosine LRS, while Figures (d)&(e)&(f) illustrate how training loss varies across different LR settings through the training process. Figures (g)&(h)&(i) are loss slope dynamics at varying loss levels, obtained from experiments with constant learning rates.

how learning rate influences the loss descent velocity, we conduct several sets of experiments with constant LR values. We show the correlation between loss curve slopes and learning rate settings in Figures 1(g)(h)(i), and the corresponding curves are not only convex, but also sharing the same optima as the training loss curves. Some data points in Figures 1(g)(h)(i) are not displayed because the corresponding inappropriate learning rate cannot optimize the model to such loss levels. These experiment results strongly support the validity of Hypothesis 1.

In practice, V is also influenced by the current loss value and optimization states. As described in Section 2.1, AdaLRS performs trial training steps before comparing loss slopes to eliminate the influence of loss levels, and we assume that the difference in optimization states is limited between consecutive training steps. As a result, we present V as a univariate function in our proof.

Hypothesis 2. *Let v_t denote the empirical estimate of $V(\eta_t)$, computed over a sliding window of k training steps. Denoting the estimation error bound as e, i.e. $\forall t > 0$, $|v_t - V(\eta_t)| < e$, there exists a sliding window length k such that e can be sufficiently small.*

This guarantees that AdaLRS converges to a relatively narrow neighborhood of the optimum η^*.

2.2.2 Convergence

Equipped with the above hypotheses, we seek to prove the following theorem:

Theorem 2.1. *The learning rate sequence $\{\eta_t\}$ generated by the algorithm converges almost surely to the e-neighborhood of the optimal learning rate $N_e(\eta^*) \triangleq \{\eta : |\eta - \eta^*| < e\}$, i.e.,*

$$\lim_{t\to\infty} \mathbb{P}(|\eta_t - \eta^*| < e) = 1. \tag{6}$$

Proposition 2.2. *If $\eta_t, \max(\lambda^t\alpha, 1)\eta_t < \eta^* - e$, $\exists T < \infty$ such that η_{t+T} moves towards η^*, and vice versa.*

Since we gradually push the the loss slope decay threshold θ to 1 (detailed in Appedix A), $v_{t+k} < v_t * \theta$ will almost surely trigger the LR upscaling adjustment in finite steps. We denote the time step as T, with the loss descent velocity before or after trial LR adjustment as v_T and v_{T+k}, respectively. The LR upscaling adjustment is kept and only kept when $v_{T+k} - v_T > 2e$, which can be reformulated as $V(\eta_{T+k}) + e_{T+k} - V(\eta_T) - e_T > 2e$, where e_{T+k} and e_T stand for loss slope estimation error at corresponding steps. Substituting gives $V(\eta_{T+k}) - V(\eta_T) > 2e - e_{T+k} + e_T > 2e - |e_{T+k}| - |e_T| > 0$, which means the upscaling condition in Equation 1 ensures an increase in the true loss descent velocity. According to Hypothesis 1, the upscaling adjustment should be retained under this condition. For η_t, $\frac{\eta_t}{\max(\lambda^t\beta,1)} > \eta^* + e$ cases, we can derive that trial LR upscaling adjustment gives lower loss descent velocity. A downscaling adjustment should and will occur under such circumstance.

Proposition 2.3. *$\exists T' < \infty$, for $\forall t > T'$, the gap between η_t and η^* is bounded by the decaying LR scaling factors.*

$$\forall \eta_t, \frac{\eta^* - e}{\alpha'}\beta' < \eta_t < \frac{\eta^* + e}{\beta'}\alpha' \tag{7}$$

where $\beta' = \frac{1}{\max(\lambda^t\beta,1)}$ and $\alpha' = \max(\lambda^t\alpha, 1)$ are decaying LR adjustment factors.

In practice, the value of λ is set relatively large, ensuring that the learning rate η converges to the neighborhood of η^* at the current LR adjustment magnitude before the scaling factor becomes too small. Since the desired LR adjustment is guaranteed by Proposition 2.2, we begin with $\frac{\eta^*-e}{\alpha'} < \eta_t < \frac{\eta^*+e}{\beta'}$. In this way, the learning rate η_t either oscillates within this range, or attempts to search the learning rate in $[\frac{\eta^*+e}{\beta'}, \frac{\eta^*+e}{\beta'}\alpha']$ and $[\frac{\eta^*-e}{\alpha'}\beta', \frac{\eta^*-e}{\alpha'}]$ respectively for higher or lower learning rate values. Since we have $\frac{\eta^*+e}{\beta'} > \eta^* + e$ always holds true, the LR upscaling attempts will trigger LR downscaling according to Proposition 2.2, maintaining the LR value given in Proposition 2.3. Similarly, learning rate downscaling attempts will be rejected by Proposition 2.2.

Proposition 2.3 indicates that η_t is bounded to the neighborhood of η^* by decaying LR adjustment factors. As the LR adjustment factor bound narrows, *e.g.* $\lim_{t\to\infty} \frac{\alpha'}{\beta'} = 1$ and $\lim_{t\to\infty} \frac{\beta'}{\alpha'} = 1$, η_t will fall into the e-neighborhood of η^* eventually. Therefore, we have Theorem 2.1 proved.

2.3 Complexity Analysis

We introduce the following theorem to demonstrate AdaLRS's geometric error decay.

Theorem 2.4. *There exists $\gamma \in (0, 1)$ such that for $\eta_t \notin N_e(\eta^*)$, $|\eta_{t+k} - \eta^*| \leq \gamma|\eta_t - \eta^*|$ when LR adjustment is triggered.*

For $\eta_t \notin N_e(\eta^*)$, let $\gamma_a \in (\frac{\eta^*-\alpha'\eta_0}{\eta^*-\eta_0}, 1)$ for the learning rate upscaling process. Since η moves towards η^* in the upscaling process, the following inequation holds for the whole upscaling process:

$$\forall t > 0, \frac{\eta^* - \alpha'\eta_t}{\eta^* - \eta_t} < \frac{\eta^* - \alpha'\eta_0}{\eta^* - \eta_0}, |\eta_{t+k} - \eta^*| < \gamma|\eta_t - \eta^*|. \tag{8}$$

Similarly, γ_b can be selected from $(\frac{\beta'\eta_0-\eta^*}{\eta_0-\eta^*}, 1)$ for the LR downscaling process. Selecting $\gamma = max(\gamma_a, \gamma_b)$, Theorem 2.4 is proved directly.

Theorem 2.4 implies the geometric error decay of the AdaLRS algorithm bounded by γ. Let R be the range of the learning rate search space. AdaLRS is able to approximate the optimal learning rate within $\mathcal{O}(\log R)$ adjustments. As a result, for foundation model pretraining tasks with data size larger than $\mathcal{O}(\log R)$, AdaLRS reaches the optimum neighborhood within a single run. The effectiveness of AdaLRS surpasses traditional auto hyperparameter search algorithms significantly, which often require extensive independent experiments to take effect.

https://doi.org/10.52202/085713-3088

Table 1: Detailed hyperparameters for the main experiments. "Fit", "Large", and "Small" refer appropriate, too large, and too small learning rates, respectively. "BSZ" stands for batch size.

Hyperparameter	*2B LLM*			*7B LLM*			*2B VLM*		
	Fit	Large	Small	Fit	Large	Small	Fit	Large	Small
Learning Rate	$2e^{-4}$	$2e^{-3}$	$2e^{-5}$	$2e^{-4}$	$2e^{-3}$	$2e^{-5}$	$8e^{-3}$	$4e^{-1}$	$2e^{-4}$
BSZ / Micro BSZ	$1024/512$			$2048/512$			$2048/1024$		
Window Size k	2500			2000			1000		
Data Composition	Detail Caption & OCR			SlimPajama Train Set			SlimPajama Train Set		
Search Step Ratio	$[0.1, 0.4]$			$[0.1, 0.35]$			$[0.1, 0.35]$		

3 Experiments

3.1 Experiment Setup

Model Training. We conduct experiments on both LLM and VLM pretraining. Qwen2.5-1.5B and Qwen2.5-7B [43] are adopted for LLM pretraining, with parameters randomly initialized via He initialization [21], Glorot initialization [18], and etc. We use a total of approximately 64M samples from the SlimPajama [42] dataset for LLM training from scratch, with all samples shuffled randomly. For VLM pretraining, on the other hand, we adopt the model structure of SAIL-VL [16], with InternViT-300M [13] and Qwen2.5-1.5B adopted as backbone models. We use a collection of detail caption and image OCR data to train the vision-to-language projector from scratch. Detail caption datasets are curated via a similar recaption procedure as described in SAIL-VL [16], while OCR data is collected from a series of opensource datasets [6, 45, 20].

To demonstrate the effectiveness of the AdaLRS algorithm, we set different learning rates for each training task, *i.e.*, learning rates too small, too large, and appropriate for pretraining. For each setting, we train a baseline model and a model optimized by AdaLRS for fair comparison. We set the upscaling factor α, downscaling factor β, and decaying factor λ as 3, 2, and 0.99 in all experiments for LR adjustment effectiveness. Approximately 120B and 160B tokens are used for LLM and VLM pretraining, with roughly 10,000 and 20,000 910B NPU hours consumed for 2B and 7B model pretraining experiments. A cosine learning rate scheduler [32] is applied in the main experiments. Detailed model training recipes are elaborated in Table 1.

Evaluation. To demonstrate the effectiveness of the proposed AdaLRS algorithm, we show the learning rate search and training loss dynamics of our foundation model training experiments. For LLM pretraining, we quantify the performance advantage of models trained with AdaLRS with final training loss and perplexity (PPL) computed on SlimPajama [42] train, validation, and test splits. We also conduct a lightweight SFT on 6M samples from the Infinity-Instruct dataset [27], and evaluate the downstream model performance on open-ended generation tasks such as Alpaca-Gen and KNIGHT-Gen [47]. For VLM experiments, however, there is no widely accepted benchmark or metric for pretrained VLM evaluation. Therefore, we tune the pretrained VLM on 3M samples from Infinity-MM [19] stage3 data with all parameters unfrozen, and use a series of publicly available VLM benchmarks for model evaluation, such as LLaVABench [30], MMVet [51], MMStar [10], DocVQA [34], OCRBench [31], TextVQA [40], DetailCaps-4870 [17], etc.

3.2 Main Results

AdaLRS Approximates the Appropriate Learning Rate Effectively. As shown in Figure 2, the proposed AdaLRS algorithm (*i.e.*, blue lines) pushes inappropriate learning rates (*i.e.*, gray lines) towards the optimal ones (*i.e.*, red dashed lines) effectively. For models trained with appropriate initial LR settings (Figures 2(a)(d)(g)), AdaLRS conducts trial attempts to adjust the learning rates and eventually converges to the vicinity of the optimum. For excessively large or small LR settings (Figures 2(b)(c)(e)(f)(h)(i)), the adjusted learning rates produced by AdaLRS demonstrate marked advantages in consistency with the optimum compared with baseline experiments.

Although exhibiting compatibility with varying LR settings, several undesired LR scaling operations can be observed in Figures 2(c)(d)(f). We attribute them to the aggressive upscaling and downscaling factor settings and the limited training steps for LR adjustment, which can be mitigated by milder LR scaling factors or longer LR search processes.

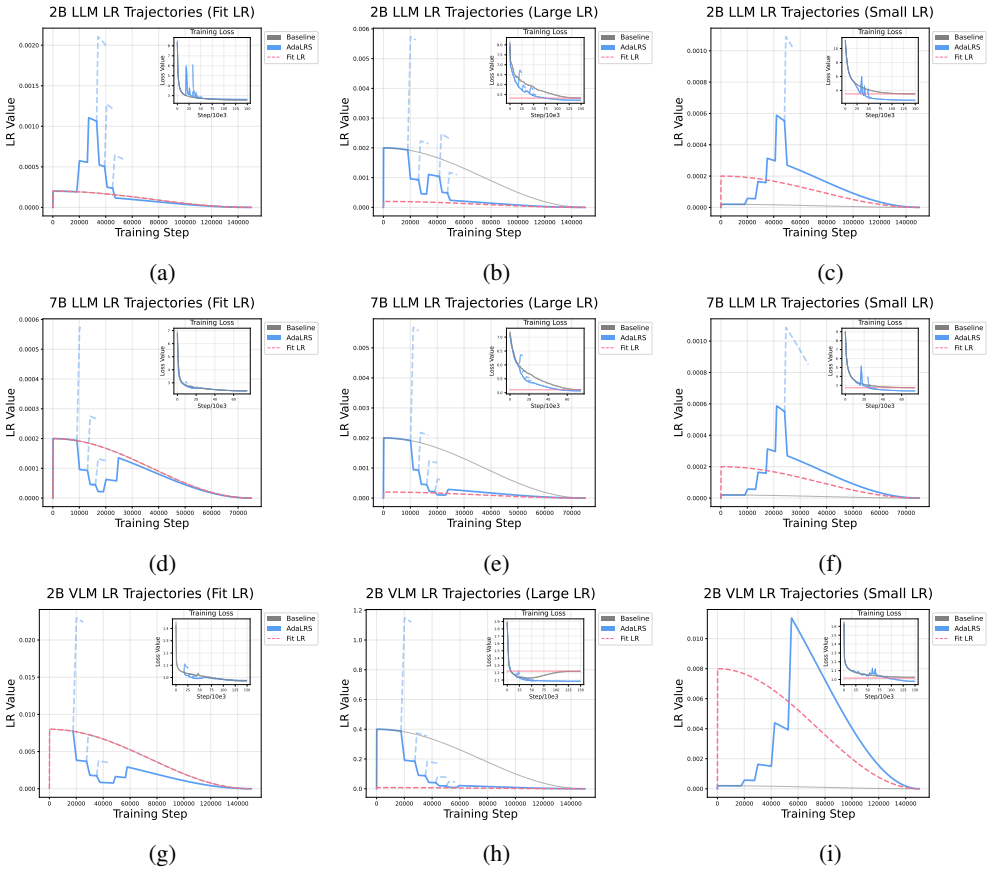


Figure 2: AdaLRS's Learning rate adjustment process in foundation model pretraining under differnt LR settings. "Fit LR" refers to learning rate appropriate for the pretraining task estimated by pilot study results. Dashed curves represent failed LR upscaling attempts.

Table 2: Evaluation results of our LLM pretraining experiments. We use "/" to separate performance scores of models trained with AdaLRS and baselines. Black and gray numbers indicate better or worse performance.

Benchmark	*2B LLM* Fit LR	*2B LLM* Large LR	*2B LLM* Small LR	*7B LLM* Fit LR	*7B LLM* Large LR	*7B LLM* Small LR
Training Loss	2.62/2.54	5.21/5.32	2.56/3.50	2.38/2.39	5.07/5.11	2.38/2.74
Training PPL	13.72/12.65	183.28/205.00	12.88/33.21	10.84/10.88	158.54/165.67	10.76/15.49
Validation PPL	13.51/12.42	183.94/204.97	12.66/32.69	10.67/10.72	158.22/165.70	10.61/15.30
Test PPL	13.51/12.43	183.70/204.68	12.66/32.66	10.69/10.74	158.16/165.61	10.63/15.32
Alpaca-Gen	17.29/18.56	7.11/6.09	17.10/15.40	21.76/21.61	6.35/5.55	21.15/20.68
KNIGHT-Gen	10.13/11.29	3.96/2.02	10.81/8.36	13.62/13.35	4.08/3.82	13.53/12.29

AdaLRS Accelerates Pretraining Convergence Significantly. As shown in the top-right corner of each subfigure in Figure 2, models trained with AdaLRS demonstrate significant advantages in training losses. For learning rates too small or too large, models trained with vanilla cosine LRS suffer from slow loss descent velocities, with severe training instability observed under certain circumstances (Figure 2(h)). AdaLRS not only improves training loss convergence effectively, but also aids such training instability problems. In experiments with appropriate learning rate settings, AdaLRS introduces slight loss spikes in model training, but the resulted training loss remains close to the baseline, exhibiting desired stability.

We also illustrate the acceleration ratio of our method for inappropriate LR settings. As shown in the training loss curves of Figures 2(b)(c)(e)(f)(h)(i), AdaLRS experiments reach the final training loss of the baselines at early training stages. For excessively small learning rates, AdaLRS surpasses the baselines at approximately 50% training steps, while more than 30% training costs can be reduced for

https://doi.org/10.52202/085713-3088

Table 3: Evaluation results for the our 2B VLM experiments. DetailCaps-4870 is used to evaluate the basic visual understanding ability of the pretrained model, while other benchmarks scores are evaluated with the SFT model.

LR Setting	LLaVABench	MMVet	MMStar	DocVQA	OCRBench	TextVQA	DetailCaps-4870	Average
Fit LR	39.5/38.5	34.58/32.02	48.67/49.53	77.99/78.00	718/735	64.89/63.74	55.68/55.30	56.16/55.80
Large LR	36.8/35.7	31.47/30.50	44.47/44.33	57.53/57.42	631/606	60.32/58.08	49.02/47.08	48.96/47.67
Small LR	44.3/39.2	36.15/30.23	48.67/49.20	77.75/77.47	730/689	64.85/57.86	56.65/53.51	57.34/53.77

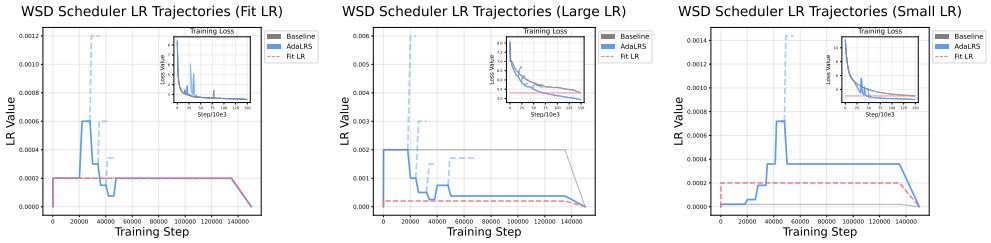


Figure 3: AdaLRS's Learning rate adjustment process in 2B LLM pretraining with WSD scheduler. We refer to Figure 2 for denotation definitions.

large LR settings. Although the LR upscaling attempts may introduce extra training costs, significant training loss advantages are still witnessed in models trained with AdaLRS under exactly the same training budgets.

It is also worth noticing that although AdaLRS adjusts learning rates reasonably for large LR settings, the training loss remains relatively high throughout the training process. We attribute this problem to the disruption of model initialization parameters due to the overly large learning rates. However, AdaLRS still reaches the vicinity of the optimal learning rate in a single run under such circumstances, which still improves the efficiency of the learning rate search process.

AdaLRS Enhances Pretrained Foundation Model Performance Markedly. To demonstrate the model performance advantage of training with AdaLRS, we provide PPL and downstream task evaluation results for LLM experiments in Table 2. AdaLRS not only surpasses baseline LLMs pretrained with suboptimal learning rates, but also achieves comparable model performance with Fit LR baselines. It is worth noticing that AdaLRS improves model performance significantly for excessively small LRs, even surpassing the Fit LR baselines in 7B LLM pretraining. This observation implies that starting with a relatively small LR setting, AdaLRS can potentially achieve both the optimal learning rate and model performance for unexplored pretraining tasks within a single run.

For VLM pretraining, we show VQA benchmark scores of the SFT VLM models in Table 3. Models pretrained with the AdaLRS algorithm demonstrate remarkable performance advantages across most of the benchmarks, which encompass visual understanding tasks in both natural scenes and OCR-related tasks. Similar to experiments on LLM pretraining, we also observe that AdaLRS starting from a small learning rate yields better performance than Fit LR baselines, which validates the effectiveness of the AdaLRS algorithm in unexplored pretraining tasks.

3.3 Extending AdaLRS to WSD Scheduler

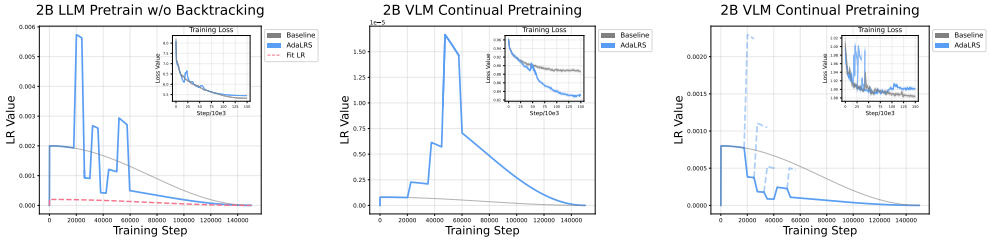
Since AdaLRS simply applies scaling factors on the learning rates produced by the scheduler and conducts LR search only in the early training stages, we can apply it to many mainstream schedulers seamlessly. Considering the generalizability of the conclusion and experiment efficiency, we use WSD [22] scheduler instead of Cosine scheduler in our experiments, to demonstrate the compatibility of AdaLRS with other schedulers. We use a linear decay function for the final 10% training steps in the implementation of the WSD scheduler. Experiment results are shown in Figure 3. Similar to the experiment results on cosine learning rate schedulers shown in Figure 2(a)(b)(c), AdaLRS adjusts unreasonable LR settings effectively, and shows desired stability for appropriate configurations.

3.4 Robustness Across Hyperparameter Settings

In this part, we investigate the model performance dynamics with different AdaLRS hyperparameters. We take 2B LLM pretraining as an example and conduct experiments with small and large learning

Table 4: Model performance of 2B LLMs trained with different AdaLRS hyperparameters. α, β, and λ refer to LR upscaling factor, downscaling factor, and decay factor, respectively. Baselines with AdaLRS disabled are shown with "$-$" as hyperparameters.

	Small LR						**Large LR**				
α/β	–	3/2	2/1.67	1.5/1.43	2/1.67	2/1.67	–	3/2	2/1.67	1.5/1.43	2/1.67
λ	–	0.99	0.99	0.99	0.95	0.9	–	0.99	0.99	0.99	0.9
Final LR	$2.0e^{-5}$	$3.6e^{-4}$	$3.1e^{-4}$	$1.5e^{-4}$	$3.1e^{-4}$	$2.2e^{-4}$	$2.0e^{-3}$	$3.8e^{-4}$	$5.2e^{-4}$	$7.2e^{-4}$	$7.9e^{-4}$
Training Loss	3.07	2.55	2.55	2.58	2.54	2.54	5.30	4.94	5.10	5.15	5.08
Validation PPL	21.59	12.87	12.75	13.11	12.61	12.68	201.54	140.13	163.95	172.01	159.55
Test PPL	21.61	12.89	12.77	13.13	12.63	12.70	201.36	140.07	163.81	171.85	159.47



(a) (b) (c)

Figure 4: Ablation studies for the backtracking downscaling strategy (a) and the training dynamics of AdaLRS on VLM continual pretraining (b)(c).

rate settings, respectively. As shown in Table 4, models trained with AdaLRS demonstrate marked advantages in training loss and evaluation PPL compared to baselines, showing strong robustness across varying hyperparameter combinations. We also observe a clear trend that, as the LR scaling and decay factors decrease, the final learning rate adjusted by AdaLRS tends to decrease for small-LR settings and increase for large-LR settings, respectively. In the experiments with $\alpha, \beta, \lambda = 2, 1.67, 0.99$ and $\alpha, \beta, \lambda = 2, 1.67, 0.95$, both settings yield nearly the same final learning rate, despite their different hyperparameters. This is because LR overshooting in the former triggers a downscaling adjustment, whereas the 0.95 decay factor in the latter prevents such adjustment.

3.5 Backtracking LR Downscaling Strategy Ablation

In this part, we evaluate the effectiveness of our backtracking LR downscaling strategy, which restores model training states before LR downscaling for training stability. We show model training dynamics w/o the backtracking strategy applied on the 2B LLM pretraining task in Figure 4, and the scaling factors are set the same as mentioned in Section 3.1.

As shown in Figure 4(a), the training loss curve moves upwards during LR upscaling attempts, which may introduce instability in model parameter distribution. Although AdaLRS adjusts the learning rate downwards to approximately 38% of the baseline, the training loss remains relatively high in the entire training process. At the end of training, AdaLRS algorithm without backtracking yields higher training loss than the baseline. We refer to Figure 2(b) for training loss dynamics with the backtracking strategy applied. Loss upward movements and underlying disruptive parameter updates are eliminated by restoring model states. Comparing Figure 4(a) and Figure 2(b), the backtracking strategy improves the resulting model training loss by a large margin.

3.6 AdaLRS for Continual Pretraining

In this part, we explore the compatibility of the proposed AdaLRS algorithm with the continual pretraining paradigm. Different from pretraining foundation models from scratch, disruptive parameter updates may result in catastrophic forgetting and harm model performance markedly in continual pretraining [33]. We take VLM continual pretraining as an example, updating both the projector and vision encoder simultaneously to enable more model capacity used for visual understanding. It is a common practice to improve VLM model performance [2, 50, 11]. To be specific, we conduct 2B VLM continual pretraining from the model checkpoint after from-scratch pretraining, *i.e.*, the baseline model in Figure 2(g). We locate the optimal learning rate in this stage at approximately $4e^{-5}$, with $8e^{-7}$ and $8e^{-4}$ used as small and large LR, respectively. The same set of VLM pretraining datasets is used in this stage, and the experiment results are shown in Figure 4(b)(c).

https://doi.org/10.52202/085713-3088

As expected, AdaLRS adjusts learning rates towards the optimum effectively in VLM continual pretraining. For the small learning rate experiment shown in Figure 4(b), AdaLRS improves the training loss by a large margin from 0.8851 to 0.8286. However, although AdaLRS downscales the large learning rate in Figure 4 (c), the final training loss remains higher than 1.88. The LR downscaling adjustments fail to improve model performance and even introduce severer loss oscillation, resulting in higher training loss. These observations show that AdaLRS performs promisingly for continual foundation model training with small initial learning rates, but can not eliminate the catastrophic forgetting problem caused by large learning rates.

4 Related Work

Optimal Learning Rate Prediction in Foundation Model Pretraining. Due to the prohibitively expensive pretraining costs, a large number of research seeks to predict the optimal learning rate for large scale LLM pretraining. This line of work typically focuses on summarizing model performance dynamics as a function *w.r.t.* hyperparameter settings [24, 5, 8, 26], such as model size, batch size, training budget, etc. MM1 [35] further extends this approach to VLM pretraining, discovering the dependency of optimal LR on the number of model parameters. Although reducing hyperparameter search costs for larger scale model training, these methods require hundreds or even thousands of pretraining experiments on smaller models to form an optimal learning rate expression. What's worse, the resulted optimal LR expressions are often restricted by certain model structure or data composition designs to take effect, suffering from poor generalizability.

Transferring Hyperparameter across Model Sizes. Tensor Program series work studies the transferability of hyperparameter settings across varying model sizes. Recently proposed μP [48] and μTransfer [49] prove the transferability of hyperparameter settings (including learning rate) across Multi-Layer Perceptron [37] (MLP) and Transformers [44, 14, 9]. Powered by μTransfer, optimal learning rate search can be performed at proxy models and then transferred to larger ones [15, 22], reducing the search cost by a large margin. However, searching for the optimal learning rate on proxy models can still be resource-consuming, and the optimal learning rate may shift across model sizes in complicated model designs.

Auto Hyperparameter Search. This line of work treats hyperparameter search as an optimization problem. Since learning rate is a non-differentiable variable in model training, researchers propose to search for optimal learning rates via grid/random search [4], Bayesian optimization [41], multi-armed bandit algorithm [28], evolution-based method [23], reinforcement learning [36], etc. Nevertheless, despite the search cost reduction compared with brutal search, these methods still require extensive independent runs to establish the underlying optimal learning rate prediction model.

It is worth noticing that a recent work Hypergradient Descent (HD) [3] also conducts online LR adjustments. However, HD serves as a learning rate scheduler in model training and is only verified to be effective against constant schedulers in small neural networks. AdaLRS differs from HD in that our method is a LR search algorithm which is compatible with modern base LRSs, such as cosine and WSD schedulers, and is examined to be effective for modern foundation model pretraining.

5 Conclusions

In this paper, we propose AdaLRS, an online optimal learning rate search algorithm which optimizes the loss descent velocity to approximate the optimal learning rate. To validate this algorithm, we provide both theoretical and experimental analyzes to show the convexity of foundation model pretraining loss and its slope, as well as their shared optimum in LR. We also provide convergence and complexity analysis of AdaLRS to show its effectiveness and efficiency. Experiments in a series of foundation model pretraining tasks demonstrates that AdaLRS effectively adjusts inappropriate learning rates to the vicinity of the optimum in a single run and accelerates model convergence speed by a large margin. Starting from excessively small learning rates, AdaLRS achieves comparable and even superior performance with baseline models trained with near-optimal learning rates. AdaLRS is validated to improve model performance with varying model sizes, training paradigms, and base schedulers, exhibiting promising potential to be applied in unexplored pretraining tasks. We refer to Appendix C for the discussion of the limitations.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 9

[3] Atilim Gunes Baydin, Robert Cornish, David Martinez Rubio, Mark Schmidt, and Frank Wood. Online learning rate adaptation with hypergradient descent. *arXiv preprint arXiv:1703.04782*, 2017. 10

[4] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *The journal of machine learning research*, 13(1):281–305, 2012. 2, 10

[5] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024. 1, 10

[6] Ali Furkan Biten, Ruben Tito, Lluis Gomez, Ernest Valveny, and Dimosthenis Karatzas. Ocr-idl: Ocr annotations for industry document library dataset. In *European Conference on Computer Vision*, pages 241–252. Springer, 2022. 6

[7] Åke Björck. Least squares methods. *Handbook of numerical analysis*, 1:465–652, 1990. 2, 15

[8] Johan Bjorck, Alon Benhaim, Vishrav Chaudhary, Furu Wei, and Xia Song. Scaling optimal lr across token horizons. *arXiv preprint arXiv:2409.19913*, 2024. 10

[9] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 10

[10] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024. 6

[11] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024. 1, 9

[12] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024. 1

[13] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023. 6

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 10

[15] Nolan Dey, Gurpreet Gosal, Hemant Khachane, William Marshall, Ribhu Pathria, Marvin Tom, Joel Hestness, et al. Cerebras-gpt: Open compute-optimal language models trained on the cerebras wafer-scale cluster. *arXiv preprint arXiv:2304.03208*, 2023. 10

[16] Hongyuan Dong, Zijian Kang, Weijie Yin, Xiao Liang, Chao Feng, and Jiao Ran. Scalable vision language model training via high quality data curation. *arXiv preprint arXiv:2501.05952*, 2025. 1, 3, 6

 https://doi.org/10.52202/085713-3088

[17] Hongyuan Dong, Jiawen Li, Bohong Wu, Jiacong Wang, Yuan Zhang, and Haoyuan Guo. Benchmarking and improving detail image caption. *arXiv preprint arXiv:2405.19092*, 2024. 6

[18] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. 6

[19] Shuhao Gu, Jialing Zhang, Siyuan Zhou, Kevin Yu, Zhaohu Xing, Liangdong Wang, Zhou Cao, Jintao Jia, Zhuoyi Zhang, Yixuan Wang, et al. Infinity-mm: Scaling multimodal performance with large-scale and high-quality instruction data. *arXiv preprint arXiv:2410.18558*, 2024. 6

[20] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 6

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 6

[22] Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024. 8, 10

[23] Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, et al. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017. 10

[24] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 1, 10

[25] Weixian Lei, Jiacong Wang, Haochen Wang, Xiangtai Li, Jun Hao Liew, Jiashi Feng, and Zilong Huang. The scalability of simplicity: Empirical analysis of vision-language learning with a single transformer. *arXiv preprint arXiv:2504.10462*, 2025. 1

[26] Houyi Li, Wenzhen Zheng, Jingcheng Hu, Qiufeng Wang, Hanshan Zhang, Zili Wang, Shijie Xuyang, Yuantao Fan, Shuigeng Zhou, Xiangyu Zhang, et al. Predictable scale: Part i–optimal hyperparameter scaling law in large language model pretraining. *arXiv preprint arXiv:2503.04715*, 2025. 1, 10

[27] Jijie Li, Li Du, Hanyu Zhao, Bo-wen Zhang, Liangdong Wang, Boyan Gao, Guang Liu, and Yonghua Lin. Infinity instruct: Scaling instruction selection and synthesis to enhance language models. *arXiv preprint arXiv:2506.11116*, 2025. 6

[28] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018. 2, 10

[29] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. 1

[30] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 6

[31] Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024. 6

[32] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6

[33] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*, 2023. 9

[34] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021. 6

[35] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Anton Belyi, et al. Mm1: methods, analysis and insights from multimodal llm pre-training. In *European Conference on Computer Vision*, pages 304–323. Springer, 2024. 1, 10

[36] Othmane Mounjid and Charles-Albert Lehalle. Improving reinforcement learning algorithms: Towards optimal learning rate policies. *Mathematical Finance*, 34(2):588–621, 2024. 10

[37] Marius-Constantin Popescu, Valentina E Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8(7):579–588, 2009. 10

[38] Fabian Schaipp, Alexander Hägele, Adrien Taylor, Umut Simsekli, and Francis Bach. The surprising agreement between convex optimization theory and learning-rate scheduling for large model training. *arXiv preprint arXiv:2501.18965*, 2025. 2

[39] Yikang Shen, Matthew Stallone, Mayank Mishra, Gaoyuan Zhang, Shawn Tan, Aditya Prasad, Adriana Meza Soria, David D Cox, and Rameswar Panda. Power scheduler: A batch size and token number agnostic learning rate scheduler. *arXiv preprint arXiv:2408.13359*, 2024. 1

[40] Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8317–8326, 2019. 6

[41] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012. 2, 10

[42] Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama. https://www.cerebras.net/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama, 2023. 3, 6

[43] Qwen Team. Qwen2.5: A party of foundation models, September 2024. 3, 6

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 10

[45] Zilong Wang, Mingjie Zhan, Xuebo Liu, and Ding Liang. Docstruct: A multimodal method to extract hierarchy structure in document for general form understanding. *arXiv preprint arXiv:2010.11685*, 2020. 6

[46] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. 1

[47] Dingkang Yang, Dongling Xiao, Jinjie Wei, Mingcheng Li, Zhaoyu Chen, Ke Li, and Lihua Zhang. Improving factuality in large language models via decoding-time hallucinatory and truthful comparators. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25606–25614, 2025. 6

[48] Greg Yang and Edward J Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning*, pages 11727–11737. PMLR, 2021. 1, 10

[49] Greg Yang, Edward J Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv preprint arXiv:2203.03466*, 2022. 1, 10

 https://doi.org/10.52202/085713-3088

[50] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024. 1, 9

[51] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. In *International conference on machine learning*. PMLR, 2024. 6

A AdaLRS Algorithm

In this section, we show the workflow of the proposed AdaLRS algorithm in detail. As shown in Algorithm 1, AdaLRS tracks the velocity of loss descent in pretraining tasks and computes the velocity as the loss curve slope via the least squares method [7]. Learning rate upscaling is triggered when the velocity decays, followed by several validation steps to measure the loss descent velocity after LR adjustment. We then compare the loss descent velocity after LR adjustment with that corresponding to similar loss values before upscaling. If the velocity becomes larger, the upscaling adjustment is retained; otherwise, the model and optimizer states are restored to the exact step before LR upscaling, followed by a LR downscaling adjustment instead. Restoring training states is designed to stabilize the training process, where the upscaled learning rate may disrupt model parameter distribution during training. After several loops of learning rate adjustment, the resulting learning rate eventually falls in the neighborhood of the optimum.

Algorithm 1: Adaptive Learning Rate Scheduling (AdaLRS)

Input: Initial learning rate η_0, Upscale factor $\alpha > 1$, Downscale factor $\beta > 1$ ($\gcd(\alpha, \beta) = 1$), Scale decay factor $\lambda < 1$, Loss slope inspection window size k, Loss slope decay threshold $0 < \theta < 1$, Loss slope estimation error e, AdaLRS start and end step $t_{\text{start}}, t_{\text{end}}$

Output: Optimal learning rate η^*

Function `AdaLRS`(η_0, α, β, k, θ, t_{start}, t_{end})

- **Initialize:** $t \leftarrow 0$, $\eta_t \leftarrow \eta_0$, $\theta \leftarrow \theta_0$
- Initialize loss history $\mathcal{H} \leftarrow \emptyset$
- **while** $t_{start} < t < t_{end}$ **do**
  - `// 1.Collect loss observations`
  - Run k training steps with $\{\eta_t, ..., \eta_{t+k}\}$
  - Record losses $\{l_t, ..., l_{t+k}\}$
  - Update $\mathcal{H} \leftarrow \mathcal{H} \cup \{l_t, ..., l_{t+k}\}$
  - `// 2.Calculate loss descent velocity`
  - $v_t \leftarrow$ least_square_fit($\{(1, l_t), ..., (k, l_{t+k})\}$)
  - `// 3.Check velocity degradation`
  - **if** $|\mathcal{H}| \geq 2k$ ***and*** $v_t < v_{t-k} \cdot \theta$ **then**
    - `// Trigger upscale phase`
    - Run k up-scaling steps with η_{temp} gradually up-scaled to $\alpha\eta_t$
    - `// Find comparable historical records`
    - Run another k validation steps with loss records $\mathcal{L}_{\text{new}} = \{l_{t+2k+1}, ..., l_{t+3k}\}$
    - $\mathcal{L}_{\text{ref}} \leftarrow \underset{\mathcal{L} \subseteq \mathcal{H}}{\operatorname{argmin}} |\bar{\mathcal{L}} - \bar{\mathcal{L}_{\text{new}}}|$
    - `// Compare velocities`
    - **if** $v_{new} > v_{ref} + 2e$ **then**
      - $\eta_{t+k} \leftarrow \max(\alpha\lambda^t, 1)\eta_t$ `// Keep up-scaled LR`
    - **else if** $v_{new} < v_{ref} - 2e$ **then**
      - $\eta_{t+k} \leftarrow \frac{\eta_t}{\max(\beta\lambda^t, 1)}$ `// Downscale after failed upscale`
      - Restore model states to step $t + k$
    - **else**
      - Restore model states to step $t + k$
    - $\mathcal{H}, \theta \leftarrow \emptyset, \theta_0$
  - **else if** $|\mathcal{H}| \geq 2k$ ***and*** $v_t, v_{t-k} < 0$ **then**
    - $\eta_{t+k} \leftarrow \frac{\eta_t}{\max(\beta\lambda^t, 1)}$ `// Downscale if loss rises consecutively`
    - $\mathcal{H}, \theta \leftarrow \emptyset, \theta_0$ Restore model states to step $t + k$
  - **else**
    - $\eta_{t+k} \leftarrow \eta_t$ `// Maintain current LR`
    - $\theta \leftarrow \frac{\theta+1}{2}$ `// Narrow the decay threshold`
  - $t \leftarrow t + k$
- **return** η_t

https://doi.org/10.52202/085713-3088

In the algorithm pseudo code, we omit the following details for presentation simplicity: 1) the learning rate η_t is updated by a base scheduler according to the training step t, multiplied by a scaling factor from AdaLRS; 2) during learning rate up-scaling, an early stopping mechanism is triggered if the training loss elevates to be higher than the maximum loss value in loss history; 3) for corner cases where $\min_{\mathcal{L}\subseteq\mathcal{H}} |\bar{\mathcal{L}} - \bar{\mathcal{L}_{\text{new}}}|$ exceeds a certain threshold, LR up-scaling or down-scaling are triggered if the resulted training loss is higher or lower than all history records.

B Proof for the Validity of LR Scaling Factor Values

In this section, we demonstrate the validity of the multiplicatively independent design of LR scaling factors.

Theorem B.1. *Let $\alpha, \beta > 1$ be multiplicatively independent real numbers (for all integers m, n, $\alpha^m = \beta^n \implies m = n = 0$), for any positive rational number r and $\epsilon > 0$, there exist integers $m, n \geq 0$, such that:*

$$|\alpha^m \beta^{-n} - r| < \epsilon. \tag{9}$$

We start by pointing out that $\frac{\ln \alpha}{\ln \beta}$ is irrational. This is rather straightforward because if $\frac{\ln \alpha}{\ln \beta}$ is rational, then $\alpha^q = \beta^p$, where p, q are positive integers, which contradicts with the multiplicatively independent design, and thus $\frac{\ln \alpha}{\ln \beta}$ must be irrational.

Then we apply Kronecker's Theorem. Let $\theta = \frac{\ln \alpha}{\ln \beta}$. For any real number c and $\epsilon' > 0$, there exist integers $m, n \in \mathbb{N}$ such that $|m\theta - n - c| < \epsilon'$. Choose $c = \frac{\ln r}{\ln \beta}$. Then there exist $m, n \in \mathbb{N}$ satisfying:

$$\left| m\frac{\ln \alpha}{\ln \beta} - n - \frac{\ln r}{\ln \beta} \right| < \epsilon'. \tag{10}$$

Multiplying through by $\ln \beta$, we get:

$$|m \ln \alpha - n \ln \beta - \ln r| < \epsilon' \ln \beta. \tag{11}$$

Finally, let $\epsilon' = \frac{\epsilon}{|q| \ln \beta}$, for sufficiently small ϵ', by the continuity of the exponential function:

$$\left| \frac{\alpha^m}{\beta^n} - r \right| = |r| \cdot \left| e^{m \ln \alpha - n \ln \beta - \ln r} - 1 \right|. \tag{12}$$

Using Equation 11, we bound:

$$\left| \frac{\alpha^m}{\beta^n} - r \right| < |r| \cdot \epsilon' \ln \beta = \epsilon, \tag{13}$$

which proves Theorem B.1.

C Limitations

Despite the promising optimal learning rate search effectiveness and performance improvement, AdaLRS fails to achieve comparable results with appropriate LR baselines with excessively large LR settings. As discussed in Section 3.2 and Section 3.6, we attribute this problem to the disruptive parameter updates performed by large learning rates. This result restricts the application of AdaLRS on arbitrary LR settings. We recommend to apply AdaLRS with relatively small initial learning rates for optimal performance, which is validated to work smoothly in both from-scratch and continual pretraining, and leave this problem to be solved in future works.

Moreover, the design of AdaLRS prioritizes the generalizability to unexplored foundation model pretraining tasks, and therefore we do not study the extent to which AdaLRS can approximate the optimal LR in this work. We will explore the impact of the scaling factor designs and learning rate search ranges to demonstrate the fine-grained convergence of AdaLRS in our future work.

We also point out that the near-optimal learning rates used in Fit LR experiments in Section 3 are obtained from grid search pilot experiments. We search for the optimal LR setting with exponentially larger LR in a given range, such as $[2e-6, 4e-6, 8e-6, 2e-5, ..., 8e-3]$ for LLM experiments. Although sufficiently precise for our experiments, the resulting approximated search result may introduce slight variance in optimal LR estimation.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

https://doi.org/10.52202/085713-3088

Justification: We discuss the limitations of our work in Appendix C.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide theoretical proof and experiment results to validate our hypotheses and claims in Section 2.2.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have included necessary details for our experiments.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will release the proposed benchmark and code after the paper is accepted.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

 https://doi.org/10.52202/085713-3088

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We show all necessary training and test details in our paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We follow the same experiment settings with baseline works.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide compute resource details in Section 3.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact that our work may cause. Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper poses no such risks.

https://doi.org/10.52202/085713-3088

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the creators or original owners of assets used in the paper credited properly, and the license and terms of use explicitly mentioned and are respected properly. All datasets we use are from internet open source datasets under CC-BY licenses and are cited properly.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: No new assets are introduced in our paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Our paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Our paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [No]

Justification: LLMs are not used in any of the core methods in our research.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

https://doi.org/10.52202/085713-3088