Surge Phenomenon in Optimal Learning Rate and Batch Size Scaling

Shuaipeng Li[*,1,†], **Penghao Zhao**[*,1,2], **Hailin Zhang**[*,1,2], **Xingwu Sun**[*,1,3], **Hao Wu**[1], **Dian Jiao**[1], **Weiyan Wang**[1], **Chengjun Liu**[1], **Zheng Fang**[1], **Jinbao Xue**[1], **Yangyu Tao**[1], **Bin Cui**[2,4†], **Di Wang**[1,†]

[1] Tencent Hunyuan
[2] School of Computer Science & Key Lab of High Confidence
Software Technologies (MOE), Peking University
[3] University of Macau
[4] Institute of Computational Social Science, Peking University (Qingdao)

Abstract

In current deep learning tasks, Adam-style optimizers—such as Adam, Adagrad, RMSprop, Adafactor, and Lion—have been widely used as alternatives to SGD-style optimizers. These optimizers typically update model parameters using the sign of gradients, resulting in more stable convergence curves. The learning rate and the batch size are the most critical hyperparameters for optimizers, which require careful tuning to enable effective convergence. Previous research has shown that the optimal learning rate increases linearly (or follows similar rules) with batch size for SGD-style optimizers. However, this conclusion is not applicable to Adam-style optimizers. In this paper, we elucidate the connection between optimal learning rates and batch sizes for Adam-style optimizers through both theoretical analysis and extensive experiments. First, we raise the scaling law between batch sizes and optimal learning rates in the "sign of gradient" case, in which we prove that the optimal learning rate first rises and then falls as the batch size increases. Moreover, the peak value of the surge will gradually move toward the larger batch size as training progresses. Second, we conduct experiments on various CV and NLP tasks and verify the correctness of the scaling law.

1 Introduction

Deep learning techniques, initiated by Stochastic Gradient Descent (SGD) learning on large datasets, have significantly revolutionized various real-world applications [1]. Over the past decade, numerous optimizers, such as momentum [2], Adagrad [3], ADADELTA [4], RMSprop [5], Adam [6], Adafactor [7], and Lion [8], have been introduced to stabilize the iterative learning process and expedite convergence. Among them, the Adam optimizer is the most widely used across various domains including Computer Vision (CV) [9–11], Natural Language Processing (NLP) [12–15] and many others [16, 17]. It retains the first and second moment information of parameters to facilitate adaptive learning step size. Unlike SGD-style optimizers that use the raw gradient to determine the learning direction and step size, Adam and its variants (Adagrad, RMSprop, Lion, etc.) employ the sign of gradient for this purpose, thereby ensuring greater robustness [18].

Beyond specific hyper-parameters in optimizer configurations, the batch size and the learning rate are the most critical hyperparameters influencing convergence. As the scale of training datasets in various

[*] Equal contribution. [†] Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

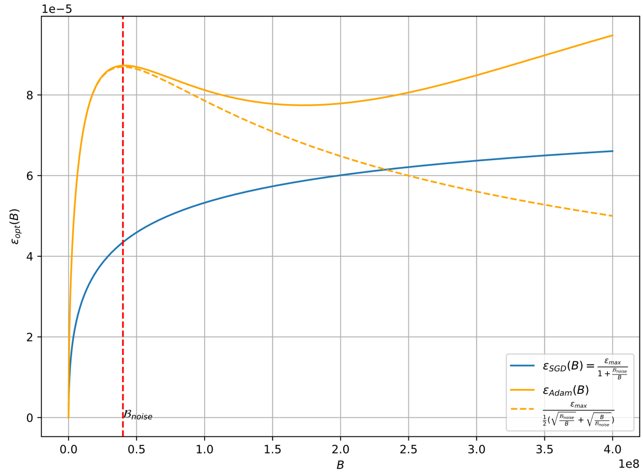


Figure 1: The relationship between the optimal learning rate and the batch size is different between Adam and SGD. The orange line represents the tendency of the optimal learning rate to converge to a non-zero value when the batch size is large enough.

workloads (e.g. CV [19, 20], NLP [21, 14], and others) continues to grow, there is an increasing demand for large batch size training across multiple data parallel workers. However, large batch training presents great challenges for robust training and meticulous tuning. The learning rate, which determines the actual step size in each learning iteration, is highly dependent on the batch size used. Prior research has explored methods to determine an optimal learning rate according to the batch size in scenarios utilizing SGD optimizers, including square root scaling [22], linear scaling [19, 23], and others [24]. Among these, the empirical model focusing on large batch training [24] yields convincing results in both theoretical and empirical contexts, proposing the following rule to depict the relationship between the optimal learning rate and the batch size:

$$\epsilon_{opt}(B) = \frac{\epsilon_{max}}{1 + \frac{\mathcal{B}_{noise}}{B}} \tag{1}$$

For Adam-style optimizers, though existing works also provide some approximation [24, 23], they fail to capture the true scaling law of optimal learning rates with batch sizes. For illustrative purposes, Figure 1 presents curves that simulate the optimal learning rates for the Adam optimizer. We find that, in scenarios involving small batch sizes, the optimal learning rate initially increases and then decreases, resembling a surge in the sea, as depicted by the dashed orange line. For large batch sizes, we identify a value to which the optimal learning rate converges. The solid orange line represents a schematic based on our findings for both small and large batch sizes, showing that the learning rate tends to rise initially, then decrease, and subsequently gradually ascend to asymptotically approach a stable value. For clarity in visualization, we have omitted the final asymptotic portion of the curve.

In this paper, we aim to elucidate and formalize the connection between optimal learning rates and batch sizes for Adam-style optimizers. By following the notation from the empirical model [24] and conducting a more in-depth theoretical analysis, we discover that the relationship between the optimal learning rate and the batch size in the above parameter update formular satisfies:

$$\epsilon_{opt}(B) = \frac{\epsilon_{max}}{\frac{1}{2}\left(\sqrt{\frac{\mathcal{B}_{noise}}{B}} + \sqrt{\frac{B}{\mathcal{B}_{noise}}}\right)}, \tag{2}$$

which differs from SGD, especially when the batch size is not too large. Here the meaning of $\mathcal{B}_{noise}$ is consistent with papers of scaling laws [24, 25], representing the trade-off point between training speed and data efficiency. When the batch size is equal to $\mathcal{B}_{noise}$, the optimal learning rate reaches a local maximum in accordance with Eq 2. Furthermore, we provide additional proof that when the batch size becomes significantly large, the optimal learning rate gradually converges to a non-zero value. We also prove that the previous conclusions about training speed and data efficiency are still valid for Adam-style optimizers, and the variable $\mathcal{B}_{noise}$ gradually increases as the training progresses. It is worth noting that when $B \ll \mathcal{B}_{noise}$, for SGD, the scaling law of optimal learning rates with batch sizes transitions into linear scaling, consistent with previous conclusions [19, 23]:

$$\epsilon_{opt}(B) \approx \frac{\epsilon_{max}}{\mathcal{B}_{noise}} B; \tag{3}$$

while for Adam, the relationship transitions into square root scaling, aligning with previous approximations [23, 24]:

$$\epsilon_{opt}(B) \approx \frac{2\epsilon_{max}}{\sqrt{\mathcal{B}_{noise}}} \sqrt{B}. \tag{4}$$

In addition to theoretical analysis, our extensive empirical study on various CV and NLP workloads further validate the conclusions. The true optimal learning rate, across different Adam configurations, exhibits a clear downward trend after reaching its peak value as the batch size increases. This behavior contradicts previous research, but demonstrates the correctness and generalizability of our theory. The experiments also reveal a gradual increase in the variable $\mathcal{B}_{noise}$, corresponding to the peak optimal learning rate, as training progresses.

2 Theorems

2.1 Batch Size and Optimal Learning Rate

In this section, we theoretically derive the optimal learning rate for a given batch size. Initially, we introduce an approximation of Adam-style optimizers. In alignment with the insights elucidated in [26], a thorough examination of the Adam optimizer and its variants reveals that their primary distinction from SGD resides in the utilization of the gradient's sign for updates during each iteration, as opposed to the gradient itself:

$$\theta_{i+1} = \theta_i - \epsilon \cdot sign(G_{est}), \tag{5}$$

where θ_t is the parameter at time t, G_{est} is the gradient estimated via mini-batch, and ϵ is the learning rate. As the batch size increases, the expected value of the update amount tends to saturate. For example, assuming that the mean value of the gradient is positive, when the accumulated gradient of the mini-batch is positive, increasing the batch size will have no contribution to the signed update amount. This is significantly different from the behavior of SGD where the larger the batch size, the more accurate the gradient estimate. In Appendix A, we provide a detailed discussion on this approximation for the Adam optimizer. Next, we derive the optimal learning rate that maximizes the loss improvement. And then we establish a lemma that addresses the optimal learning rate given an estimated mini-batch gradient:

Lemma 1. *Suppose that we are updating the parameter θ using the mini-batch gradient V, with the true gradient being G and the true Hessian being H. Then the optimal learning rate that maximizes the decrease in loss is:*

$$\epsilon_{opt} \equiv argmax_{\epsilon}\, \mathbb{E}[\Delta L] = \frac{G^T \mathbb{E}[V]}{tr[H \cdot cov(V)] + \mathbb{E}[V]^T H \mathbb{E}[V]}, \tag{6}$$

and the corresponding loss improvement ΔL is:

$$\Delta L_{opt} = \frac{G^T \mathbb{E}[V]}{2} \epsilon_{opt}. \tag{7}$$

The proof is in Appendix B. Although our conclusion is based on an approximation, we adopt the equal sign here to simplify the analysis, following the notation used in previous work [24].

Now let's consider the case where $V = sign(G_{est})$, and assuming that the estimated gradient G_{est} follows a Gaussian distribution. The Gaussian distribution assumption is motivated by the following: if the mini batch size is sufficiently large, we can invoke the Central Limit Theorem (CLT) and approximate the distribution as Gaussian - a common assumption in previous research [27–30]. Furthermore, our experimental results confirm that the gradient distributions closely approximate Gaussian distributions, as illustrated in Figure 8 of Appendix H. We have the following theorem:

Theorem 2. *Suppose the gradient of parameter i for each sample follows a Gaussian distribution with mean μ_i and variance σ_i^2, the expected loss improvement is:*

$$\Delta L_{opt} = \frac{1}{2} \frac{\sum_i \sum_j \mathcal{E}_i \mathcal{E}_j \mu_i \mu_j}{\sum_i (1 - \mathcal{E}_i^2) H_{i,i} + \sum_i \sum_j \mathcal{E}_i \mathcal{E}_j H_{i,j}}, \tag{8}$$

and the corresponding optimal learning rate is

$$\epsilon_{opt} = \frac{\sum_i \mathcal{E}_i \mu_i}{\sum_i (1-\mathcal{E}_i^2) H_{i,i} + \sum_i \sum_j \mathcal{E}_i \mathcal{E}_j H_{i,j}}, \tag{9}$$

where $\mathcal{E}_i$ is a function (derived from the Gauss error function) with respect to the batch size B:

$$\mathcal{E}_i(B) = \frac{2}{\sqrt{\pi}} \int_0^{\sqrt{\frac{B}{2}}\frac{\mu_i}{\sigma_i}} e^{-t^2} dt \approx \frac{\frac{\mu_i}{\sigma_i}}{\sqrt{\frac{\pi}{2B} + \left(\frac{\mu_i}{\sigma_i}\right)^2}}. \tag{10}$$

We prove the above theorem in Appendix C.

An important observation from the proof is that, not only is the covariance matrix of $sign(G_{est})$ related to B, but its expected value also depends on B. This implies that in Eq 6 the numerator is the first-order form of the function about B, and the denominator is the second-order form of the function about B:

$$\epsilon(B) = \frac{\beta f(B)}{f(B)^2 + \gamma} = \frac{\beta}{f(B) + \frac{\gamma}{f(B)}}. \tag{11}$$

Therefore, the conclusion in the case of Adam cannot be derived by simply following the form mentioned in [24]:

$$\epsilon(B) \neq \frac{\epsilon_*}{\left(1 + \frac{\mathcal{B}_{noise}}{B}\right)^\alpha}. \tag{12}$$

Then we aim to derive the specific expression for the optimal learning rate with respect to the batch size through the following theorems.

Theorem 3. *When $B \ll \frac{\pi\sigma_i^2}{2\mu_i^2}$, the optimal learning rate is a function with respect to batch size B:*

$$\epsilon_{opt}(B) \approx \frac{1}{\frac{1}{2}\left(\sqrt{\frac{\mathcal{B}_{noise}}{B}} + \sqrt{\frac{B}{\mathcal{B}_{noise}}}\right)} \frac{\sqrt{\frac{\mathcal{B}_{noise}}{2\pi}} \sum_i \frac{\mu_i^2}{\sigma_i}}{\sum_i H_{i,i}} \leq \frac{\sqrt{\frac{\mathcal{B}_{noise}}{2\pi}} \sum_i \frac{\mu_i^2}{\sigma_i}}{\sum_i H_{i,i}} \quad , \tag{13}$$

where $\mathcal{B}_{noise}$ is a variable unrelated to batch size B:

$$\mathcal{B}_{noise} = \frac{\pi \sum_i H_{i,i}}{2 \sum_i \sum_j \begin{cases} \frac{\mu_i \mu_j}{\sigma_i \sigma_j} & i \neq j \\ 0 & i = j \end{cases} H_{i,j}} \quad . \tag{14}$$

Defining B_{peak} as the batch size at which the optimal learning rate reaches a peak value, it is obvious that:

$$B_{peak} = \mathcal{B}_{noise}. \tag{15}$$

The peak value is:

$$\epsilon_{max} = \frac{\sqrt{\frac{\mathcal{B}_{noise}}{2\pi}} \sum_i \frac{\mu_i^2}{\sigma_i}}{\sum_i H_{i,i}}. \tag{16}$$

We prove the theorem in Appendix D. From the theorem we can finally get Eq 2, which implies that there is an interval where the batch size becomes larger and the optimal learning rate needs to be reduced. Considering that $\frac{\pi\sigma_i^2}{2\mu_i^2}$ is much larger than normal batch sizes in research and industry (as shown in Figure 2), this theorem can cover most of the scenarios. To make the conclusion more comprehensive, we also derive the following theorem:

Theorem 4. *When $B \gg \frac{\pi\sigma_i^2}{2\mu_i^2}$, the optimal learning rate becomes:*

$$\epsilon_{opt} = \frac{\sum_i |\mu_i|}{\sum_i \sum_j sign(\mu_i) sign(\mu_j) H_{i,j}}. \tag{17}$$

We prove the theorem in Appendix E.

Therefore, when B increases infinitely, the optimal learning rate will eventually converge to a non-zero value. If we make an (unrealistic) assumption that $\frac{\mu_i}{\sigma_i} \approx sign(\mu_i)$, we will find that the lower bound of ϵ_{max} in Theorem 3 will become the one in Theorem 4, which means that the local peak value of the optimal learning rate is larger than the final convergence value. However, considering that the variance of the gradient in the later stages of training is very small, which makes the above conclusion $\frac{\mu_i}{\sigma_i} \approx sign(\mu_i)$ difficult to establish, so the stable value in the later stages of training is more likely to exceed the local maximum. We provide a reference curve in Figure 1.

2.2 Data/Time Efficiency Trade-off

Following the empirical model for large-batch training [24], we also review the trade-off between data and time efficiency during batch size selection. We have the following theorem:

Theorem 5. *When $B \ll \frac{\pi\sigma_i^2}{2\mu_i^2}$, the derived loss improvement with respect to the batch size is*

$$\Delta L_{opt}(B) = \frac{\Delta L_{max}}{1 + \frac{\mathcal{B}_{noise}}{B}}, \tag{18}$$

where ΔL_{max} is defined as

$$\Delta L_{max} = \frac{\sum_i \sum_j \frac{\mu_i^2 \mu_j^2}{\sigma_i \sigma_j}}{2 \sum_i \sum_j \begin{cases} \frac{\mu_i \mu_j}{\sigma_i \sigma_j} & i \neq j \\ 0 & i = j \end{cases} H_{i,j}}, \tag{19}$$

We prove the theorem in Appendix F. This result aligns with the conclusion drawn in the SGD situation [24], indicating that many related conclusions also remain valid.

It has been concluded in previous work [24] that, when using the SGD optimizer with the same form as Eq 18, the relationship between training speed (number of steps S) and data efficiency (number of samples E) is given by:

$$\left(\frac{S}{S_{min}} - 1\right)\left(\frac{E}{E_{min}} - 1\right) = 1. \tag{20}$$

Here $S_{(min)}$ represents training speed, the actual (minimum) possible number of steps taken to reach a specified model performance; and $E_{(min)}$ represents data efficiency, the actual (minimum) possible number of training examples processed to reach that same level of performance. For more details, please refer to the Eq 2.11 and the Appendix D in [24]. Additionally, as referenced in the Eq 2.12 in [24] and Eq 1.4 in [25], $\mathcal{B}_{noise}$ is the balance point between training speed and data efficiency:

$$\mathcal{B}_{noise} \approx \mathcal{B}_{crit} = \frac{E_{min}}{S_{min}} \approx \frac{B_*}{L^{\frac{1}{\alpha_B}}}. \tag{21}$$

Since in Adam optimizer we arrive at the same Eq 18 as in SGD optimizer, the above equations 20 and 21 still hold. In Adam scenarios, $B_{peak} = \mathcal{B}_{noise}$ is not only the local maximum of the optimal learning rate, but also the balance point between training speed and data efficiency. Moreover, as training progresses and the loss decreases, according to Eq 21, B_{peak} will gradually becomes larger.

2.3 Summary

In this section, we have drawn several conclusions from our theoretical analysis, which are summarized as follows:

1. As the batch size increases, the optimal learning rate demonstrates a decreasing trend within a specified range (Eq 2).
2. The batch size that corresponds to the local maximum optimal learning rate is consistent with the balance point of training speed and data efficiency (Eq 21). As the training progresses and the loss decreases, B_{peak} will gradually becomes larger.

3 Experiments

In this section, we carry out a series of experiments to corroborate the theoretical scaling law we proposed in Section 2 and detail the experimental workloads and configurations in Section 3.1. The process for deriving the estimated variables from our theory is elucidated in Section 3.2. We also showcase and dissect the applicability of our scaling law across a variety of workloads in Section 3.3.

3.1 Experimental Setup

Workloads. In our empirical study, we incorporate 4 open-source workloads that are extensively utilized: (1) training a 5-layer CNN model on the Fashion-MNIST [31], which is a typical CV test case to start with. It consists of 60000 28x28 grayscale images in 10 classes; (2) training a ResNet-18 model [32] on the Tiny-ImageNet dataset [33], which contains 100000 images of 200 classes (500 for each class) downsized to 64×64 colored images. In each epoch we train the model with random 10k samples to reduce the overall complexity; (3) training a dense Transformer model [12] (simplified DistilGPT2 [34]) on the ELI5-Category dataset [35], which is a smaller but newer and categorized version of the original ELI5 dataset [36]. It contains 10.5k complex, diverse questions that require explanatory multi-sentence answers. (4) training a fine-grained Mixture-of-Experts (MoE) model, similar in structure to Mistral-MoE [37] but with shared experts [38], on the RedPajama-v2 dataset [39], which contains 30 trillion filtered and deduplicated tokens (100+ trillions raw) from 84 CommonCrawl dumps covering 5 languages, along with 40+ pre-computed data quality annotations. These workloads are popular in both academia and industry, covering typical deep learning tasks in the domains of CV and NLP.

Batch sizes and learning rates. To showcase the optimal learning rate for each batch size configuration, we leverage a grid-search-style experiments set. Each point in the grid search corresponds to a certain round with the same configuration but a different random number seed. The start point, stop point, and the interval of different workloads are listed in Table 1. In NLP tasks, the term "batch size" refers to the number of tokens in a batch, as practiced in related works [25].

Table 1: Grid search configurations.

Workload	**Adam**		**Learning Rate**			**Batch Size**			**Round**
	β_1	β_2	**Start**	**Stop**	**Step**	**Start**	**Stop**	**Step**	
CNN	0.9	0.999	1e-4	1e-3	1e-4	1	12	1	100
CNN	0.9	0.999	1e-4	1e-3	1e-4	64	1164	100	100
DistilGPT2	0.9	0.999	1e-5	1.09e-3	1.2e-4	4	114	10	30
DistilGPT2	0.0	0.0	1e-5	5.5e-4	6e-5	4	114	10	30
ResNet18	0.0	0.0	1e-4	7.876e-4	7.65e-5	16	376	33	100
MoE	0.9	0.999	2e-6	6e-5	2e-6	192k	12M	1.2M	10

Hyper-parameters. Since we derive the theorems on Adam-style optimizers, we conduct experiments using the Adam optimizer. We experiment on both the "sign of gradient" configuration ($\beta_1 = 0$, $\beta_2 = 0$) and the default hyper-parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$), as shown in Table 1.

Hardware environment. We execute each round of experiments utilizing an NVIDIA A100 card. The training time of each round for the datasets are approximately 1 hour for Fashion-MNIST, 1.5 hours for TinyImageNet, 2 hours for ELI5-Category and 11 hours for RedPajama-v2. Given our primary focus on the convergence process, the specific hardware environment does not matter in our experiments. Our theoretical and empirical findings can be generalized to other hardware settings. Additionally, some system optimizations [40–43] are also beneficial to enhancing training efficiency.

3.2 Variable Estimation

We try to estimate the value of $\mathcal{B}_{noise}$ and the expectation of ϵ_{max} through curve fitting. After using Eq 21 to simplify Eq 20 (see Appendix G for details), we can record the actual possible number of steps taken S and the actual possible number of training examples processed E to reach a specified level of performance corresponding to the optimal learning rate of each batch size in the grid search results, and then perform linear fitting to obtain the estimated value of $\mathcal{B}_{noise}$:

$$\frac{1}{S} = -\mathcal{B}_{noise}\frac{1}{E} + \frac{1}{S_{min}} \tag{22}$$

Subsequently, we use the optimal learning rate and batch size of the grid search results to estimate the max optimal learning rate of Adam-style $\mathbb{E}[\epsilon_{max}]_{Adam}$:

$$\mathbb{E}[\epsilon_{max}]_{Adam} = \mathbb{E}\left[\frac{\epsilon_{opt}}{2}\left(\sqrt{\frac{\mathcal{B}_{noise}}{B}} + \sqrt{\frac{B}{\mathcal{B}_{noise}}}\right)\right] \tag{23}$$

and SGD-style $\mathbb{E}[\epsilon_{max}]_{SGD}$:

$$\mathbb{E}[\epsilon_{max}]_{SGD} = \mathbb{E}\left[\epsilon_{opt}\left(1 + \frac{\mathcal{B}_{noise}}{B}\right)^{\alpha}\right] \tag{24}$$

Previous research [24] represents the SGD optimizer and the Adam optimizer using Eq 24 with $\alpha = 1$ and $\alpha = 0.5$, respectively. We include these fitted curves as comparisons in the following section.

While we use grid search to estimate the value of $\mathcal{B}_{noise}$, in practice we can efficiently approximate it using the scaling law from previous studies [24, 25], where $\mathcal{B}_{noise} \approx B_{crit} = \frac{B_*}{L^{1/\alpha}}$. With this approximation, we only need a simple search for a pair of (batch size, optimal learning rate) to determine the final hyperparameter ϵ_{max}. Therefore, the costly grid-search can be avoided.

3.3 Results

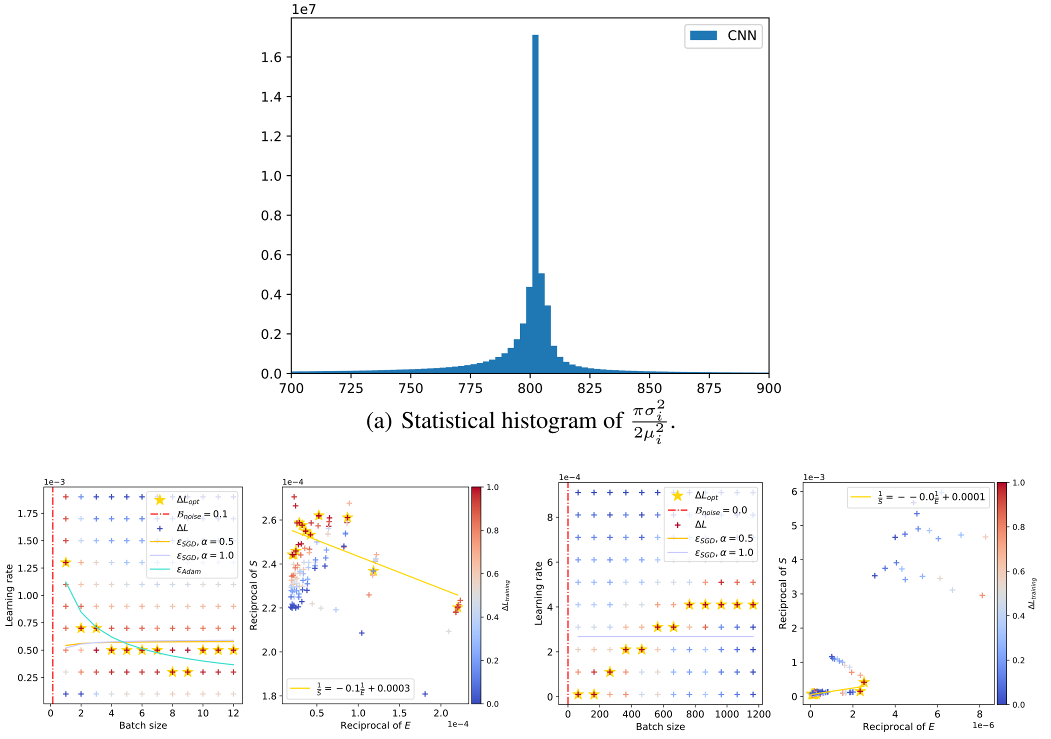
Following Section 3.2, we first estimate the variables and fit the curves using observations, then conduct grid-search-style experiments for learning rates and batch sizes.

Figure 2, 3, 4, 5 illustrate the experimental results of CNN-FashionMNIST, ResNet18-TinyImageNet, DistilGPT2-ELI5Category and MoE-RedPajama-v2, respectively. Each figure is divided into two parts: the left subfigure illustrates the grid-search results for batch sizes and learning rates, and these data points are utilized to fit the curve of Eq 22 in the right subfigure. In order to estimate the variables, we train models from scratch using different learning rates and batch sizes, then record the number of steps S and examples E in each experiment to achieve an equivalent training loss. Using the recorded S and E, we fit the curve in the right subfigure and obtain the estimated $\mathcal{B}_{noise}$. In the left subfigure, upon achieving the desired training loss, all experiments continue to train the same number of steps. Any subsequent decrease in training loss is represented through different colors, as indicated in the color bar. For each batch size, we highlight the optimal learning rate that results in the most significant reduction in training loss. We also plot the batch size $\mathcal{B}_{noise}$ that corresponds to the peak optimal learning rate, the fitted SGD curves with $\alpha = 0.5$ and $\alpha = 1$ as derived from previous research [24], and the fitted Adam curve as derived from our theorems.

For the CNN-FashionMNIST workload, we train exactly 10 more step after achieving the desired training loss. As shown in Figure 2(a), the batch size bound $\frac{\pi\sigma_i^2}{2\mu_i^2}$ for Theorem 3 is around 800 in this task. Given the simplicity of the CNN-FashionMNIST workload, commonly-used batch sizes are usually smaller than the batch size bound. We plot the situations corresponding to Theorem 3 and Theorem 4 in Figure 2(b) and 2(c), respectively. In both cases, the trend predicted by our theory is consistent with the actual optimal learning rate performance, showing a declining range at small batch sizes and a saturation range at large batch sizes.

For the ResNet18-TinyImageNet workload, we train 50 more steps after achieving the desired training loss. We plot the figures for Theorem 3 at different achieved training losses, which represent the progress of training, as shown in Figure 3. The observed optimal learning rates primarily exhibit a downward trend after the batch size exceeds the estimated $\mathcal{B}_{noise}$. Although the SGD curve with $\alpha = 0.5$, which is claimed by [24] to represent the Adam optimizer, serves as a good approximation in certain cases, it fails to capture the peak optimal learning rate as our Adam curve does. Comparing the red dashed lines in different figures, we can see that the estimated $\mathcal{B}_{noise}$ gradually increases as the training progresses (i.e. training loss decreases), which corroborates the second conclusion in Section 2.3.

For the DistilGPT2-Eli5Category workload, we train 50 more steps after achieving the desired training loss. As shown in Figure 4, we test on two distinct Adam configurations for Theorem 3: the first with $\beta_1 = 0.0$, $\beta_2 = 0.0$, and the second with $\beta_1 = 0.9$, $\beta_2 = 0.999$. In both configurations, promising learning rates that lead to a substantial decrease in loss are consistent with our Adam curve. It is worth noting that another curve, SGD with $\alpha = 0.5$ [24], also provides a suitable approximation in this scenario. To more clearly demonstrate the accuracy of our theoretical predictions, we present



(a) Statistical histogram of $\frac{\pi\sigma_i^2}{2\mu_i^2}$.

(b) The situation of small batch size in Theorem 3.

(c) The situation of large batch size in Theorem 4.

Figure 2: Batch size versus optimal learning rate within the context of CNN trained on FashionMNIST.

detailed results from a finer-grained grid search in Figure 6 of Appendix H. These experiments demonstrate that our theorems can be generalized to different optimizer configurations, validating the analysis in Appendix A.

For the sparse MoE model using the RedPajama-v2 dataset, we train 300 more steps after achieving the desired training loss. Figure 5 demonstrates that our predictions on optimal learning rate are both accurate and appropriate.

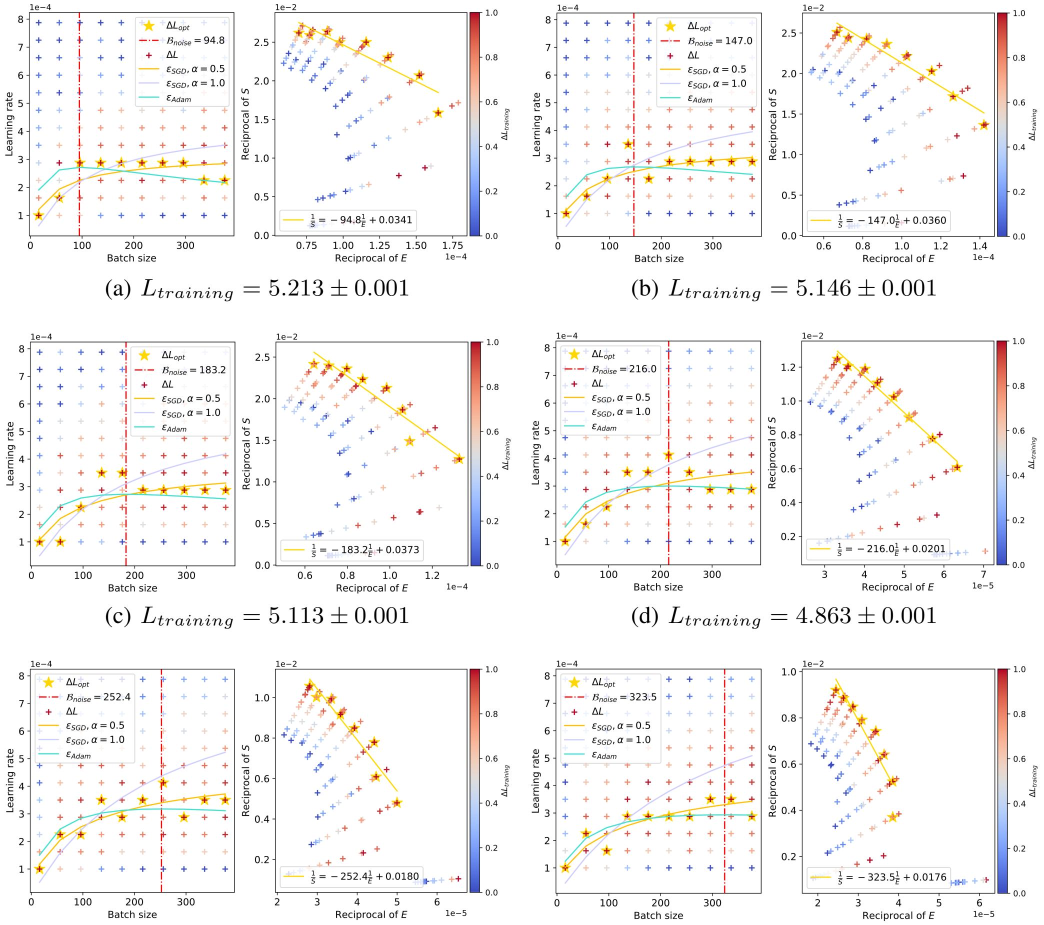
In addition to the above workloads, we also conduct an analysis of experimental results from third parties, confirming that our conclusions remain valid. Detailed results are presented in Figure 7 of Appendix H.

4 Discussion

We have carried out empirical studies on representative workloads using the Adam optimizer. Our investigation into the scaling laws of learning rates relative to batch sizes has provided deeper insights into the training dynamics of deep learning models. This understanding can help fine-tune hyperparameters, enhance convergence speeds, and circumvent exhaustive grid searches. By leveraging prior knowledge that the optimal learning rate decreases after reaching a peak, researchers and engineers can more effectively adjust the learning rate to achieve efficient training outcomes.

In real-world applications, there are numerous different learning workloads [44–47]. Other factors, beyond the scope of this paper, may influence the learning process - the specific optimizer used, weight decay, gradient clipping, etc. While we assert that our theorem can be applied to numerous practical scenarios, it may not fully encompass all situations involving intricate training configurations.

As one of our conclusions points out, the variable $\mathcal{B}_{noise}$ will gradually increases as the training progresses. It is natural to implement adaptive learning rates (and batch sizes) if possible, to speed up the training process. As mentioned in [24], using adaptive batch sizes and warmed-up learning rates brings considerable benefits. Fully exploring the potential of batch size and learning rate scheduling requires meticulous design, which we leave as future work.



(a) $L_{training} = 5.213 \pm 0.001$

(b) $L_{training} = 5.146 \pm 0.001$

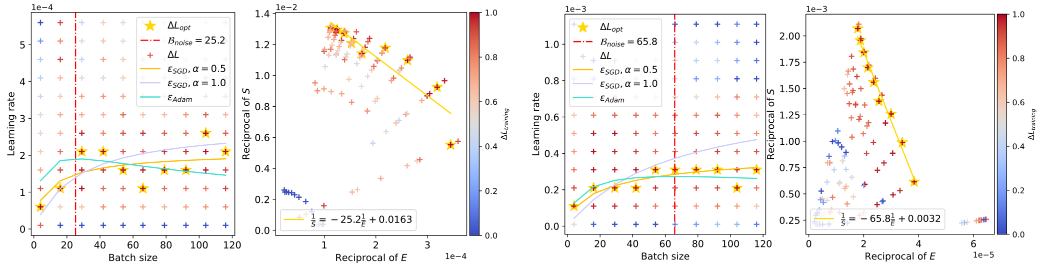
(c) $L_{training} = 5.113 \pm 0.001$

(d) $L_{training} = 4.863 \pm 0.001$

(e) $L_{training} = 4.741 \pm 0.001$

(f) $L_{training} = 4.633 \pm 0.001$

Figure 3: The relationship between batch sizes and optimal learning rates within the context of ResNet-18 trained on TinyImageNet. The red dashed line accurately predicts the peak value, and as the training loss decreases, the peak value gradually shifts to the right.



(a) $\beta_1 = 0.0, \beta_2 = 0.0$

(b) $\beta_1 = 0.9, \beta_2 = 0.999$

Figure 4: The relationship between batch sizes and optimal learning rates within the context of DistilGPT2 trained on Eli5Category.

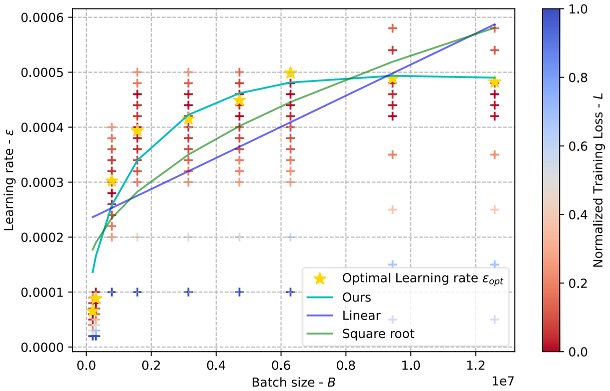


Figure 5: Grid search results for the MoE [37, 38] structure model.

Our theory is based on the quadratic approximation of the loss function. Experimental results demonstrate that conclusions drawn from this second-order expansion effectively predict the surge phenomenon observed in most mainstream scenarios. However, recent studies [48–50] have proposed that quadratic approximations do not accurately capture the loss in scenarios involving large learning rates. We acknowledge the potential benefits of exploring higher-order approximations and consider this a promising direction for future research.

5 Related Work

Aiming to accelerate convergence, our work analyzes the scaling law of optimal learning rates with respect to batch sizes for Adam-style optimizers. Numerous related studies have been proposed to enhance the convergence of deep learning tasks by investigating optimal learning rates, developing new optimizers, and analyzing gradient noise.

Earlier works have proposed various scaling laws to tune learning rates for SGD-style optimizers, including square root scaling [22], linear scaling [19, 23], and others [24]. They also obtained a scaling law for Adam-style optimizers [24, 23] through approximation, revealing a square root-like relationship where the optimal learning rate monotonically increases with the batch size. However, as illustrated in Section 1 and 2, their analysis holds only for small batch sizes, whereas the true scaling law exhibits greater complexity, with the optimal learning rate reaching a peak value at a balanced batch size.

There are many meticulously designed optimizers for various tasks and scenarios. First-order optimizers dominate nowadays deep learning models, including adaptive methods [3–7], sign-based methods [18, 8], layer-wise methods (for large-batch training) [51, 52]. Second-order optimizers [53–55], though with stronger theoretical guarantees, are not efficient for large-scale models due to quadratic complexity with respect to the number of parameters. Despite the emergence of new optimizers, empirical evidence confirms that Adam has remained the most widely used and effective optimizer over the past decade.

Our analysis is inspired by the empirical model of large-batch training [24], which predicts the useful batch size using the gradient noise scale. Gradient noise can help with learning rate determination [56], batch size selection [57, 58], and gaining deeper insights into the convergence process [59–62].

6 Conclusion

In this paper, we established a scaling law between optimal learning rates and batch sizes for Adam-style optimizers. We theoretically proved that the optimal learning rate initially increases and then decreases as the batch size grows, and that the peak value of the surge represents a trade-off point between training speed and data efficiency. Through extensive experiments, we validated our theory on diverse deep learning models and datasets.

Acknowledgments and Disclosure of Funding

This work is supported by National Science and Technology Major Project (2022ZD0116315), National Natural Science Foundation of China (U22B2037, U23B2048), Beijing Municipal Science and Technology Project (Z231100010323002), research grant No. SH-2024JK29, PKU-Tencent joint research Lab, and High-performance Computing Platform of Peking University. Bin Cui, Shuaipeng Li and Di Wang are the corresponding authors.

References

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[2] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, pages 1139–1147, 2013.

[3] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12(7), 2011.

[4] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[5] Tijmen Tieleman and Geoffrey Hinton. Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. *COURSERA Neural Networks Mach. Learn*, 17, 2012.

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[7] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *ICML*, pages 4596–4604, 2018.

[8] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, et al. Symbolic discovery of optimization algorithms. In *NeurIPS*, 2023.

[9] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *NeurIPS*, 30, 2017.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[11] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, pages 12179–12188, 2021.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019.

[14] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[15] An Wang, Xingwu Sun, Ruobing Xie, Shuaipeng Li, Jiaqi Zhu, Zhen Yang, Pinxue Zhao, JN Han, Zhanhui Kang, Di Wang, et al. Hmoe: Heterogeneous mixture of experts for language modeling. *arXiv preprint arXiv:2408.10681*, 2024.

[16] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[17] Hailin Zhang, Yujing Wang, Qi Chen, Ruiheng Chang, Ting Zhang, Ziming Miao, Yingyan Hou, Yang Ding, Xupeng Miao, Haonan Wang, et al. Model-enhanced vector index. *NeurIPS*, 36, 2024.

[18] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *ICML*, pages 560–569. PMLR, 2018.

[19] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[20] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021.

[21] 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.

[22] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014.

[23] Diego Granziol, Stefan Zohren, and Stephen Roberts. Learning rates as a function of batch size: A random matrix theory approach to neural network training. *JMLR*, 23(173):1–65, 2022.

[24] Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018.

[25] 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.

[26] Lukas Balles and Philipp Hennig. Dissecting adam: The sign, magnitude and variance of stochastic gradients. In *ICML*, volume 80, pages 413–422, 2018.

[27] Stephan Mandt, Matthew Hoffman, and David Blei. A variational analysis of stochastic gradient algorithms. In *ICML*, pages 354–363. PMLR, 2016.

[28] Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

[29] Qianxiao Li, Cheng Tai, and E Weinan. Stochastic modified equations and adaptive stochastic gradient algorithms. In *ICML*, pages 2101–2110, 2017.

[30] Zhanxing Zhu, Jingfeng Wu, Bing Yu, Lei Wu, and Jinwen Ma. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. In *ICML*, volume 97, pages 7654–7663, 2019.

[31] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[33] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[34] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC^2 Workshop*, 2019.

[35] Jingsong Gao, Qingren Zhou, and Rui Qiu. ELI5-Category: a categorized open-domain qa dataset. 2021.

[36] Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. ELI5: long form question answering. In *ACL*, pages 3558–3567, 2019.

[37] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[38] Damai Dai, Chengqi Deng, Chenggang Zhao, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. In *ACL*, pages 1280–1297, 2024.

[39] Together Computer. Redpajama: an open dataset for training large language models, 2023.

[40] Guozheng Wang, Yongmei Lei, Zeyu Zhang, and Cunlu Peng. A communication efficient admm-based distributed algorithm using two-dimensional torus grouping allreduce. *Data Science and Engineering*, 8(1):61–72, 2023.

[41] Xian-He Sun and Xiaoyang Lu. The memory-bounded speedup model and its impacts in computing. *Journal of Computer Science and Technology*, 38(1):64–79, 2023.

[42] Keshi Ge, Yiming Zhang, Yongquan Fu, Zhiquan Lai, Xiaoge Deng, and Dongsheng Li. Accelerate distributed deep learning with cluster-aware sketch quantization. *Science China Information Sciences*, 66(6):162102, 2023.

[43] Xupeng Miao, Xiaonan Nie, Hailin Zhang, Tong Zhao, and Bin Cui. Hetu: A highly efficient automatic parallel distributed deep learning system. *Science China. Information Sciences*, 66(1):117101, 2023.

[44] Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, and Bin Cui. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473*, 2024.

[45] Fanglue Zhang, Junhong Zhao, Yun Zhang, and Stefanie Zollmann. A survey on 360 images and videos in mixed reality: algorithms and applications. *Journal of Computer Science and Technology*, 38(3):473–491, 2023.

[46] Hailin Zhang, Penghao Zhao, Xupeng Miao, Yingxia Shao, Zirui Liu, Tong Yang, and Bin Cui. Experimental analysis of large-scale learnable vector storage compression. *VLDB*.

[47] Hailin Zhang, Zirui Liu, Boxuan Chen, Yikai Zhao, Tong Zhao, Tong Yang, and Bin Cui. Cafe: Towards compact, adaptive, and fast embedding for large-scale recommendation models. *Proceedings of the ACM on Management of Data*, 2(1):1–28, 2024.

[48] Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *ICLR*, 2021.

[49] Jeremy M Cohen, Behrooz Ghorbani, Shankar Krishnan, Naman Agarwal, Sourabh Medapati, Michal Badura, Daniel Suo, David Cardoze, Zachary Nado, George E Dahl, et al. Adaptive gradient methods at the edge of stability. *arXiv preprint arXiv:2207.14484*, 2022.

[50] Alex Damian, Eshaan Nichani, and Jason D Lee. Self-stabilization: The implicit bias of gradient descent at the edge of stability. *arXiv preprint arXiv:2209.15594*, 2022.

[51] Yang You, Zhao Zhang, Cho-Jui Hsieh, James Demmel, and Kurt Keutzer. Imagenet training in minutes. In *Proceedings of the 47th international conference on parallel processing*, pages 1–10, 2018.

[52] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training BERT in 76 minutes. In *ICLR*, 2020.

[53] Zhewei Yao, Amir Gholami, Sheng Shen, Mustafa Mustafa, Kurt Keutzer, and Michael Mahoney. Adahessian: An adaptive second order optimizer for machine learning. In *AAAI*, volume 35, pages 10665–10673, 2021.

[54] Jimmy Ba, Roger Grosse, and James Martens. Distributed second-order optimization using kronecker-factored approximations. In *Iclr*, 2022.

[55] Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. In *ICLR*, 2024.

[56] Tom Schaul, Sixin Zhang, and Yann LeCun. No more pesky learning rates. In *ICML*, pages 343–351. PMLR, 2013.

[57] Lukas Balles, Javier Romero, and Philipp Hennig. Coupling adaptive batch sizes with learning rates. In *UAI*, 2017.

[58] Samuel L. Smith and Quoc V. Le. A bayesian perspective on generalization and stochastic gradient descent. In *ICLR*, 2018.

[59] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *ICLR*, 2017.

[60] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[61] Umut Simsekli, Levent Sagun, and Mert Gurbuzbalaban. A tail-index analysis of stochastic gradient noise in deep neural networks. In *ICML*, pages 5827–5837, 2019.

[62] Christopher J Shallue, Jaehoon Lee, Joseph Antognini, Jascha Sohl-Dickstein, Roy Frostig, and George E Dahl. Measuring the effects of data parallelism on neural network training. *JMLR*, 20(112):1–49, 2019.

[63] 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.

A Parameter Update Amount in the Adam Optimizer

The update amount in the Adam optimizer consists of two parts, the first moment:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1)G_{est,t} \\ \widehat{m_t} &= \frac{m_t}{1-\beta_1^t} \end{aligned} \tag{25}$$

and the second moment:

$$\begin{aligned} v_t &= \beta_2 v_{t-1} + (1-\beta_2)G_{est,t}^2 \\ \widehat{v_t} &= \frac{v_t}{1-\beta_2^t} \end{aligned} \tag{26}$$

where $m_0 = 0$ and $v_0 = 0$. The final update amount is

$$V = \frac{\widehat{m_t}}{\sqrt{\widehat{v_t}} + \epsilon_{Adam}} = \frac{\frac{1-\beta_1}{1-\beta_1^t}\sum_i^t \beta_1^{t-i} G_{est,i}}{\sqrt{\frac{1-\beta_2}{1-\beta_2^t}\sum_i^t \beta_2^{t-i} G_{est,i}^2} + \epsilon_{Adam}} \tag{27}$$

After ignoring the role of ϵ_{Adam}, when $\beta_1 \to 1$ and $\beta_2 \to 1$, Eq 27 transforms to

$$V = \frac{\frac{\sum_i^t G_{est,i}}{t}}{\sqrt{\frac{\sum_i^t G_{est,i}^2}{t}}} = \frac{\mathbb{E}_t[G_{est}]}{\sqrt{\mathbb{E}_t[G_{est}^2]}} = \frac{sign(\mathbb{E}_t[G_{est}])}{\sqrt{1 + \frac{var_t(G_{est})}{\mathbb{E}_t[G_{est}]^2}}} \tag{28}$$

The equation is obtained by using $var_t(G_{est}) = \mathbb{E}_t[G_{est}^2] - \mathbb{E}_t[G_{est}]^2$. Note that the expected value $\mathbb{E}_t[G_{est}]$ is over the iteration distribution, not over the data distribution. Obviously, when the variance of G_{est} is small, the update amount is approximately $sign(G_{est})$.

On the other hand, when $\beta_1 \to 0$ and $\beta_2 \to 0$, Eq 27 can be simplified to

$$V = \frac{G_{est}}{\sqrt{G_{est}^2}} = sign(G_{est}) \tag{29}$$

Therefore, we can approximate the parameter update amount of the Adam optimizer as $sign(G_{est})$, without affecting the theoretical conclusion.

B Proof of Lemma 1

Proof. We can perturb the parameters θ by some vector V with learning rate ϵ, and approximate the true loss using a quadratic expansion in terms of ϵ via Taylor expansion:

$$L(\theta - \epsilon \cdot V) \approx L(\theta) - \epsilon G^T V + \frac{1}{2}\epsilon^2 V^T H V. \tag{30}$$

Consider the expected value of loss improvement over a data distribution $\rho(x)$ over data points x:

$$\mathbb{E}[\Delta L] = \mathbb{E}[L(\theta) - L(\theta - \epsilon \cdot V)] \approx \epsilon G^T \mathbb{E}[V] - \frac{1}{2}\epsilon^2 \mathbb{E}[V^T H V]. \tag{31}$$

By maximizing the expected loss improvement, we obtain the optimal learning rate in Eq 6 and the optimal loss improvement in Eq 7. □

C Proof of Theorem 2

Proof. Consider a random variable $x \sim \mathcal{N}(\mu, \sigma^2)$, let $y = \frac{x-\mu}{\sigma}$ and $z = -\frac{\mu}{\sigma}$, then

$$\begin{aligned} \mathbb{E}[sign(x)] &= \int_{-\infty}^{\infty} sign(x)\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx \\ &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} sign(\mu + \sigma y) e^{-\frac{y^2}{2}} dy \\ &= \frac{1}{\sqrt{2\pi}}(\int_z^{\infty} e^{-\frac{y^2}{2}} dy - \int_{-\infty}^{z} e^{-\frac{y^2}{2}} dy) \\ &= (1 - \Phi(z)) - \Phi(z) = erf\left(\frac{\mu}{\sqrt{2}\sigma}\right) \end{aligned} \tag{32}$$

and the variance is

$$var(sign(x)) = \mathbb{E}[sign(x)^2] - \mathbb{E}[sign(x)]^2 = 1 - erf\left(\frac{\mu}{\sqrt{2}\sigma}\right)^2, \tag{33}$$

where Φ represents the cumulative distribution function of the standard normal distribution, and erf is the Gauss error function that is defined as $erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$.

Given that the gradient G_x of any data point x in the data distribution $\rho(x)$ relative to a certain parameter θ_i follows a Gaussian distribution with mean μ_i and variance σ_i^2, then

$$G_{est}(\theta_i) = \frac{1}{B}\sum^{B} G_x(\theta_i) \sim \mathcal{N}\left(\mu_i, \frac{\sigma_i^2}{B}\right). \tag{34}$$

Therefore, when $V = sign(G_{est})$, using Eq 32 and Eq 33 we can get the expectation

$$\mathbb{E}[V] = \begin{pmatrix} \vdots \\ erf\left(\sqrt{\frac{B}{2}}\frac{\mu_i}{\sigma_i}\right) \\ \vdots \end{pmatrix}, \tag{35}$$

and the covariance matrix

$$cov(V) = \begin{pmatrix} \ddots & & 0 \\ & 1 - erf\left(\sqrt{\frac{B}{2}}\frac{\mu_i}{\sigma_i}\right)^2 & \\ 0 & & \ddots \end{pmatrix}. \tag{36}$$

Given that the real gradient satisfies:

$$G = \begin{pmatrix} \vdots \\ \mu_i \\ \vdots \end{pmatrix}, \tag{37}$$

and define $\mathcal{E}_i$ as a function with respect to the token batch size B, based on the Gauss error function:

$$\mathcal{E}_i(B) = erf\left(\sqrt{\frac{B}{2}}\frac{\mu_i}{\sigma_i}\right) = \frac{2}{\sqrt{\pi}}\int_0^{\sqrt{\frac{B}{2}}\frac{\mu_i}{\sigma_i}} e^{-t^2}dt, \tag{38}$$

substituting the expectation and the variance of V in Eq 6 and Eq 7 from Lemma 1 using the above equations, we can get Eq 8 and Eq 9 respectively.

To simplify the subsequent computation, we approximate the function $\mathcal{E}_i$ by other sigmoid-like analytical forms. Specifically, we find that $\mathcal{E}_i(B) \approx \frac{\frac{\mu_i}{\sigma_i}}{\sqrt{\frac{\pi}{2B} + \left(\frac{\mu_i}{\sigma_i}\right)^2}}$, which results in Eq 10.

□

D Proof of Theorem 3

Proof. When $B \ll \frac{\pi\sigma_i^2}{2\mu_i^2}$, Eq 10 reduces to:

$$\mathcal{E}_i(B) \approx \sqrt{\frac{2B}{\pi}}\frac{\mu_i}{\sigma_i}. \tag{39}$$

Substituting the function $\mathcal{E}_i$ in Eq 9 using the above equation, and defining $\mathcal{B}_{noise}$ and ϵ_{max} as in Eq 14 and 16, we can deduce the relationship between the optimal learning rate and the token batch size:

$$\begin{aligned}\epsilon_{opt}(B) &= \frac{\sum_i \sqrt{\frac{2B}{\pi}} \frac{\mu_i}{\sigma_i}\mu_i}{\sum_i H_{i,i} + \frac{2B}{\pi}\sum_i\sum_j \begin{cases} \frac{\mu_i\mu_j}{\sigma_i\sigma_j} & i\neq j \\ 0 & i=j\end{cases} H_{i,j}} \\ &= \frac{\frac{\sum_i \sqrt{\frac{2B}{\pi}}\frac{\mu_i^2}{\sigma_i}}{\sum_i H_{i,i}}}{1 + B\frac{2\sum_i\sum_j \begin{cases} \frac{\mu_i\mu_j}{\sigma_i\sigma_j} & i\neq j \\ 0 & i=j\end{cases} H_{i,j}}{\pi\sum_i H_{i,i}}} \\ &= \frac{\sqrt{B}}{1+\frac{B}{\mathcal{B}_{noise}}}\frac{\sqrt{\frac{2}{\pi}}\sum_i \frac{\mu_i^2}{\sigma_i}}{\sum_i H_{i,i}} \\ &= \frac{\frac{\sqrt{\mathcal{B}_{noise}}}{2}}{\frac{1}{2}\left(\sqrt{\frac{\mathcal{B}_{noise}}{B}}+\sqrt{\frac{B}{\mathcal{B}_{noise}}}\right)}\frac{\sqrt{\frac{2}{\pi}}\sum_i \frac{\mu_i^2}{\sigma_i}}{\sum_i H_{i,i}} \\ &= \frac{\epsilon_{max}}{\frac{1}{2}\left(\sqrt{\frac{\mathcal{B}_{noise}}{B}}+\sqrt{\frac{B}{\mathcal{B}_{noise}}}\right)} \leq \epsilon_{max}.\end{aligned} \tag{40}$$

It should be noted that since $\mathcal{B}_{noise} > 0$ and $\epsilon_{max} > 0$, then both $\sum_i H_{i,i}$ and $\sum_i\sum_j \begin{cases} \frac{\mu_i\mu_j}{\sigma_i\sigma_j} & i\neq j \\ 0 & i=j\end{cases} H_{i,j}$ are greater than 0. Therefore, ϵ_{max} can be expressed as a form that does not contain $\mathcal{B}_{noise}$:

$$\begin{aligned}\epsilon_{max} &= \frac{\sqrt{\frac{\mathcal{B}_{noise}}{2\pi}}\sum_i \frac{\mu_i^2}{\sigma_i}}{\sum_i H_{i,i}} \\ &= \frac{\sqrt{\sum_i H_{i,i}}}{2\sqrt{\sum_i\sum_j \begin{cases} \frac{\mu_i\mu_j}{\sigma_i\sigma_j} & i\neq j \\ 0 & i=j\end{cases} H_{i,j}}}\frac{\sum_i \frac{\mu_i^2}{\sigma_i}}{\sum_i H_{i,i}} \\ &= \frac{\sum_i \frac{\mu_i^2}{\sigma_i}}{2\sqrt{\sum_i\sum_j \begin{cases} \frac{\mu_i\mu_j}{\sigma_i\sigma_j} & i\neq j \\ 0 & i=j\end{cases} H_{i,j}} \cdot \sqrt{\sum_i H_{i,i}}}. \\ &\geq \frac{\sum_i \frac{\mu_i^2}{\sigma_i}}{\sum_i\sum_j \begin{cases} \frac{\mu_i\mu_j}{\sigma_i\sigma_j} & i\neq j \\ 1 & i=j\end{cases} H_{i,j}}\end{aligned} \tag{41}$$

The last inequality is derived using the AM–GM inequality ($a^2 + b^2 \geq 2ab$).

□

E Proof of Theorem 4

Proof. When $B \gg \frac{\pi\sigma_i^2}{2\mu_i^2}$, Eq 10 converges to:

$$\mathcal{E}_i = sign\left(\frac{\mu_i}{\sigma_i}\right) = sign(\mu_i). \tag{42}$$

Substituting the function $\mathcal{E}_i$ in Eq 9, we can obtain Eq 17. □

F Proof of Theorem 5

Proof. When $B \ll \frac{\pi\sigma_i^2}{2\mu_i^2}$, we have the approximate results in Eq 39 from Theorem 3. Substituting $\mathcal{E}_i$ and $\mathcal{B}_{noise}$ using Eq 39 and 14, and defining ΔL_{max} as in 19, the optimal loss improvement in Eq 8 can be expressed as:

$$\begin{aligned}
\Delta L_{opt}(B) &= \frac{\cancel{\frac{1}{2}} \cdot \frac{\cancel{2}B}{\pi} \sum_i \sum_j \frac{\mu_i^2\mu_j^2}{\sigma_i\sigma_j}}{\sum_i H_{i,i} + \frac{2B}{\pi} \sum_i \sum_j \begin{cases} \frac{\mu_i\mu_j}{\sigma_i\sigma_j} & i \neq j \\ 0 & i = j \end{cases} H_{i,j}} \\
&= \frac{\sum_i \sum_j \frac{\mu_i^2\mu_j^2}{\sigma_i\sigma_j}}{\frac{\pi \sum_i H_{i,i}}{B} + 2 \sum_i \sum_j \begin{cases} \frac{\mu_i\mu_j}{\sigma_i\sigma_j} & i \neq j \\ 0 & i = j \end{cases} H_{i,j}} \\
&= \frac{1}{\frac{\mathcal{B}_{noise}}{B} + 1} \frac{\sum_i \sum_j \frac{\mu_i^2\mu_j^2}{\sigma_i\sigma_j}}{2 \sum_i \sum_j \begin{cases} \frac{\mu_i\mu_j}{\sigma_i\sigma_j} & i \neq j \\ 0 & i = j \end{cases} H_{i,j}} \\
&= \frac{\Delta L_{max}}{\frac{\mathcal{B}_{noise}}{B} + 1} \leq \Delta L_{max}
\end{aligned} \tag{43}$$

Following the Appendix D in [24], this allows both the total number of steps and data examples processed to still be written as

$$\begin{aligned}
S &= \int \left(1 + \frac{\mathcal{B}_{noise}}{B}\right) ds \\
E &= \int (\mathcal{B}_{noise} + B) ds
\end{aligned} \tag{44}$$

Therefore the conclusion in [24]

$$\begin{aligned}
S_{min} &= \int ds \\
E_{min} &= \int \mathcal{B}_{noise} ds
\end{aligned} \tag{45}$$

and Eq 20, 21 still hold.

□

G Variable Estimation for Data/Time Efficiency Relationship Equation

When considering S, E, S_{min} and $E_{min} > 0$, Eq 20 can be simplified to

$$\begin{aligned}
\left(\frac{S}{S_{min}} - 1\right)\left(\frac{E}{E_{min}} - 1\right) &= 1 \\
SE - S_{min}E - SE_{min} + \cancel{S_{min}E_{min}} &= \cancel{S_{min}E_{min}} \\
S_{min}E + SE_{min} &= SE \\
\frac{S_{min}}{S} + \frac{E_{min}}{E} &= 1 \\
\frac{1}{S} &= -\frac{E_{min}}{S_{min}}\frac{1}{E} + \frac{1}{S_{min}}
\end{aligned} \tag{46}$$

so a linear fit to the relationship between $\frac{1}{S}$ and $\frac{1}{E}$ can estimate S_{min} and E_{min}.

H Additional Experiments Results

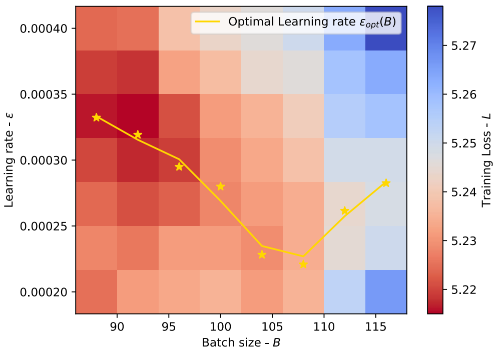
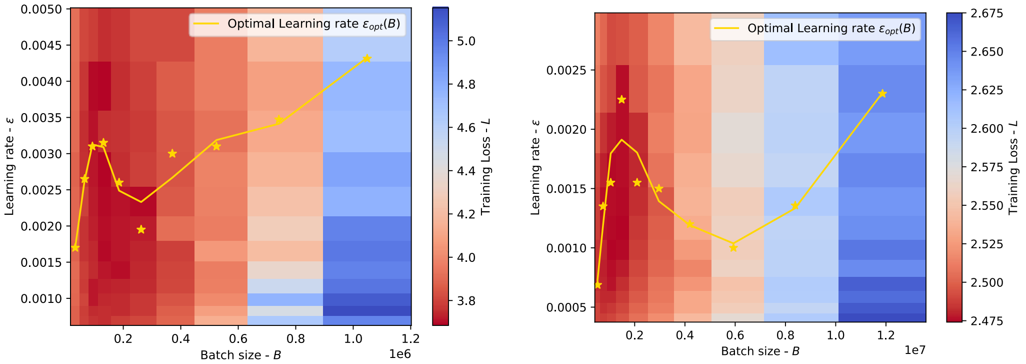


Figure 6: Finer-grained grid search results for the experiments shown in Figure 4(b).



(a) 1e17 FLOPs (177M FLOPs/token)

(b) 1e20 FLOPs (2.94B FLOPs/token)

Figure 7: The optimal learning rates, based on the results presented in the Deekseek paper [63], align with our theorems.

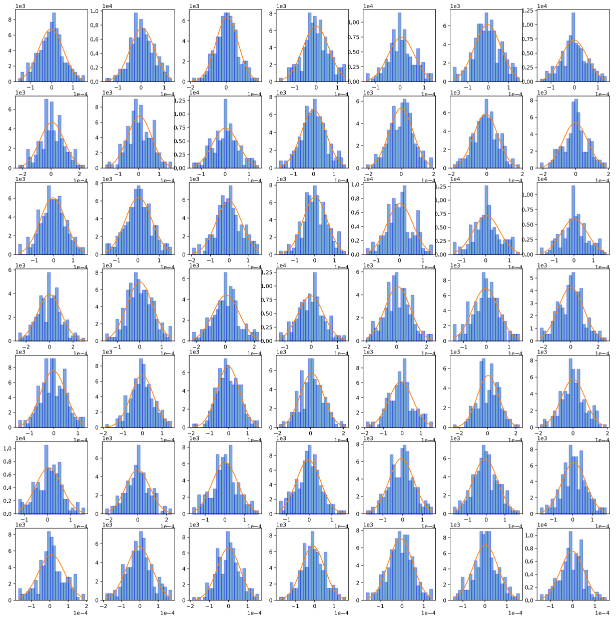


Figure 8: Examples of gradient distributions observed during the training of an MoE structure model, which approximate Gaussian distributions.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The claims provide a coherent and consistent overview of the research objectives, methodologies, and findings.

   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]

   Justification: See Section 4.

   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: See Section 2 and Appendix A, B, C, D, E, F.

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: See Section 3.1.

   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: [NA]

Justification: The model structure and data required for the experiments are publicly available. And the necessary experimental configuration has been provided in Section 3.1 for reproduction.

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).
- 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: The necessary experimental configurations have been provided in Section 3.1 for reproduction.

   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: [Yes]

   Justification: See Section 3.

   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: See 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 conforms 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 of the work performed. This work is a foundational research and is not tied to particular applications.

    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: The paper poses no such risks.

    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: We cite the creators and provide necessary information in Section 3.1.

    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: [NA]

    Justification: The paper does not release new assets.

    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: The 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: The 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.