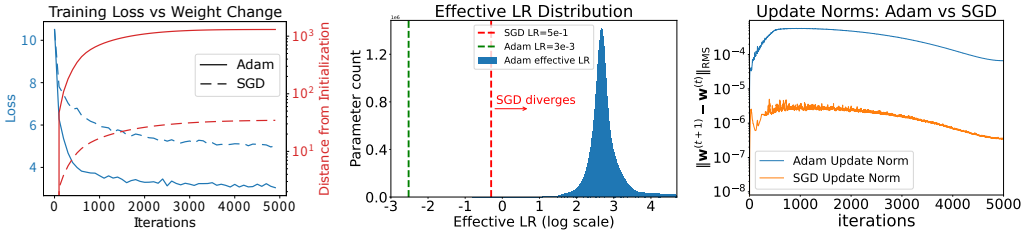
Revisiting the Adam-SGD Gap in LLM Pre-Training: The Role of Large Effective Learning Rates

Athanasios Glentis **Dawei Li** **Chung-Yiu Yau** **Mingyi Hong**

University of Minnesota
{glent007,li004678,cyau,mhong}@umn.edu

Abstract

It is widely believed that stochastic gradient descent (SGD) performs significantly worse than adaptive optimizers such as Adam in pre-training Large Language Models (LLMs). Yet the underlying reason for this gap remains unclear. In this work, we attribute a large part of the discrepancy to SGD's inability to sustain learning rates comparable to Adam's much larger effective learning rates. Through empirical and theoretical analysis of LLM pre-training dynamics, we identify that training is characterized by small gradient norms and large weight-to-gradient ratios, an effect that becomes more pronounced with larger batch sizes typical in pre-training, necessitating such large effective learning rates. However, we find that output-layer gradient magnitudes become highly uneven across token classes, and that large gradient spikes frequently occur during training. Together, these effects severely restrict the admissible learning rate of SGD. Guided by this understanding, we show that simple clipping mechanisms that stabilize SGD at large learning rates enable it to recover most of Adam's performance. In our large-scale experiments, the validation loss gap between large-learning-rate SGD and Adam shrinks from more than 50% to only about 2.3% when pre-training a 7B-parameter LLaMA model with a 1M-token batch size. Code is available at this link.



(a) (b) (c)

Figure 1: (a) The pre-training dynamics of Adam and (momentum) SGD: (Left vertical axis) Training loss. (Right vertical axis) Distance from initialization $\|\mathbf{w}^{(t)} - \mathbf{w}^{(0)}\|_F$. The effective solution space of SGD is about $50\times$ smaller than that of Adam. (b) Effective learning rates (LR) $a_i^{(t)}$ at iteration $t = 500$. For SGD $a_i^{(t)} \equiv \eta_t$ for all parameters; for Adam $a_i^{(t)} = \eta_t/(\sqrt{v_i^{(t)}} + \varepsilon)$ for each $w_i^{(t)}$ (see Section 3.1 for details). The effective LR of Adam are 2-4 orders of magnitude larger than SGD's. Base LR tuned to 3e-3 for Adam and 0.5 for SGD (dashed lines). Increasing SGD's LR further leads to divergence. (c) The update norm of Adam stays about 2 orders of magnitude larger than SGD's during training. All figures are from training LLaMA 130M on C4 with a 0.5M token batch size.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Adaptive optimization algorithms such as Adam [Kingma and Ba, 2015] have been empowering the optimization of large-scale LLM pre-training tasks. In contrast, the classical baseline algorithm (momentum) SGD [Robbins and Monro, 1951] has received significantly less attention in the literature of pre-training. For instance, SGD is no longer seen as a baseline in recent benchmarks of optimizers for pre-training [Wen et al., 2025, Semenov et al., 2025]. This points to the common belief that SGD is not suitable for the task of LLM pre-training.

Notably, the reported performance gap between SGD and Adam in standard pre-training settings is exceptionally large and consistently observed across studies [Pan and Li, 2023, Kunstner et al., 2024, Zhang et al., 2024, Marek et al., 2025]. The same gap is also reproduced in our Figure 1 (a), where SGD results in over 60% final loss increase relative to Adam. This indicates that SGD plateaus quickly in a high loss region, failing to drive the loss to acceptable levels.

Taking this gap as given, prior works have focused considerable effort on explaining why SGD is ineffective for pre-training. Among others, Kunstner et al. [2024] studied the heavy-tailed class imbalanced data distribution in pre-training, observing that SGD struggles to progress on infrequent targets. Pan and Li [2023] analyzed the empirical local geometry of the pre-training loss and suggested that SGD has ill-conditioned sharpness. Zhang et al. [2024] observed Hessian block heterogeneity to be coexisting with the poor performance of SGD.

While these prior results are highly important, providing valuable insights into specific aspects of SGD in pre-training, the broader training dynamics and their relation to SGD's poor performance remain significantly less explored. Thus, in this work, rather than focusing on isolated behaviors of SGD, we examine closely its broader, high-level "symptoms" tied to pre-training dynamics that lead to the considerable gap with Adam.

In particular, we begin by observing that SGD, even after extensive learning rate tuning, only explores a small neighborhood around initialization, considerably smaller than Adam (as shown in Figure 1 (a)). This phenomenon is not immediately apparent, as Adam typically uses a much smaller (base) learning rate than SGD. This motivates us to examine their *effective* learning rates, i.e., the scaling each method applies to the momentum of the gradient, where we observe a striking difference: the effective learning rates of Adam are orders of magnitude larger than the learning rate of SGD (Figure 1 (b)). Consequently, the overall update strength of the two methods differs substantially (Figure 1 (c)), suggesting that SGD's smaller updates are insufficient to keep pace. We further argue that pre-training dynamics, characterized by small gradient norms and large weight-to-gradient ratios, necessitate such large effective learning rates, which we support theoretically and relate to the typical large batch size used in pre-training.

To further investigate this gap, we next ask *why* the learning rate in SGD cannot simply be increased further. We find that, at larger learning rates gradient irregularities become a key limiting factor for SGD due to its unconstrained update. Motivated by this observation, we validate our conjecture by "treating" these irregularities via simple, yet targeted gradient clipping, which enables us to substantially increase the SGD learning rate and close most of the gap to Adam.

We emphasize that our perspective is fresh and complementary to prior work: while existing studies explain specific mechanisms underlying SGD's behavior, our focus is on high-level training dynamics that account for a large part of the observed discrepancy, leaving finer-grained factors to explain the remaining gap.

We summarize the main contributions of our work as follows:

- We empirically study the effective learning rates of Adam compared to that of SGD in the setting of LLM pre-training. Even after tuning both algorithms' base learning rates, we find a substantial discrepancy, i.e., Adam's effective learning rates are 2-4 orders of magnitude larger than SGD's. We further observe the consequence of that, being insufficient update norm for SGD and thus limited exploration of the solution space (see Figure 1 for details).
- We identify that the need for high learning rates in LLM pre-training arises from the training dynamics, in particular the substantially small gradient norms observed. This results in a higher weight-to-stochastic-gradient (weight-to-SG) norm ratio than that observed in other training setups, e.g., convolutional network training [You et al., 2017]. We further observe that this effect intensifies with increasing the batch size.

- Theoretically, we provide an explanation for the small stochastic gradient norms observed in LLMs. In particular, by decomposing the per-token-class gradients under two complementary regimes (i.e., bias-variance decomposition (8) and background-hit decomposition (9)) we derive a universal upper bound for the norm of the stochastic gradient (see Theorem 3.2). We further analyze the terms in the upper bound, showing that under assumptions in the training dynamics and token distribution, this bound decreases with batch size, consistent with our empirical observations.
- We identify two types of gradient irregularities in pre-training that contribute to SGD's divergence at high learning rates. First, we observe frequent layer-wise gradient norm spikes especially at the early stage of training. Second, at the output layer of size $d \times |V|$ where $|V|$ is the number of the output tokens, we observe that token-class SG norms $\|\mathbf{g}_1\|_2, ..., \|\mathbf{g}_V\|_2$ are highly imbalanced.
- Finally, we demonstrate that enabling larger learning rates in SGD can close most of the observed performance gap to Adam in pre-training. To achieve this we use two simple, yet targeted, types of gradient clipping to "treat" the aforementioned gradient irregularities. To our knowledge, this is the first time SGD has been successfully used to pre-train LLMs with such large learning rates (100–300 in our experiments), substantially narrowing the gap to Adam. Specifically, in our largest experiment pre-training LLaMA 7B on C4 with SGD having learning rate 300 and a 1M-token batch size, the validation loss gap to Adam closes from over 50% to only about 2.3%.

2 Preliminaries and Related Works

The **LLM pre-training objective** can be expressed as:

$$\min_{\mathbf{w}} \ell(\mathbf{w}) := \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}; \xi_i) \tag{1}$$

where ℓ is the loss function typically defined as the cross entropy loss of predicting the next token, $\mathbf{w}$ are the trainable parameters of the model and ξ_i, $i = 1, ..., n$ are our training input sequences.

To solve (1), **SGD**, at every training iteration t, draws a batch of samples $\{\xi_{t,b}\}_{b=1,..,B}$ and performs an update toward the negative stochastic gradient direction:

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta_t \mathbf{g}^{(t)}, \quad \text{with} \quad \mathbf{g}^{(t)} := \frac{1}{B} \sum_{b=1}^{B} \nabla \ell(\mathbf{w}^{(t)}; \xi_{t,b}) \tag{2}$$

Here B is batch size and η_t is the learning rate at iteration t. A straightforward extension is momentum SGD (which we will simply refer to as SGD throughout), replacing the gradient with its momentum to smooth the updates:

$$\mathbf{m}^{(t)} = \beta \mathbf{m}^{(t-1)} + (1-\beta)\mathbf{g}^{(t)} \quad \mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta_t \mathbf{m}^{(t)}. \tag{3}$$

In contrast, **Adam** [Kingma and Ba, 2015] updates the parameters using a more sophisticated scheme as shown below ($\odot$ is element-wise product; bias correction term omitted for simplicity):

$$\begin{aligned} &\mathbf{m}^{(t)} = \beta_1 \mathbf{m}^{(t-1)} + (1-\beta_1)\mathbf{g}^{(t)}, \\ &\mathbf{v}^{(t)} = \beta_2 \mathbf{v}^{(t-1)} + (1-\beta_2)\mathbf{g}^{(t)} \odot \mathbf{g}^{(t)} \end{aligned} \quad \mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta_t \frac{\mathbf{m}^{(t)}}{\sqrt{\mathbf{v}^{(t)}} + \epsilon}. \tag{4}$$

Note that, unless explicitly stated otherwise, we use the standard hyperparameters $\beta = 0.9$ for SGD and $(\beta_1 = 0.9, \beta_2 = 0.95)$ for Adam in all our experiments.

Related works. It was commonly observed that SGD is far behind adaptive optimizers such as Adam on typical LLM pre-training problems [Pan and Li, 2023, Kunstner et al., 2024, Zhang et al., 2024, Marek et al., 2025, Srećković et al., 2025], while such large gap is not typically observed in other training settings [Keskar and Socher, 2017, Zhang et al., 2020].

One line of works tries to attribute such gap to the text data distribution. Notably, Zhang et al. [2020] first argued that heavy-tailed stochastic gradient noise in Wikipedia corpus BERT pre-training causes SGD non-convergence, while well-concentrated stochastic gradient noise in Imagenet Resnet training does not. Their solution depends on an adaptive step size that is at the same complexity as Adam. In a related work, Kunstner et al. [2024] argues that LLMs are pre-trained on class-imbalanced data

distributions, implying that heavy-tail rare tokens contribute to heavy-tail large stochastic gradient noise and SGD fails to capture the rare tokens' gradient that deviates from the momentum.

Another line of works focuses on the geometry of LLM optimization. Pan and Li [2023] suggests that SGD has high directional sharpness as compared to Adam, which results to slow convergence. Meanwhile, Zhang et al. [2024] proposed that block heterogeneous Hessian prevents SGD from achieving good performance.

More recently, Srećković et al. [2025] point out that the performance gap between SGD and Adam narrows in the very small batch size setting, debunking the prior explanations on this performance gap. Similarly, Marek et al. [2025] suggests that training with very small batch sizes (as low as batch size 1) displays minimal sensitivity to other training hyperparameters and choice of optimizer. Our insights will further support those previous works, since we will show that the pre-training phenomena which hinder SGD's performance become worse as batch size increases.

Our approach. The main focus of this work is to study the high-level pre-training dynamics, including effective learning rates and weight to gradient norm ratios, combining empirical and theoretical insights. This analysis will suggest that SGD needs much larger base learning rate to make effective training progress, which can be enabled by "treating" gradient irregularities and close *most* of the gap with Adam. Thus, we complement prior works which focus on specific limitations of SGD in pre-training that combined may explain the remaining portion of the gap.

3 LLM Training Dynamics and the Need for Large Learning Rate

In this section, we explore the LLM training dynamics. In particular, we observe empirically that SGD has a much smaller "effective" learning rate compared to Adam. Then, both empirically and analytically, we provide justifications for the necessity for a large "effective" learning rate in LLM pre-training. The empirical observations here are from pre-training a LLaMA 130M model [Touvron et al., 2023a] on C4 [Raffel et al., 2020] with token batch size 0.5M (see Appendix C for more details).

3.1 Effective Momentum Learning Rate

To begin with, we justify the difference in the magnitude of update between different optimizers. To quantify the magnitude of update, we focus on the case of stochastic momentum algorithms and define **effective momentum learning rate** as follows:

Definition 3.1. For stochastic momentum optimization algorithms under the following update rule for model parameters $\mathbf{w}^{(t)} = [w_1^{(t)}, ..., w_d^{(t)}] \in \mathbb{R}^d$ with arbitrary initialization $\mathbf{w}^{(0)}$, $\mathbf{a}^{(t)} = [a_1^{(t)}, ..., a_d^{(t)}] \geq \mathbf{0}$, $\beta \in [0, 1)$, $\mathbf{m}^{(0)} = \nabla f(\mathbf{w}^{(0)}; \xi^{(0)})$:

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \mathbf{a}^{(t)} \odot \mathbf{m}^{(t)}, \tag{5a}$$

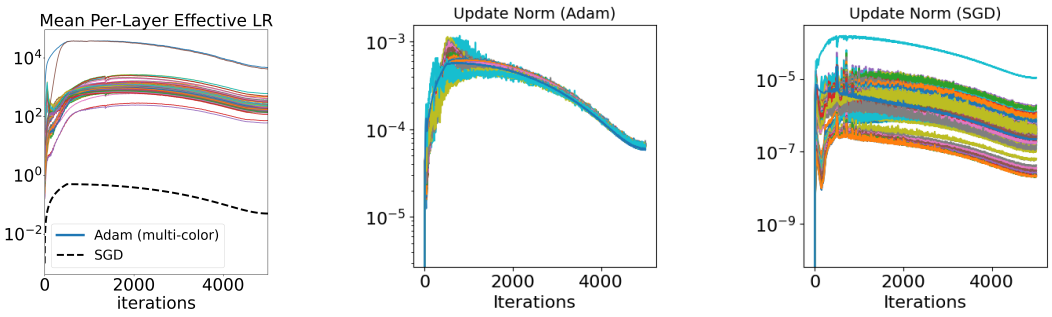
$$\mathbf{m}^{(t+1)} = \beta \mathbf{m}^{(t)} + (1 - \beta) \nabla f \left(\mathbf{w}^{(t+1)}; \xi^{(t+1)} \right), \tag{5b}$$

we define $a_i^{(t)}$ as the **effective momentum learning rate** for parameter $w_i^{(t)}$ (also referred as *effective learning rate* for simplicity).

By Definition 3.1, the effective learning rate of SGD is simply the learning rate $a_i^{(t)} \equiv \eta_t$ for all parameters $i \in \{1, ..., d\}$. On the other hand, by the Adam update (4), the effective learning rate of Adam for $w_i^{(t)}$ is $a_i^{(t)} = \eta_t / (\sqrt{v_i^{(t)}} + \varepsilon)$. Note that η_t follows the cosine learning rate schedule with warm-up and the base learning rate η (i.e., the peak value of η_t) is tuned for both optimizers.

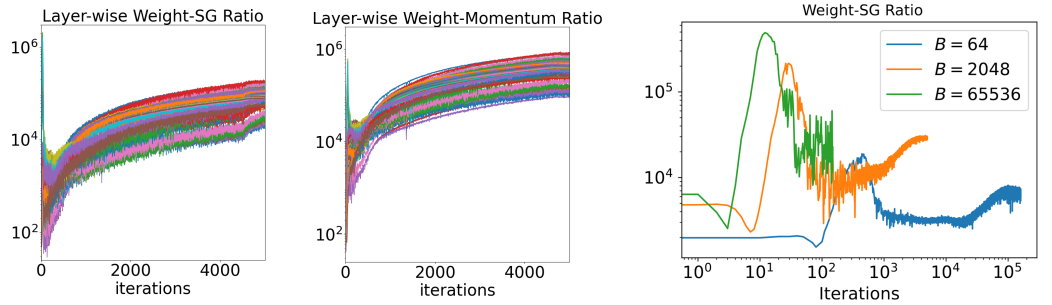
We find that despite the small learning rate typically used for Adam (tuned to 3×10^{-3} in our experiment), $a_i^{(t)}$ is significantly large on average as shown in Figure 2 (a), ranging from around 10^2 to 10^4 throughout most of the training. In comparison, the same quantity for SGD peaks only to 0.5, while larger learning rates result to training divergence (similar to prior observations such as Zhang et al. [2024]).

The small effective learning rate of SGD indicates that the update norm of SGD during training is also much smaller than Adam's. We observe such phenomenon as well in Figures 2 (b-c). Such small update norm suggests that SGD has limited exploration behavior, as shown in Figure 1 (a), restricting its ability to keep pace with Adam.



(a) (b) (c)

Figure 2: (a) Mean effective learning rate per layer for Adam and SGD. We use cosine scheduler with warmup and base learning rate tuned to 3e-3 for Adam and 0.5 for SGD. The Adam effective learning rates are 2-4 orders of magnitude larger than SGD's. (b-c) The dynamics of layer-wise update norm $\|\mathbf{w}_\ell^{(t+1)} - \mathbf{w}_\ell^{(t)}\|_{\mathrm{RMS}}$ during training. The update norm of SGD is over $100\times$ smaller than that of Adam. We use different colors to represent the averages for different layers.



(a) (b) (c)

Figure 3: (a) The dynamics of layer-wise weight to stochastic gradient (SG) norm ratio $\|\mathbf{w}^{(t)}\|_F / \|\nabla f(\mathbf{w}^{(t)}; \xi^{(t)})\|_F$ during training. (b) Layer-wise weight to momentum (of the stochastic gradient) norm ratio during training. As momentum is a smoothed version of the gradient (thus typically having smaller norm), this ratio tends to be even larger. Ratios of different layers are represented by different colors. (c) The weight-SG ratio under different batch-size settings. As batch size increases, the weight-SG ratio increases.

3.2 Weight-gradient Norm Ratio

Given the aforementioned observation that Adam has a much larger effective learning rate than SGD in LLMs, a natural question is: whether such a large effective learning rate is necessary for obtaining good performances. In this section, we proceed to answer this question by observing the dynamics of model weight parameters and stochastic gradient (SG) via the weight-SG norm ratio $\|\mathbf{w}\|_F / \|\nabla f(\mathbf{w}; \xi)\|_F$.

Intuitively, as the weight-SG norm ratio gets larger, the learning rate needs to also scale up to ensure sufficiently large update for the weights. Note also that, a large weight-SG norm ratio implies a large (or often even larger) weight-momentum norm ratio, as the momentum is simply a smoothed version of the SG. In earlier works on large batch training of convolutional neural network on ImageNet, this ratio is at the order of 10^0 to 10^3 [You et al., 2017]. Meanwhile in the setting of LLM pre-training, our measured weight-SG norm ratio is significantly larger, ranging from 10^1 to 10^4 at initialization, and 10^3 to 10^5 during training; see Figure 3.

To better understand the reason why the weight-SG norm ratio becomes so large during LLM pre-training, we conduct a more detailed theoretical analysis of the gradient matrix. In particular, we show that the gradient norm remains small when the batch size is large under LLM-specific training dynamics and token distribution. In the meantime, the weight norm is typically lower bounded (see Figure 12) in Appendix E. Therefore, the weight-SG ratio becomes large.

Let $\mathcal{V} = [V] := \{1, ..., V\}$ denote the vocabulary, and $\mathcal{D}$ denotes the distribution of the input-output data pair (x, y). For some weight parameters W, we consider decomposing its mini-batch gradient G into class-wise contributions G_j such that

$$G_j := \frac{1}{B}\sum_{b=1}^{B}\big(p_j(\xi_b) - \mathbf{1}\{y_b = j\}\big)U_j(\xi_b), \qquad j \in [V], \tag{6}$$

where (ξ_b, y_b) are i.i.d. samples of forward and output information associated with the parameter block W and data $(x_b, y_b) \sim \mathcal{D}$, and $p_j(\xi)$ is the model-predicted probability of class j given input ξ. $U_j(\xi)$ is the class-wise gradient feature associated with the weight parameter, which can be calculated in closed form as shown later. Further, define the marginal frequency of token j and the corresponding conditional frequency on the input ξ as

$$q_j := \mathbb{P}_{(\xi,y)\sim\mathcal{D}}(y = j), \qquad q_j(\xi) := \mathbb{P}_{(\cdot,y)\sim\mathcal{D}}(y = j \mid \xi).$$

Let $\mathcal{H}$ and $\mathcal{T}$ be a partition of $[V]$. Then the full gradient norm admits the following upper bound.

Theorem 3.2. *Let G be the mini-batch stochastic gradient of a parameter block satisfying* (6)*. Let two sets $\mathcal{H}$ and $\mathcal{T}$ be a partition of the tokens $\mathcal{V} := \{1, ..., V\}$ such that $\mathcal{H} \cap \mathcal{T} = \varnothing$ and $\mathcal{H} \cup \mathcal{T} = \mathcal{V}$. Define $c_q := \max_{j\in\mathcal{T}} q_j$. Further, denote*

$$\varepsilon_j^2 := \mathbb{E}\big[\|(p_j(\xi) - q_j(\xi))U_j(\xi)\|^2\big], \qquad s_j^2 := \mathbb{E}\big[\|U_j(\xi)\|^2 \mid y = j\big].$$

Assume that the full gradient G satisfies $\|G\|^2 \le \kappa \sum_{j=1}^{V} \|G_j\|^2$ for some structural constant $\kappa \ge 1$. Then the following holds:

$$\begin{aligned}\frac{1}{\kappa}\mathbb{E}\|G\|_F^2 \le \left(1 + \frac{1}{B}\right)\sum_{j\in\mathcal{H}}\varepsilon_j^2 + \frac{1}{B}\sum_{j\in\mathcal{H}}\mathbb{E}\big[q_j(\xi)(1 - q_j(\xi))\|U_j(\xi)\|^2\big] \\ + \left(\frac{2}{B} + 2c_q\right)\sum_{j\in\mathcal{T}} q_j s_j^2 + 2\sum_{j\in\mathcal{T}}\mathbb{E}\big[p_j(\xi)^2\|U_j(\xi)\|^2\big].\end{aligned} \tag{7}$$

Theorem 3.2 provides a unified upper bound for the gradient norm that holds for arbitrary choice of the partition and any degree of token heterogeneity. The bound separates the gradient norm $\|G\|_F$ into the following four contributing terms: *(i)* The class-wise population gradient contribution for head tokens, controlled by ε_j; *(ii)* The covariance contribution for head tokens, which scales as $O(1/B)$. *(iii)* The softmax background of tail tokens, controlled by $p_j(\xi)$. *(iv)* The label-hit contribution of tail tokens, controlled by q_j (and uniformly by c_q). We note that κ is found to be small in the pre-training setting; see Appendix F for details.

The main point of Theorem 3.2 is to provide a mechanism for the large weight-SG norm ratio observed previously. In particular, a large batch size B suppresses the covariance term, while a suitable partition can separate head tokens, for which ε_j is small, from tail tokens, for which $p_j(\xi)$ and c_q are sufficiently small. Thus under such partition, the right-hand side of (7) can be small, leading to a small $\|G\|_F$ and hence a large weight-SG norm ratio. We next explain why these conditions hold in LLM training:

• ε_j is small for high-frequency tokens. High-frequency tokens are typically well-trained and exhibit lower per-token loss than low-frequency tokens [Chung and Kim, 2025]. This suggests that $|p_j(\xi) - q_j(\xi)|$ is small, leading to a small ε_j.

• B is large. In large-scale LLM training, the effective batch size is measured by the number of processed tokens per update, which is often on the order of 10^5–10^7.

• $p_j(\xi)$ is small for low-frequency tokens. Due to limited exposure during training, rare tokens are more likely to remain under-trained [Bao et al., 2023]. Therefore, their per-token loss (approximately $-\log p_j(\xi)$) can be high, indicating a smaller value of $p_j(\xi)$.

• c_q is small for low-frequency tokens. Recall that $c_q := \max_{j\in\mathcal{T}} q_j$ is the maximal frequency for all tail tokens. It is known that tail tokens have very small marginal probabilities in natural language corpora [Piantadosi, 2014].

More discussions about these terms can be found in Appendix E.

Theorem 3.2 gives a unified bound for all weight parameter blocks whose gradients admit class-wise decomposition (6). For specific layers, this bound can be more concrete by identifying ξ and $U_j(\xi)$. We illustrate this specialization in two representative cases, each with a different choice of κ.

(i) Output layer. For the output-layer weight matrix $W \in \mathbb{R}^{V\times d}$, take $\xi = h$ and $U_j(\xi) = h$ where $h \in \mathbb{R}^d$ is the output-layer hidden representation. Note that the logit and the model predictive distribution are

$$z = Wh, \qquad p(h) = \mathrm{softmax}(Wh) \in \mathbb{R}^V.$$

Then $G = \frac{1}{B}\sum_{b=1}^{B}(p(h_b) - e_{y_b})h_b^\top$, and the class-wise contribution G_j is simply the j-th row $\mathbf{g}_j$ of G. Therefore $\|G\|_F^2 = \sum_{j=1}^{V}\|\mathbf{g}_j\|^2 = \sum_{j=1}^{V}\|G_j\|^2$, which implies $\kappa = 1$. Consequently,

$$\mathbb{E}\|G\|_F^2 \le \left(1 + \frac{1}{B}\right)\sum_{j\in\mathcal{H}}\varepsilon_j^2 + \frac{1}{B}\mathbb{E}\|h\|^2 + \left(\frac{2}{B} + 2c_q\right)\sum_{j\in\mathcal{T}} q_j s_j^2 + 2\sum_{j\in\mathcal{T}}\mathbb{E}\big[p_j(h)^2\|h\|^2\big].$$

(ii) Intermediate layer. For an intermediate-layer weight matrix $W^{(l)}$ in layer l, take $\xi = h^{(l-1)}$ where $h^{(l-1)}$ is the hidden representation from the $(l-1)$-th layer. Due to the product-wise structure of the gradient, $U_j(\xi)$ can be represented by a rank-1 matrix. Then we can simply choose $\kappa = V$ to obtain the bound. We defer the detailed derivation for this case to Appendix H.

The detailed proof of Theorem 3.2 with the two cases can be found in Appendix G and Appendix H. Here we provide the proof outline.

By definition (6), G_j is the per-token-class gradient corresponding to token j. For any tokens $j \in \mathcal{H}$, we exploit a bias–variance decomposition:

$$\mathbb{E}\big[\|G_j\|^2\big] = \|\mathbb{E}[G_j]\|^2 + \mathrm{tr}\big(\mathrm{Cov}(G_j)\big). \tag{8}$$

We note that $\mathbb{E}[G_j]$ represents the population class-wise gradient contribution of token j. Under this decomposition, we have

$$\|\mathbb{E}[G_j]\|^2 \le \varepsilon_j^2, \qquad \mathrm{tr}\big(\mathrm{Cov}(G_j)\big) \le \frac{1}{B}\left(\varepsilon_j^2 + \mathbb{E}\big[q_j(\xi)(1-q_j(\xi))\|U_j(\xi)\|^2\big]\right),$$

yielding the first two terms in the upper bound in (7). This decomposition and the resulting bounds are used to bound tokens where the population gradient $\mathbb{E}[G_j]$ is relatively well estimated, so that the gradient norm is dominated by the covariance term.

For any tokens $j \in \mathcal{T}$, we isolate the label hits $\{b : y_b = j\}$ via a background–hit decomposition:

$$G_j = \underbrace{\frac{1}{B}\sum_{b=1}^{B} p_j(\xi_b)\,U_j(\xi_b)}_{\text{“softmax background”}} - \underbrace{\frac{1}{B}\sum_{b:y_b=j} U_j(\xi_b)}_{\text{“label hits”}}. \tag{9}$$

The “softmax background” term involves the softmax output $p_j(\xi)$ of all samples in the batch, while the “label hits” term records the possible spikes involved when token j is sampled. Under this decomposition, we have

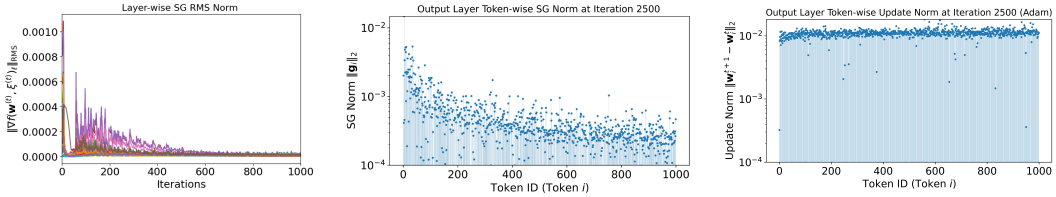
$$\mathbb{E}\left\|\frac{1}{B}\sum_{b=1}^{B} p_j(\xi_b)\,U_j(\xi_b)\right\|^2 \le \mathbb{E}\big[p_j(\xi)^2\|U_j(\xi)\|^2\big], \qquad \mathbb{E}\left\|\frac{1}{B}\sum_{b:y_b=j} U_j(\xi_b)\right\|^2 \le \frac{q_j}{B}s_j^2 + q_j^2 s_j^2,$$

yielding the last two terms in the upper bound in (7). Decomposition (9) and the resulting bounds are used to characterize the regime where the token frequency is rare (so that q_j is small), or where the token is under-trained (so that $p_j(\xi)$ is small).

4 Shrinking the SGD-Adam Gap in LLM Pre-Training

4.1 Challenges in Shrinking the Gap: Irregular Stochastic Gradient

Now we have acknowledged that the effective learning rate of SGD is too small for pre-training. However, increasing the learning rate of SGD further leads to divergence (e.g., see Figure 11 of Zhang et al. [2024]). In this section, we identify two instabilities that arise from the stochastic gradient that destabilize SGD under large learning rates.



(a) (b) (c)

Figure 4: (a): Layer-wise stochastic gradient norm $\|\nabla f(\mathbf{w}^{(t)}; \xi^{(t)})_\ell\|_{\mathrm{RMS}}$ for all main layers $\ell = 1, ..., L$ of the L-layer model. (b): Per-token-class stochastic gradient norm $\|\mathbf{g}_i\|_2$ for $\mathbf{g}_1, ..., \mathbf{g}_V$ the stochastic gradient of each token class output head (up to the first 1000 token IDs). Weight vectors corresponding to frequent token class (lower token ID) receive higher SG norm. (c): Per-token-class update norm $\|\mathbf{w}_i^{t+1} - \mathbf{w}_i^t\|_2$ of Adam for $\mathbf{w}_1, ..., \mathbf{w}_V$ the weight vector of each token class output head (up to the first 1000 token IDs). Adam effectively normalizes the heterogeneous token class SG.

(Type I) Layer-wise Stochastic Gradient Spikes. As illustrated in Figure 4 (a), we observe spikes in the form of unstable large stochastic gradient norm layer-wise, especially in the early stage of training. It is known that such spikes often result in a loss spike when the learning rate is large, leading to unstable training and divergence [Takase et al., 2023].

(Type II) Output Layer SG Disparity. As we see from Figure 4 (b), high-frequency token classes (small token IDs) receive higher magnitude of stochastic gradient at the output layer classification head. Intuitively, this disparity makes the gradient concentrated on high-frequency tokens. This implies that even if the overall gradient norm is relatively small, the per-token-class gradients of the high-frequency tokens can still have a large magnitude, resulting in the training instability for these tokens. We notice that adaptive optimizers like Adam will normalize such heterogeneous token class stochastic gradient to stabilize training under class imbalanced samples, by observing the per-token-class weight update norm Figure 4 (c).

Below, we provide a formal theoretical understanding on why the output-layer gradient norm is skewed toward high-frequency tokens. Denote $[\mathbf{g}_1, ..., \mathbf{g}_V] \in \mathbb{R}^{d \times V}$ as the output layer gradient. We show that the per-token-class gradient norm $\mathbb{E}\|\mathbf{g}_j\|^2$ is approximately proportional to the token frequency in the early training stage, which can lead to substantial disparities across tokens.

Theorem 4.1. *Assume there exists a constant $0 < c \ll B$ such that $\mathbb{E}\big[p_j(h)^2\|h\|^2\big] \le \frac{c^2}{B^2}\mathbb{E}\|h\|^2$. In addition, define $m_j := \mathbb{E}[h \mid y = j]$, $s_j^2 := \mathbb{E}[\|h\|^2 \mid y = j]$ and assume that $0 < s_j^2 < \infty$ for the tokens under consideration. Then,*

$$\mathbb{E}\|\mathbf{g}_j\|^2 \;\ge\; \frac{1}{2B}\, q_j\, s_j^2 \;-\; \frac{c^2}{B^2}\,\mathbb{E}\|h\|^2.$$

Consequently, for any two tokens i, j,

$$\frac{\mathbb{E}\|\mathbf{g}_i\|^2}{\mathbb{E}\|\mathbf{g}_j\|^2} \;\ge\; \frac{\frac{1}{2B}q_i s_i^2 - \frac{c^2}{B^2}\mathbb{E}\|h\|^2}{\frac{2q_j}{B}\, s_j^2 + 2q_j^2\|m_j\|^2 + \frac{2c^2}{B^2}\,\mathbb{E}\|h\|^2}. \tag{10}$$

Theorem 4.1 provides the lower bound on the per-token-class gradient in the early training stage, showing that it scales with the token frequency. In particular, when two tokens differ significantly in frequency, i.e., q_i/q_j is large, we obtain

$$\frac{\mathbb{E}\|\mathbf{g}_i\|^2}{\mathbb{E}\|\mathbf{g}_j\|^2} = \Omega\left(\frac{q_i}{q_j}\right)$$

for small enough c^2/B^2. This explains the empirical observation that high-frequency tokens tend to exhibit much larger per-token-class gradient norms in the output layer. Furthermore, for infrequent tokens with sufficiently small $q_j\|m_j\|$, the lower bound of the ratio in (10) increases with B over a large range of batch sizes, given other terms fixed. Empirically, we also observe greater disparity between per-token-class gradient norms at larger batch sizes; see Figure 7 in Appendix A.

4.2 Large Learning Rate SGD Enabled via Clipping Shrinks the Gap

With the knowledge about the Type-I and Type-II irregularity of the stochastic gradients that can potentially cause the SGD instability, we use two simple forms of clipping, being layer-wise RMS

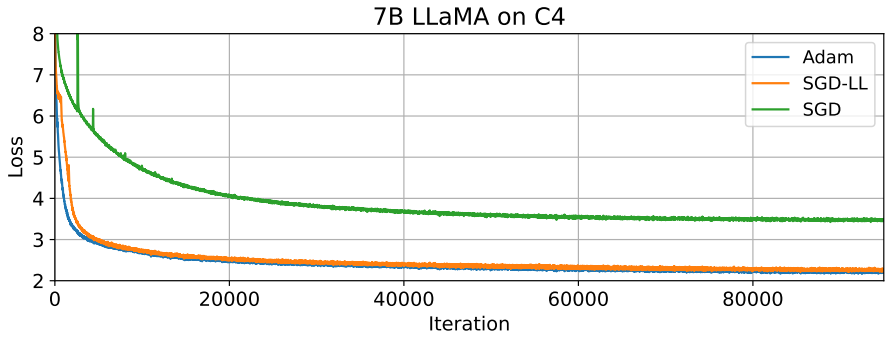


Figure 5: Training loss curves for the 7B model pre-trained on 100B C4 tokens. The loss trajectory of SGD-LL closely matches that of Adam, whereas SGD maintains substantially higher loss.

	130M (2.6B tokens)			**350M** (7.8B tokens)			**1B** (20B tokens)			**7B** (100B tokens)		
	Val. Loss	PPL	LR	Val. Loss	PPL	LR	Val. Loss	PPL	LR	Val. Loss	PPL	LR
Adam	3.11	22.46	3×10^{-3}	2.78	16.09	2×10^{-3}	2.51	12.32	2×10^{-3}	2.21	9.15	1×10^{-3}
SGD	5.07	158.90	0.5	4.14	63.22	0.5	3.95	51.87	0.5	3.48	32.32	0.25
SGD-LL	3.24	25.70	100	2.85	17.35	200	2.60	13.47	300	2.26	9.63	300

Table 1: Comparison on training various model sizes on C4. We report validation loss, perplexity (PPL) and base learning rate (LR). For the 130M and 350M models we use a 0.5M token batch size while for the 1B and 7B we increase it to 1M tokens.

norm clipping and output layer per-token-class clipping, to mitigate the issues and enable SGD with Large Learning rate, which we refer to as SGD-LL.

More specifically, define the clipping operator as $\text{clip}_{r,\|\cdot\|}(\mathbf{x}) := \min\left\{\mathbf{1}, \frac{\mathbf{r}}{\|\mathbf{x}\|}\right\}\mathbf{x}$. For layer weights $\mathbf{w}_1, ..., \mathbf{w}_{L-1}$ and their momentums $\mathbf{m}_1, ..., \mathbf{m}_{L-1}$, we apply layer-wise RMS norm clipping:

$$\mathbf{w}_\ell^{(t+1)} = \mathbf{w}_\ell^{(t)} - \text{clip}_{\tau,\|\cdot\|_{\text{RMS}}}\left(\eta_t \mathbf{m}_\ell^{(t)}\right), \qquad \text{for } \ell = 1, ..., L-1.$$

For the output layer classifier head class weights $\mathbf{w}_{L,1}, ..., \mathbf{w}_{L,V}$ and their momentums $\mathbf{m}_{L,1}, ..., \mathbf{m}_{L,V}$, we additionally apply output layer per-token-class clipping:

$$\mathbf{w}_L^{(t+1)} = \mathbf{w}_L^{(t)} - \text{clip}_{\tau,\|\cdot\|_{\text{RMS}}}\left(\eta_t \widetilde{\mathbf{m}}_L^{(t)}\right), \quad \widetilde{\mathbf{m}}_{L,i}^{(t)} = \text{clip}_{\delta,\|\cdot\|_2}\left(m_{L,i}^{(t)}\right), \qquad \text{for } i = 1, ..., V.$$

Here $\tau > 0$ is the layer-wise clipping threshold and $\delta > 0$ is the per-token-class clipping threshold.

Notice that the layer-wise RMS norm clipping ensures that $\|\mathbf{w}_\ell^{(t+1)} - \mathbf{w}_\ell^{(t)}\|_{\text{RMS}} \leq \tau$, ensuring no layers will be impacted by the Type-I gradient spikes. Moreover, it only clips the update during gradient spikes. Separately, the output layer clipping avoids the classifier head of frequent classes to diverge when receiving disproportionately large Type-II heterogeneous stochastic gradient. Finally, we remark that we find both types of clipping to be necessary to achieve the larger admissible learning rate for SGD (see Appendix D.1 for ablation).

We present our main numerical results in Table 1. We observe that SGD-LL's performance is close to Adam with a small margin, significantly improving from SGD. For instance, in training the 7B LLaMA model, the relative performance gap in validation loss reduces from 57.47% to just 2.26%.

Additionally, the training loss curves of each algorithm are illustrated Figure 5 for the 7B model and in Appendix B for the rest, where we observe that the training trajectory of SGD-LL closely follows that of Adam, while the SGD ones show a considerable gap. We also include further experiments in the Appendix, including pre-training GPT-2 models on FineWeb dataset D.5, multi-seed runs D.6, overtraining experiments D.7, and downstream results with the 7B model D.8.

Finally we note that the two clipping thresholds are chosen sufficiently large so that the intervention specifically targets the gradient irregularities, leaving the majority of the SGD's update unchanged; see Figure 6 in Appendix for illustration and D.2 and D.3 for a quantitative analysis of the clipping

frequency and the update cosine similarity. Together with the performance results, this supports the hypothesis that the lack of large learning rates is a key limiting factor for SGD in pre-training.

5 Conclusions

Overall, our work provides both empirical and theoretical evidence revealing key underlying reasons behind the SGD's poor performance on LLM pre-training. We observe that the smaller update magnitudes of SGD, induced by its much lower effective learning rate relative to Adam, lead to slower training progress. Further, we theoretically connect the need for larger effective learning rates to the small gradient norms during pre-training. Finally, we identify two forms of gradient irregularities causing SGD to become unstable at the large learning rate regime; after appropriate handling we demonstrate that this regime closes most of the gap with Adam.

A limitation of this work is that the current simple remedy leaves a small residual performance gap to Adam. However, we emphasize that the focus of this study is not to propose new optimization algorithms, but rather on gaining understanding about important pre-training phenomena that shed more light to the important open question about the SGD and Adam performance gap in pre-training.

Acknowledgments and Disclosure of Funding

A. Glentis, D. Li, C.Y. Yau and M. Hong are partly supported by NSF Awards ECCS-2426064 and EPCN-2311007.

This work used NVIDIA A100 GPUs at NCSA Delta through allocation CIS261088 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program [Boerner et al., 2023], which is supported by U.S. National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

The authors further acknowledge the Minnesota Supercomputing Institute (MSI) at the University of Minnesota for providing resources that contributed to the research results reported within this paper.

References

Guangsheng Bao, Zhiyang Teng, and Yue Zhang. Token-level fitting issues of seq2seq models. *arXiv preprint arXiv:2305.04493*, 2023.

Timothy J. Boerner, Stephen Deems, Thomas R. Furlani, Shelley L. Knuth, and John Towns. ACCESS: Advancing innovation: NSF's advanced cyberinfrastructure coordination ecosystem: Services & support. In *Practice and Experience in Advanced Research Computing*. Association for Computing Machinery, 2023. doi: 10.1145/3569951.3597559.

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.

Woojin Chung and Jeonghoon Kim. Exploiting vocabulary frequency imbalance in language model pre-training. *arXiv preprint arXiv:2508.15390*, 2025.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL `https://zenodo.org/records/12608602`.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models (2022). *arXiv preprint arXiv:2203.15556*, 2022.

Nitish Shirish Keskar and Richard Socher. Improving generalization performance by switching from adam to sgd. *arXiv preprint arXiv:1712.07628*, 2017.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations (ICLR)*, 2015.

Frederik Kunstner, Alan Milligan, Robin Yadav, Mark Schmidt, and Alberto Bietti. Heavy-tailed class imbalance and why adam outperforms gradient descent on language models. *Advances in Neural Information Processing Systems*, 37:30106–30148, 2024.

Martin Marek, Sanae Lotfi, Aditya Somasundaram, Andrew Gordon Wilson, and Micah Goldblum. Small batch size training for language models: When vanilla sgd works, and why gradient accumulation is wasteful. *arXiv preprint arXiv:2507.07101*, 2025.

Yan Pan and Yuanzhi Li. Toward understanding why adam converges faster than sgd for transformers. *arXiv preprint arXiv:2306.00204*, 2023.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL `https://openreview.net/forum?id=n6SCkn2QaG`.

Steven T Piantadosi. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review*, 21(5):1112–1130, 2014.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Herbert Robbins and Sutton Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400 – 407, 1951. doi: 10.1214/aoms/1177729586. URL `https://doi.org/10.1214/aoms/1177729586`.

Andrei Semenov, Matteo Pagliardini, and Martin Jaggi. Benchmarking optimizers for large language model pretraining. *arXiv preprint arXiv:2509.01440*, 2025.

Teodora Srećković, Jonas Geiping, and Antonio Orvieto. Is your batch size the problem? revisiting the adam-sgd gap in language modeling. *arXiv preprint arXiv:2506.12543*, 2025.

Sho Takase, Shun Kiyono, Sosuke Kobayashi, and Jun Suzuki. Spike no more: Stabilizing the pre-training of large language models. *arXiv preprint arXiv:2312.16903*, 2023.

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*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Kaiyue Wen, David Hall, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them. *arXiv preprint arXiv:2509.02046*, 2025.

Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? *Advances in Neural Information Processing Systems*, 33:15383–15393, 2020.

Yushun Zhang, Congliang Chen, Tian Ding, Ziniu Li, Ruoyu Sun, and Zhiquan Luo. Why transformers need adam: A hessian perspective. *Advances in neural information processing systems*, 37: 131786–131823, 2024.

A Additional Experimental Figures

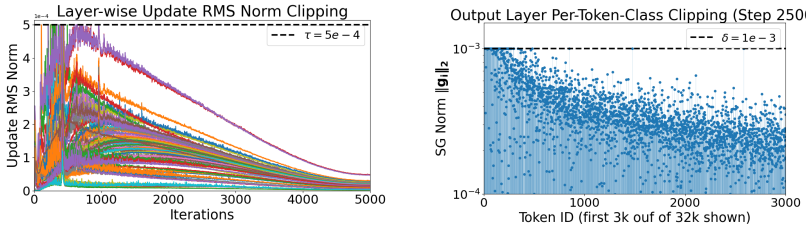


Figure 6: Illustration of the two forms of gradient clipping in SGD-LL while training LLaMA 130M, "treating" Type-I (Left) and Type-II (Right) gradient irregularities. Notice that clipping is targeted to irregularities and the vast majority of gradients remain unchanged. In the left figure update norms of different layers are represented by different colors.

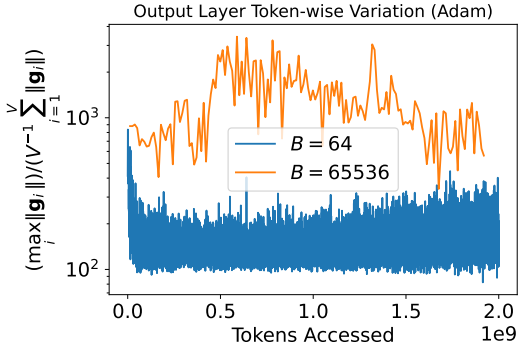


Figure 7: Measuring the token class gradient imbalance by the ratio between maximum token class gradient norm and average token-class gradient norm: $(\max_{i=1,\ldots,V} \|\mathbf{g}_i\|_2)/(V^{-1}\sum_{i=1}^{V}\|\mathbf{g}_i\|_2)$.

B Training Loss Curves

We plot the training loss curves from the 130M-1B pre-training experiments below. We observe that the SGD-LL trajectory closely follows that of Adam, while SGD shows a significant gap.

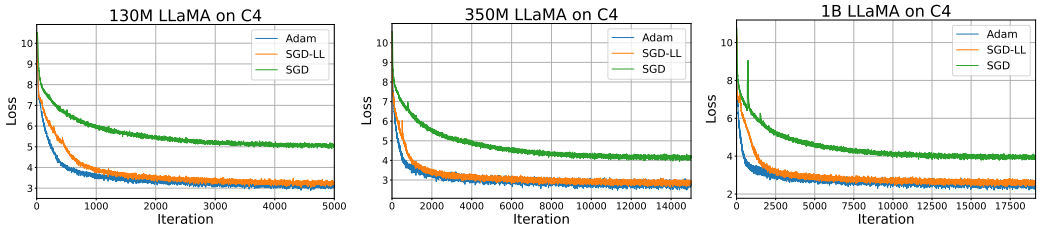


Figure 8: Training loss for model sizes 130M-1B.

C Experimental Settings

For our main analysis we pre-train the LLaMA Touvron et al. [2023a] decoder-only transformer model with about 130M trainable parameters on the C4 (Colossal Clean Crawled Corpus) dataset Raffel et al. [2020]. For tokenization, we use the T5 tokenizer Raffel et al. [2020] with a vocabulary of 32,000 tokens. We use a data-parallel setup with batch size 2048 (unless stated otherwise in the text) and sequence length 256, giving a token batch size of 524,288 tokens. We train for a total of 5,000 steps, such that the model sees roughly the Chinchilla compute-optimal Hoffmann et al. [2022] number of tokens (2.6B, i.e., 20 tokens per model parameter). We train using Pytorch's Automatic Mixed Precision (AMP, BF16/FP32) and use the typical cosine learning rate scheduler with linear warm-up for the first 10% of the iterations.

For our large scale experiments, we train the 1B LLaMA model with 256 sequence length and 4096 batch size and the 7B LLaMA model with 512 sequence length (following Semenov et al. [2025]) and 2048 batch size. Both cases result to a token batch size of 1,048,576 tokens. We choose to increase the sequence length for the 7B model in order to be able to perform zero-shot downstream testing (see Section D.8) which tends to need larger sequence lengths depending on the task. The 1B and 7B models are trained on 20B and 100B C4 tokens, respectively.

We use the loss and perplexity (defined as the exponential of the loss) of the final model checkpoint, measured on the typical C4 validation split, as our evaluation metrics. Learning rates are tuned based on best validation perplexity.

For SGD-LL, for the 130M and 350M models we choose $\tau = 5 \times 10^{-4}$. For the 1B model and 7B, we choose $\tau = 2 \times 10^{-4}$ and $\tau = 1 \times 10^{-4}$, respectively. We adopt $\delta = 10^{-3}$ for all models. Furthermore, following typical choices we use $\beta = 0.9$ for the first-order momentum of both SGD and Adam and $\beta_2 = 0.95$ for the second order momentum of Adam Brown et al. [2020], Touvron et al. [2023a,b]. We set weight decay to zero for all our main experiments in order to isolate the effect of the base optimizers, but include an ablation on weight decay in Section D.4.

Finally, we note that we trained the 130M and 350M models on 4 NVIDIA A100-40GB GPUs and the larger 1B model on 4 NVIDIA H100-96GB GPUs. The time it takes to pre-train the 130M, 350M and 1B models in our setup is about 2, 8 and 30 hours, respectively.

D Additional Experiments

D.1 Ablations on necessity of both clipping types

In this section we show that using only one of the two clipping types is inadequate to enable large learning rate SGD to close most of the gap.

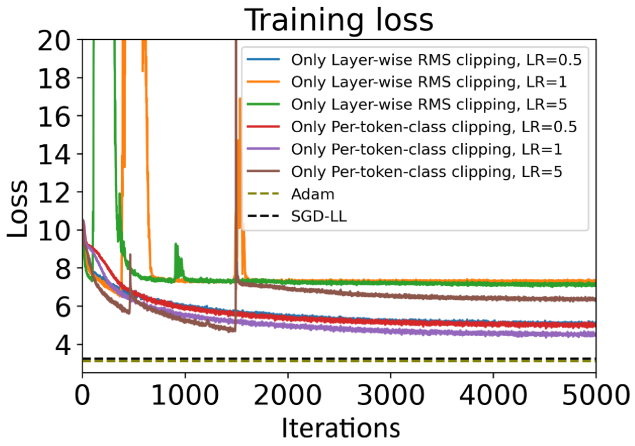


Figure 9: Training loss figure of 130M LLaMA on C4 pre-trained using SGD with either Layer-wise update RMS Norm clipping or output layer per-token-class clipping. We observe that in either case increasing the learning rate higher than 1 leads to loss spikes and divergence.

	Learning Rate	Val. Loss
Adam	3e-3	3.11
SGD + Layer-wise-RMS-clip	0.5	5.06
SGD + Layer-wise-RMS-clip	1	7.29 (diverged)
SGD + Layer-wise-RMS-clip	5	7.12 (diverged)
SGD + Per-token-class-clip	0.5	5.00
SGD + Per-token-class-clip	1	4.52
SGD + Per-token-class-clip	5	6.35 (diverged)
SGD-LL (both clippings)	100	3.24

Table 2: Final validation losses for each strategy corresponding to the trainings shown in Figure 9.

D.2 Ablations on clipping frequency

In this ablation we further analyze the impact of clipping in the final update. It is shown in Figure 6 that the applied clippings only affects a small number of layers and output classes. We further quantify this amount in the Figure below for the 130M and 1B models. We can see that out of 32,000 token classes, less than 100 classes are clipped during 98% of the whole training period in both top figures. Similarly, both bottom figures show that most of the time no layers are clipped via the layerwise clipping.

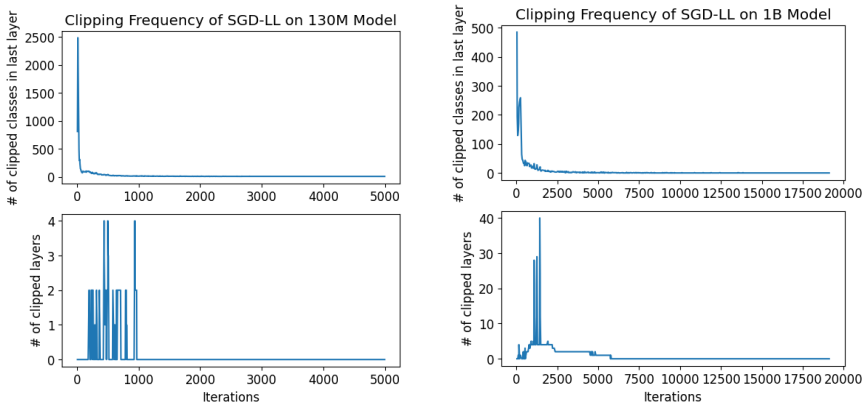


Figure 10: The number of token classes clipped in the last layer (top) and the number of clipped intermediate layers (bottom) for the 130M (left) and 1B (right) models during training.

D.3 Ablation on update cosine similarity after clipping

We show that the clipping applied to the update does not substantially change the SGD's update direction during training. Specifically, each training step we computed the cosine similarity between the full vector update before and after SGD-LL's clipping and present the result in the Figure below. We observe that, after the unstable early training phase, the cosine similarity peaks to high values close to 1.

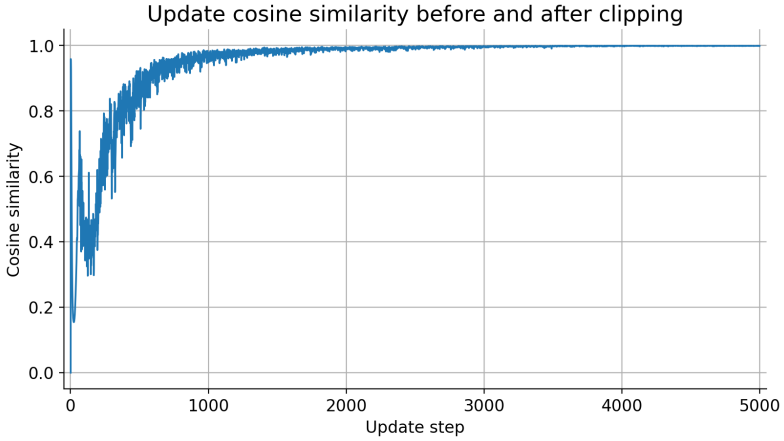


Figure 11: Cosine similarity between the full vector update before and after SGD-LL's clipping during pre-training the 130M LLaMA on C4.

D.4 Further tuning of Adam hyperparameters

Below we show that further tuning of Adam hyperparameters (β_2 and weight decay) has very limited impact on its performance and does not substantially change the observed gap. Such small differences are mainly relevant for the design of new state-of-the-art optimizers which is not the focus of our work.

Ablation on β_2 of Adam:

β_2	0.9	0.95	0.98	0.99	0.999
Val. Loss	3.115	3.112	3.114	3.114	3.145

Table 3: Tuning the β_2 hyperparameter of Adam on the 130M model with 0.5M token batch size and learning rate 3e-3. Using $\beta_2 = 0.95$ gives the best validation loss (which is the value used in the main experiments).

Ablation on weight decay:

wd	0.0	0.01	0.05	0.1	0.2	0.5
Val. Loss	3.112	3.110	3.103	3.100	3.096	3.106

Table 4: Tuning the weight decay (wd) hyperparameter of Adam(W) on the 130M model with 0.5M token batch size and learning rate 3e-3. Note that when $wd = 0$, AdamW practically reduces to Adam. We observe that the use of weight decay can give a small performance improvement, however its effect is minor (only about 0.5% improvement in loss).

D.5 Experiments with GPT-2 architecture

	NanoGPT (125M)	GPT-medium (355M)
Adam	3.34	3.21
SGD-LL	3.60	3.28
SGD	5.00	4.78

Table 5: Validation loss comparison in pre-training GPT-2 style models on FineWeb [Penedo et al., 2024] (2.6B tokens for NanoGPT and 7.8B for GPT-medium). We observe a similar pattern with the main experiments, where the SGD-LL has significantly closer loss to Adam than SGD. Note that SGD (without the added clipping) exhibited unstable training dynamics.

D.6 Multiple Random Seed Runs

We extend our results of 130M model in Table 1 with different random seed by the following table that records the last iterate validation loss. Every run uses the same hyperparameters as listed in C.

Seed	42	1042	2042	3042
Adam	3.11	3.11	3.11	3.11
SGD	5.07	6.13	5.05	5.05
SGD-LL	3.24	3.26	3.25	3.24

Table 6: Validation loss across random seeds. We observe minimal variance for Adam and SGD-LL, while for SGD there is one run that yield even higher loss. This validates that our main results about reducing the gap are consistent and not affected by experimental noise.

D.7 Overtraining experiments

We performed additional experiments with extended number of tokens reaching 20 times more than the Chinchilla compute optimal ratio [Hoffmann et al., 2022] for the 130M model using the same settings as in our main experiments and we show the results bellow:

4× Chinchilla

	Val. Loss	Relative Loss Increase from Adam
Adam	2.94	0%
SGD	4.04	37.41%
SGD-LL	3.03	3.06%

Table 7: Final validation losses and relative loss increase from Adam when training at 4 times the Chinchilla tokens, i.e., 10.4B tokens.

8× Chinchilla

	Val. Loss	Relative Loss Increase from Adam
Adam	2.89	0%
SGD	3.70	28.03%
SGD-LL	2.95	2.08%

Table 8: Final validation losses and relative loss increase from Adam when training at 8 times the Chinchilla tokens, i.e., 20.8B tokens.

20× Chinchilla

	Val. Loss	Relative Loss Increase from Adam
Adam	2.85	0%
SGD	3.40	19.30%
SGD-LL	2.89	1.40%

Table 9: Final validation losses and relative loss increase from Adam when training at 20 times the Chinchilla tokens, i.e., 52B tokens.

In this highly overtrained regime we observe a similar pattern as in our main experiments, that, after addressing the gradient irregularities, running SGD with large LR (i.e., LR=100) is able to close most of the gap to Adam. Note that SGD (without the gradient clipping) diverges for LR greater than 0.5, similar to our main experiments. We did not observe any problems (such as loss spikes or output activation issues) with SGD-LL at this scale.

D.8 Downstream results

Task	Adam	SGD-LL	SGD
arc_challenge	0.3055	0.2867	0.2022
arc_easy	0.6473	0.6242	0.2942
boolq	0.6003	0.5679	0.3783
hellaswag	0.5181	0.4881	0.2728
openbookqa	0.2660	0.2600	0.1780
piqa	0.7606	0.7628	0.6192
sciq	0.8860	0.8510	0.1160
winogrande	0.6259	0.5762	0.4925
Average	0.5762	0.5521	0.3192
Relative drop from Adam	0%	4.18%	44.60%

Table 10: Zero-shot downstream accuracy on 8 tasks of LM-Eval [Gao et al., 2024] using the 7B models we pre-trained on 100B C4 tokens using Adam, SGD-LL and SGD. The downstream results follow a similar pattern to the pre-training ones, as enabling large learning rate for SGD reduces the downstream gap to Adam (relative drop in accuracy) from 44.60% to just 4.18%.

D.9 Distance from initialization experiments

In the Table below we record the distance from initialization $\|\mathbf{w}^T - \mathbf{w}^0\|_F$ by each of the three approaches for the 130M-1B models. We can see that the large learning rate enables SGD to move substantially farther from initialization, closer than Adam's parameter displacement.

	130M (2.6B tokens)	350M (7.8B tokens)	1B* (20B tokens)
Adam	1.3×10^3	2.7×10^3	7.2×10^4
SGD	3.4×10^1	6.2×10^1	6.2×10^2
SGD-LL	3.9×10^2	9.5×10^2	1.3×10^4

Table 11: Distance from initialization, measured by $\|\mathbf{w}^T - \mathbf{w}^0\|_F$. $\mathbf{w}^0$ represent the weights at initialization and $\mathbf{w}^T$ the solution reached by the end of training. *For 1B, distances are summed over parameter tensors during training rather than combined into a full-vector norm.

E Further Connections Between the Large Weight-SG Ratio and Theoretical Analysis

In this section, we provide more evidence in relation between Theorem 3.2 and the large weight-SG ratio in LLM training.

We start by checking that the terms $\varepsilon_j, p_j(\xi), c_q$ in Theorem 3.2 are indeed small if we choose an appropriate partition $\mathcal{H}$ and $\mathcal{T}$. In particular, we check the per-token-class loss and the token frequencies during training. In particular, we separate the tokens into ten quantiles, with quantile 0 containing the highest-frequency tokens and quantile 9 containing the lowest-frequency tokens. For each iteration, we count the token occurrence in the mini-batch and demonstrate the average per-token loss from each quantile. Figure 13 illustrates the per-token loss and Figure 14 illustrates the token counts across different frequency quantiles. In this example, we can cluster tokens from quantile 0–7 into the head set $\mathcal{H}$, exhibiting higher frequencies and lower loss, while clustering tokens from quantile 8–9 into the tail set $\mathcal{T}$, exhibiting lower frequencies and higher loss. Therefore, we can apply the bias-variance decomposition (8) to achieve a smaller upper bound as the head tokens are better-trained. Besides, we can apply the background-hit decomposition (9) to the tail tokens to achieve a smaller upper bound since they are rare (i.e., small q_j) and under-trained (i.e., small $p_j(\xi)$).

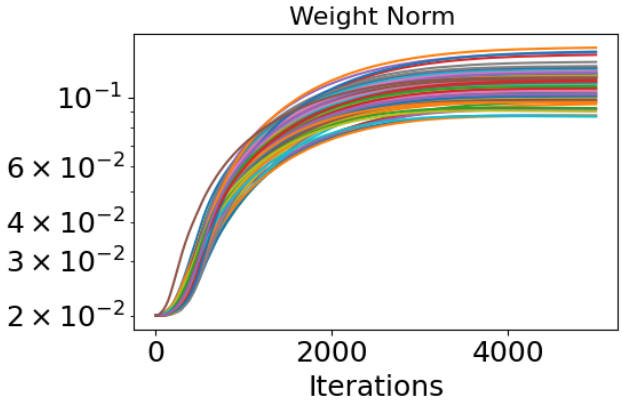


Figure 12: The RMS norms of the weight matrices during training.

By the discussion above, we conclude that $\|G\|_F^2$ is small in LLM training. On the other hand, the norm of the weight matrix is typically lower bounded (see Figure 12). Therefore, the weight-SG ratio becomes large. Moreover, we note that the upper bound (7) is a decreasing function w.r.t. B. Therefore, we expect that the weight-SG ratio should increase as we enlarge the batch size, which is indeed consistent with the observation in Figure 3(c) and Table 12.

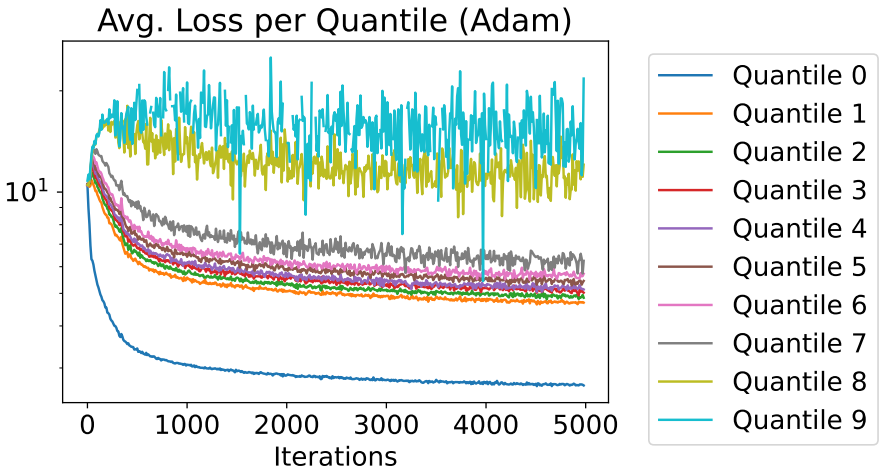


Figure 13: Average loss of the tokens with different frequency quantiles, with Quantile 0 representing the top-10% most frequent token classes and Quantile 9 representing the bottom-10% least frequent token classes. High-frequency tokens exhibits significantly lower loss.

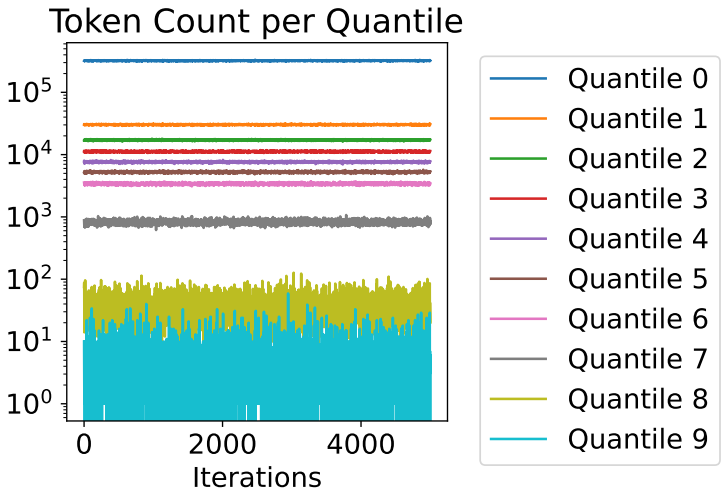


Figure 14: Token count in one batch on different frequency quantiles, with Quantile 0 representing the top-10% most frequent token classes and Quantile 9 representing the bottom-10% least frequent token classes. Low-frequency tokens exhibits substantially rare occurrence.

The large weight-SG norm ratio implies that a small choice of learning rate would limit the optimization algorithm's solution space of $\mathbf{w}^t$ to a small neighborhood around the initialization state $\mathbf{w}^0$. It is evident that the usual learning rate choice of SGD in the large batch setting will leave a significant performance gap to adaptive algorithms [Pan and Li, 2023, Marek et al., 2025, Srećković et al., 2025]. These observations suggest that the limited learning rates constrain SGD's training progress in the large-batch setting, motivating our investigation of targeted clipping as a way of enabling larger updates.

Table 12: Statistics of weight-SG ratio over the training trajectory of different batch sizes in Figure 3(c). As batch size increases, the weight-SG ratio increases.

Batch Size	**Min**	**Max**	**Average**
$B = 64$	1.5×10^3	1.9×10^4	5.9×10^3
$B = 2048$	2.3×10^3	2.2×10^5	2.3×10^4
$B = 65536$	2.6×10^3	4.9×10^5	5.8×10^4

F Experimental Analysis of Magnitude of κ in Theorem 3.2

The constant κ in Theorem 3.2 can be up to $|V|$ in the worst case. However, we observe that κ is small in practical pre-training settings. Below we present an experimental estimation of κ for different layers.

In particular, we estimate κ by calculating the ratio of $\|G\|_F^2 / \sum_j \|G_j\|_F^2$, where G_j is the contribution of token class j to the stochastic gradient $G = \sum_j G_j$. Directly calculating each G_j would result in $|V| = 32,000$ back-propagations for one estimation (where $|V|$ is the vocabulary size). To make the computation tractable, we apply a random-estimation approach to estimate $\sum_j \|G_j\|_F^2$. Specifically, let r_j be independent Rademacher variables, i.e., $r_j \in \{-1, +1\}$. Define $S_r = \sum_{j=1}^{V} r_j G_j$, then $\mathbb{E}_r \|S_r\|_F^2 = \sum_j \|G_j\|_F^2$. We therefore sample S_r by K times to obtain an unbiased estimation of the sum $\Sigma_j \|G_j\|_F^2$, which only requires K times of back-propagation (we use $K = 16$ in our experiments).

We tested on the LLaMA-130M model (with 12 total transformer layers, same setting as our main experiments) with both Adam and SGD. We tracked the gradient of the following four layers: the input embedding layer, Layer 1, Layer 6 and Layer 11. The observed $\|G\|_F^2 / \Sigma_j \|G_j\|_F^2$ ratios are presented in the following table:

Adam

Iteration	**0**	**1000**	**2000**	**3000**	**4000**	**5000**
Embedding	0.98	1.85	1.37	1.30	1.03	1.01
Layer 1	1.00	2.75	1.81	1.58	1.16	1.08
Layer 6	0.98	2.48	2.05	1.75	1.47	1.22
Layer 11	1.00	1.82	1.44	1.18	0.97	0.88

SGD

Iteration	**0**	**1000**	**2000**	**3000**	**4000**	**5000**
Embedding	0.98	1.03	0.96	0.93	0.89	0.86
Layer 1	1.00	1.90	1.33	1.24	1.02	0.94
Layer 6	0.98	1.15	1.01	0.95	0.84	0.80
Layer 11	1.00	0.82	0.80	0.79	0.72	0.72

Across all measured layers, optimizers, and checkpoints, the estimated $\|G\|_F^2 / \Sigma_j \|G_j\|_F^2$ ratio lies between 0.72 and 2.75, being considerably smaller than the worst case $|V| = 32$K. These results provide empirical evidence that the worst-case $\kappa = V$ is indeed highly conservative in practical settings for the embedding and for the intermediate transformer layers. Also note that for the output layer, κ is exactly 1. Therefore, for all layers we experimentally estimate κ to be small, which provides a more concrete connection between our theoretical result and the experimental observations.

G Characterization of the Output-Layer Gradient Norm

In this section, we characterize the gradient norm of the output-layer weight matrix from a token-wise perspective. We begin by analyzing the per-token-class gradient. In particular, by decomposing the per-token-class gradient in different ways, we obtain different bounds that apply to different regimes. We then study how gradient behavior depends on token heterogeneity. Specifically, we show that the per-token-class gradient norm is (i) positively correlated with token frequency in the early stage of training, and (ii) negatively correlated with the batch size in the late stage of training. Details are provided below.

Let V denote the vocabulary size and Δ^{V-1} denote the unit simplex in $\mathbb{R}^V$. The weight matrix of the output-layer is $W \in \mathbb{R}^{V \times d}$, and its j-th row is denoted by $w_j^\top$. Given a hidden representation $h \in \mathbb{R}^d$, the logits and the predictive distribution are

$$z = Wh, \qquad p(h) = \mathrm{softmax}(Wh) \in \Delta^{V-1},$$

where $p_j(h)$ denotes the j-th coordinate. For a true label $y \in \{1, \dots, V\}$, the cross-entropy loss is

$$\ell(h, y) = -\log p_y(h).$$

Therefore, the gradient with respect to the j-th row w_j for a single sample (h, y) is

$$g_j(h, y) \coloneqq \nabla_{w_j} \ell(h, y) = \big(p_j(h) - \mathbf{1}\{y = j\}\big)\, h \in \mathbb{R}^d. \tag{11}$$

Define $r_j(h, y) \coloneqq p_j(h) - \mathbf{1}\{y = j\}$. Then we have $g_j(h, y) = r_j(h, y)h$.

Let $(h, y) \sim \mathcal{D}$ be a generic data-generating distribution and let $\{(h_b, y_b)\}_{b=1}^B$ be a batch sampled i.i.d. from $\mathcal{D}$. The batch-averaged gradient over w_j is thus

$$\bar{g}_j \coloneqq \frac{1}{B} \sum_{b=1}^{B} g_j(h_b, y_b).$$

We allow the conditional label distribution to depend on h:

$$q_j(h) \coloneqq \mathbb{P}(y = j \mid h), \qquad q(h) \in \Delta^{V-1}. \tag{12}$$

The factorized structure of the per-token-class gradient allows us to separate it into a bias term driven by the model–data mismatch $p_j(h) - q_j(h)$ and a variance term induced by the label randomness.

Lemma G.1. *For any fixed h, we have*

$$\mathbb{E}[r_j(h, y) \mid h] = p_j(h) - q_j(h), \qquad \mathbb{E}[g_j(h, y) \mid h] = \big(p_j(h) - q_j(h)\big)\, h.$$

Proof. By (11) and (12), we have $g_j(h, y) = (p_j(h) - \mathbf{1}\{y = j\})h$ and $\mathbb{E}[\mathbf{1}\{y = j\} \mid h] = q_j(h)$. Substituting these identities yields the claim. □

Lemma G.2. *For any fixed h,*

$$\mathbb{E}[r_j(h, y)^2 \mid h] = q_j(h) - 2q_j(h)p_j(h) + p_j(h)^2. \tag{13}$$

Consequently,

$$\mathbb{E}\big[\|g_j(h, y)\|^2 \mid h\big] = \big(q_j(h) - 2q_j(h)p_j(h) + p_j(h)^2\big)\, \|h\|^2. \tag{14}$$

Proof. Conditioned on h, the event $\{y = j\}$ occurs with probability $q_j(h)$, in which case $r_j(h, y) = p_j(h) - 1$; otherwise $r_j(h, y) = p_j(h)$. Computing the conditional expectation gives

$$\mathbb{E}[r_j(h, y)^2 \mid h] = q_j(h)\big(1 - p_j(h)\big)^2 + \big(1 - q_j(h)\big)p_j(h)^2 = q_j(h) - 2q_j(h)p_j(h) + p_j(h)^2.$$

Since $g_j(h, y) = r_j(h, y)h$, (14) follows immediately. □

Lemma G.3. *The conditional covariance of $g_j(h, y)$ given h is*

$$\mathrm{Cov}\big(g_j(h, y) \mid h\big) = q_j(h)\big(1 - q_j(h)\big)\, hh^\top. \tag{15}$$

Proof. We first notice that

$$\mathrm{Var}(r_j(h, y) \mid h) = \mathbb{E}[r_j(h, y)^2 \mid h] - \mathbb{E}[r_j(h, y) \mid h]^2 = q_j(h)(1 - q_j(h)).$$

Since $g_j(h, y) = r_j(h, y)h$ and h is fixed under conditioning,

$$\mathrm{Cov}\big(g_j(h, y) \mid h\big) = \mathrm{Var}(r_j(h, y) \mid h)\, hh^\top = q_j(h)\big(1 - q_j(h)\big)\, hh^\top.$$

□

Lemmas G.1–G.3 characterize the single-sample per-token-class gradient. We next derive bounds for the batch-averaged per-token-class gradient $\bar{g}_j$ under different decompositions and regimes.

Case 1: Well-Trained Regime.

Assume that token j is well trained so that $p_j(h)$ is sufficiently close to $q_j(h)$. In this case, $\mathbb{E}[g_j]$ is small, and the batch-averaged gradient norm is dominated by the covariance term. We begin with the standard bias–variance decomposition.

Lemma G.4. *Assume that* $\{(h_b, y_b)\}_{b=1}^B$ *are sampled i.i.d. from* $\mathcal{D}$*, and that* $\mathbb{E}\|h\|^2 < \infty$*. Let* $\mu_j \coloneqq \mathbb{E}[g_j]$*. Then*

$$\mathbb{E}\big[\|\bar{g}_j\|^2\big] = \|\mu_j\|^2 + \operatorname{tr}\big(\operatorname{Cov}(\bar{g}_j)\big), \tag{16}$$

where

$$\operatorname{Cov}(\bar{g}_j) = \frac{1}{B}\Big(\mathbb{E}\big[q_j(h)(1-q_j(h))\,hh^\top\big] + \operatorname{Cov}\big((p_j(h)-q_j(h))h\big)\Big). \tag{17}$$

Proof. Since $\{(h_b, y_b)\}_{b=1}^B$ are i.i.d. sampled, we have

$$\mathbb{E}[\bar{g}_j] = \mathbb{E}[g_j], \qquad \operatorname{Cov}(\bar{g}_j) = \frac{1}{B}\operatorname{Cov}(g_j).$$

Then (16) follows the standard bias–variance decomposition, and (17) follows from the law of total variance applied to g_j conditional on h, together with Lemma G.1 and G.3. □

Given the representation of the per-token-class gradient in Lemma G.4, we are now able to bound its norm.

Proposition G.5 (Well Trained Regime Scaling)**.** *Assume* $\mathbb{E}\|h\|^2 < \infty$*. Suppose that there exists* $\varepsilon \geq 0$ *such that*

$$\mathbb{E}\big[\|(p_j(h) - q_j(h))h\|^2\big] \leq \varepsilon^2.$$

Then

$$\mathbb{E}\|\bar{g}_j\|^2 \leq \Big(1 + \frac{1}{B}\Big)\varepsilon^2 + \frac{1}{B}\,\mathbb{E}\big[q_j(h)\big(1-q_j(h)\big)\|h\|^2\big].$$

Proof. From Lemma G.4 we have

$$\operatorname{Cov}(\bar{g}_j) = \frac{1}{B}\Big(\mathbb{E}\big[q_j(h)(1-q_j(h))\,hh^\top\big] + \operatorname{Cov}\big((p_j(h)-q_j(h))h\big)\Big).$$

Taking traces on $\operatorname{Cov}(\bar{g}_j)$ gives

$$\operatorname{tr}\big(\operatorname{Cov}(\bar{g}_j)\big) = \frac{1}{B}\Big(\mathbb{E}\big[q_j(h)(1-q_j(h))\,\|h\|^2\big] + \operatorname{tr}\big(\operatorname{Cov}\big((p_j(h)-q_j(h))h\big)\big)\Big). \tag{18}$$

Recall that $\mu_j = \mathbb{E}[g_j] = \mathbb{E}[(p_j(h) - q_j(h))h]$, hence by Jensen's inequality,

$$\|\mu_j\|^2 = \|\mathbb{E}[(p_j(h) - q_j(h))h]\|^2 \leq \mathbb{E}\|(p_j(h) - q_j(h))h\|^2 \leq \varepsilon^2. \tag{19}$$

Furthermore, we have

$$\operatorname{tr}(\operatorname{Cov}\big((p_j(h)-q_j(h))h\big)) = \mathbb{E}\|(p_j(h)-q_j(h))h\|^2 - \|\mathbb{E}[(p_j(h)-q_j(h))h]\|^2 \leq \mathbb{E}\|(p_j(h)-q_j(h))h\|^2 \leq \varepsilon^2. \tag{20}$$

Combining (16), (18), (19) and (20) yields

$$\mathbb{E}\|\bar{g}_j\|^2 \leq \varepsilon^2 + \frac{1}{B}\Big(\mathbb{E}[q_j(h)(1-q_j(h))\|h\|^2] + \varepsilon^2\Big),$$

which proves the claim. □

Remark: In particular, if ε is negligible, then $\mathbb{E}\|\bar{g}_j\|^2 = O(1/B)$. The per-token-class gradient norm $\mathbb{E}\|\bar{g}_j\|$ is therefore in a $O(1/\sqrt{B})$ scaling in this case.

Case 2: Rare Tokens.

Assume that token j has low marginal frequency. In this case, the gradient norm can be characterized via a different decomposition. Specifically, the batch-averaged gradient admits

$$\bar{g}_j = \frac{1}{B}\sum_{b=1}^B (p_j(h_b) - \mathbf{1}\{y_b = j\})h_b = \underbrace{\frac{1}{B}\sum_{b=1}^B p_j(h_b)\,h_b}_{\text{“softmax background”}} - \underbrace{\frac{1}{B}\sum_{b:y_b=j} h_b}_{\text{“label hits”}}. \tag{21}$$

Since $p_j(h_b)$ is typically small for rare tokens in the under-trained regime, $\bar{g}_j$ is often dominated by the "label hits" term. In particular, we obtain the following bound.

Proposition G.6 (Rare Tokens)**.** *Let $q_j \coloneqq \mathbb{P}(y = j)$. Then the batch-averaged per-token-class gradient satisfies*

$$\mathbb{E}\|\bar{g}_j\|^2 \;\le\; \frac{2q_j}{B}\,\mathbb{E}\big[\|h\|^2 \mid y = j\big] \;+\; 2q_j^2\big\|\mathbb{E}[h \mid y = j]\big\|^2 \;+\; 2\mathbb{E}\big[p_j(h)^2\|h\|^2\big]. \tag{22}$$

Proof. We bound $\mathbb{E}\|\bar{g}_j\|^2$ using the decomposition in (21).

First, for the softmax background term, since $\{h_b\}_{b=1}^B$ are i.i.d. samples, we have

$$\mathbb{E}\left\|\frac{1}{B}\sum_{b=1}^{B} p_j(h_b)\,h_b\right\|^2 \;\le\; \frac{1}{B}\sum_{b=1}^{B}\mathbb{E}\,\|p_j(h_b)\,h_b\|^2 = \mathbb{E}\|p_j(h)h\|^2 = \mathbb{E}\big[p_j(h)^2\|h\|^2\big]. \tag{23}$$

Second, for the label-hit term, let $v \coloneqq \mathbf{1}\{y = j\}\,h$. Then

$$\frac{1}{B}\sum_{b:y_b=j} h_b = \frac{1}{B}\sum_{b=1}^{B} v_b$$

with i.i.d. copies v_b of v. Therefore,

$$\mathbb{E}\left\|\frac{1}{B}\sum_{b:y_b=j} h_b\right\|^2 = \frac{1}{B^2}\Big(B\,\mathbb{E}\|v\|^2 + B(B-1)\|\mathbb{E}v\|^2\Big) = \frac{1}{B}\,\mathbb{E}\|v\|^2 + \frac{B-1}{B}\,\|\mathbb{E}v\|^2.$$

By the law of total expectation,

$$\mathbb{E}\|v\|^2 = \mathbb{E}\big[\mathbf{1}\{y = j\}\|h\|^2\big] = q_j\,\mathbb{E}\big[\|h\|^2 \mid y = j\big],$$

and

$$\mathbb{E}v = \mathbb{E}[\mathbf{1}\{y = j\}h] = q_j\,\mathbb{E}[h \mid y = j].$$

Hence,

$$\mathbb{E}\left\|\frac{1}{B}\sum_{b:y_b=j} h_b\right\|^2 = \frac{q_j}{B}\,\mathbb{E}\big[\|h\|^2 \mid y = j\big] + \frac{B-1}{B}\,q_j^2\big\|\mathbb{E}[h \mid y = j]\big\|^2 \;\le\; \frac{q_j}{B}\,\mathbb{E}\big[\|h\|^2 \mid y = j\big] + q_j^2\big\|\mathbb{E}[h \mid y = j]\big\|^2. \tag{24}$$

Combining (23) and (24) yields

$$\mathbb{E}\|\bar{g}_j\|^2 \le 2\mathbb{E}\left\|\frac{1}{B}\sum_{b=1}^{B} p_j(h_b)\,h_b\right\|^2 + 2\mathbb{E}\left\|\frac{1}{B}\sum_{b:y_b=j} h_b\right\|^2 \le 2\mathbb{E}\big[p_j(h)^2\|h\|^2\big] + \frac{2q_j}{B}\,\mathbb{E}\big[\|h\|^2 \mid y = j\big] + 2q_j^2\big\|\mathbb{E}[h \mid y = j]\big\|^2,$$

which proves (22). □

G.1 Analysis of Full Gradient

Given the token-wise bounds above, we now analyze the full gradient matrix.

Let

$$G \;\coloneqq\; \frac{1}{B}\sum_{b=1}^{B}\big(p(h_b) - e_{y_b}\big)h_b^\top \;\in\; \mathbb{R}^{V\times d}$$

denote the mini-batch stochastic gradient of the output-layer weight matrix, where the j-th row of G is $\bar{g}_j^\top$.

Then

$$\|G\|_F^2 = \sum_{j=1}^{V}\|\bar{g}_j\|^2, \qquad \mathbb{E}\|G\|_F^2 = \sum_{j=1}^{V}\mathbb{E}\|\bar{g}_j\|^2.$$

where $\bar{g}_1^\top, \ldots, \bar{g}_V^\top$ are the row (per-token-class) vectors of G. Let $q_j \coloneqq \mathbb{P}(y = j)$ denote the marginal frequency of token j. For each token j, define

$$m_j \coloneqq \mathbb{E}[h \mid y = j], \qquad s_j^2 \coloneqq \mathbb{E}[\|h\|^2 \mid y = j], \qquad \Sigma_j \coloneqq \mathrm{Cov}(h \mid y = j),$$

so that $s_j^2 = \|m_j\|^2 + \mathrm{tr}(\Sigma_j)$.

We partition the vocabulary into a head set $\mathcal{H}$ and a tail set $\mathcal{T}$. We then upper bound the full gradient norm by applying the two token-wise bounds to the two sets, respectively.

Theorem G.7. *Let*

$$G \;\coloneqq\; \frac{1}{B}\sum_{b=1}^{B}\big(p(h_b) - e_{y_b}\big)h_b^\top \;\in\; \mathbb{R}^{V\times d}$$

be the mini-batch stochastic gradient of the output-layer weight matrix. Suppose that the tokens $\{1, ..., V\}$ are separated into two sets $\mathcal{H}$ and $\mathcal{T}$. Denote $c_q \;\coloneqq\; \max_{j\in\mathcal{T}} q_j$. Further, denote

$$\varepsilon_j^2 \;\coloneqq\; \mathbb{E}\big[\|(p_j(h) - q_j(h))h\|^2\big].$$

Then

$$\mathbb{E}\|G\|_F^2 \;\le\; \Big(1 + \frac{1}{B}\Big)\sum_{j\in\mathcal{H}}\varepsilon_j^2 + \frac{1}{B}\,\mathbb{E}\|h\|^2 + \Big(\frac{2}{B} + 2c_q\Big)\sum_{j\in\mathcal{T}} q_j s_j^2 + 2\sum_{j\in\mathcal{T}}\mathbb{E}\big[p_j(h)^2\|h\|^2\big]. \quad (25)$$

Proof. Since $\|G\|_F^2 = \sum_{j=1}^V \|\bar{g}_j\|^2$, it suffices to bound $\sum_{j\in\mathcal{H}}\mathbb{E}\|\bar{g}_j\|^2$ and $\sum_{j\in\mathcal{T}}\mathbb{E}\|\bar{g}_j\|^2$ separately.

For $j \in \mathcal{H}$, we apply Proposition G.5 to obtain

$$\mathbb{E}\|\bar{g}_j\|^2 \le \Big(1 + \frac{1}{B}\Big)\varepsilon_j^2 + \frac{1}{B}\,\mathbb{E}\big[q_j(h)\big(1 - q_j(h)\big)\|h\|^2\big].$$

Summing over $j \in \mathcal{H}$ gives

$$\sum_{j\in\mathcal{H}}\mathbb{E}\|\bar{g}_j\|^2 \le \Big(1 + \frac{1}{B}\Big)\sum_{j\in\mathcal{H}}\varepsilon_j^2 + \frac{1}{B}\sum_{j\in\mathcal{H}}\mathbb{E}\big[q_j(h)\big(1 - q_j(h)\big)\|h\|^2\big]. \quad (26)$$

We now bound the second term in (26), since $\mathcal{H} \subseteq \{1, \ldots, V\}$ and $q_j(h)(1 - q_j(h)) \le q_j(h)$, we have

$$\sum_{j\in\mathcal{H}} q_j(h)\big(1 - q_j(h)\big) \le \sum_{j=1}^{V} q_j(h)\big(1 - q_j(h)\big) = 1 - \sum_{j=1}^{V} q_j(h)^2 \le 1.$$

Therefore,

$$\sum_{j\in\mathcal{H}}\mathbb{E}\big[q_j(h)\big(1 - q_j(h)\big)\|h\|^2\big] = \mathbb{E}\Big[\sum_{j\in\mathcal{H}} q_j(h)(1 - q_j(h))\|h\|^2\Big] \le \mathbb{E}\|h\|^2,$$

and hence the head contribution satisfies

$$\sum_{j\in\mathcal{H}}\mathbb{E}\|\bar{g}_j\|^2 \le \Big(1 + \frac{1}{B}\Big)\sum_{j\in\mathcal{H}}\varepsilon_j^2 + \frac{1}{B}\mathbb{E}\|h\|^2.$$

For $j \in \mathcal{T}$, applying Proposition G.6 yields

$$\mathbb{E}\|\bar{g}_j\|^2 \le \frac{2q_j}{B}\,s_j^2 + 2q_j^2\|m_j\|^2 + 2\mathbb{E}\big[p_j(h)^2\|h\|^2\big].$$

Summing over $j \in \mathcal{T}$ gives

$$\sum_{j\in\mathcal{T}}\mathbb{E}\|\bar{g}_j\|^2 \le \frac{2}{B}\sum_{j\in\mathcal{T}} q_j s_j^2 + 2\sum_{j\in\mathcal{T}} q_j^2\|m_j\|^2 + 2\sum_{j\in\mathcal{T}}\mathbb{E}\big[p_j(h)^2\|h\|^2\big].$$

Note that for all $j \in \mathcal{T}$, we have $q_j \le c_q$, and moreover $\|m_j\|^2 \le s_j^2$. Therefore,

$$\sum_{j\in\mathcal{T}} q_j^2 \|m_j\|^2 \le \sum_{j\in\mathcal{T}} \Big(q_j \cdot c_q\Big) s_j^2 = c_q \sum_{j\in\mathcal{T}} q_j s_j^2,$$

and hence

$$\sum_{j\in\mathcal{T}} \mathbb{E}\|\bar{g}_j\|^2 \le \left(\frac{2}{B} + 2c_q\right) \sum_{j\in\mathcal{T}} q_j s_j^2 + 2\sum_{j\in\mathcal{T}} \mathbb{E}\big[p_j(h)^2 \|h\|^2\big].$$

Combining the head and tail bounds completes the proof. □

Theorem G.7 upper bounds the full gradient norm by separating tokens into head and tail sets. This bound is universal in the sense that it allows arbitrary token-wise heterogeneity and does not depend on the optimization algorithm. The bound shows that the full gradient norm is controlled by (i) the training status of head tokens (captured by ε_j), (ii) the marginal frequencies of tail tokens (captured by c_q and $\{q_j\}_{j\in\mathcal{T}}$), and (iii) the magnitude of the softmax-background term for tail tokens (captured by $\mathbb{E}[p_j(h)^2\|h\|^2]$). To obtain a simpler scaling law, we introduce the following assumptions.

Assumption G.8. Assume that there exists a constant $0 < c_q < 1$ such that the tokens $\{1, ..., V\}$ are separated into two sets $\mathcal{H} = \{j \mid q_j \ge c_q\}$ and $\mathcal{T} = \{j \mid q_j < c_q\}$, satisfying the following conditions:

- (**Well-trained head**) For $j \in \mathcal{H}$, there exists $\varepsilon_j \ge 0$ such that

  $$\mathbb{E}\big[\|(p_j(h) - q_j(h))h\|^2\big] \le \varepsilon_j^2, \qquad \sum_{j\in\mathcal{H}} \varepsilon_j^2 \le \frac{C_{\mathcal{H}}}{B}$$

  for some constant $C_{\mathcal{H}} < \infty$.

- (**Under-trained tail**) For $j \in \mathcal{T}$, there exists a constant $0 < c \ll B$ such that

  $$\mathbb{E}\big[p_j(h)^2\|h\|^2\big] \;\le\; \frac{c^2}{B^2}\,\mathbb{E}\|h\|^2.$$

- (**Tail contains only rare tokens**) There exists a constant $C_{\mathcal{T}} < \infty$ such that $c_q \le C_{\mathcal{T}}/B$.

These assumptions are motivated from the common belief that in LLM pre-training, the head tokens are trained well in a relatively small time, whereas the tail tokens remain under trained for a long time. We in turn justify the assumptions:

Corollary G.9. *Suppose that Assumption G.8 holds. Then*

$$\mathbb{E}\|G\|_F^2 \;\le\; \frac{2C_{\mathcal{H}}}{B} + \frac{1}{B}\,\mathbb{E}\|h\|^2 + \frac{2(1+C_{\mathcal{T}})}{B} \sum_{j\in\mathcal{T}} q_j s_j^2 + \frac{2|\mathcal{T}|c^2}{B^2}\,\mathbb{E}\|h\|^2. \tag{27}$$

In particular, if $\sum_{j\in\mathcal{T}} q_j s_j^2 < \infty$ *and* $\mathbb{E}\|h\|^2 < \infty$*, then* $\mathbb{E}\|G\|_F^2 = O(1/B)$.

Proof. Applying Assumption G.8 to (25) yields (27). □

Corollary G.10. *Assume that Assumptions G.8 holds and further that the output layer weight matrix* W *satisfies* $\|W\|_F \ge C_w$ *for some* $C_w > 0$*. Then*

$$\mathbb{E}\left[\frac{\|W\|_F}{\|G\|_F}\right] = \Omega(\sqrt{B})$$

Proof. Denote

$$C_G \;\coloneqq\; \sqrt{2C_{\mathcal{H}} + \mathbb{E}\|h\|^2 + 2(1+C_{\mathcal{T}}) \sum_{j\in\mathcal{T}} q_j s_j^2 + 2|\mathcal{T}|c^2\mathbb{E}\|h\|^2}.$$

By Corollary G.9, we have $\mathbb{E}\|G\|_F^2 \le C_G^2/B$. Therefore,

$$\mathbb{E}\|G\|_F \le \sqrt{\mathbb{E}\|G\|_F^2} \le \frac{C_G}{\sqrt{B}}.$$

Since $\|W\|_F \geq C_w$, we have

$$\mathbb{E}\left[\frac{\|W\|_F}{\|G\|_F}\right] \geq C_w\,\mathbb{E}\left[\frac{1}{\|G\|_F}\right] \geq \frac{C_w}{\mathbb{E}\|G\|_F} \geq \frac{C_w}{C_G}\sqrt{B},$$

which completes the proof. □

G.2 Lower Bound and Imbalance of the Per-Token-Class Gradient

We next provide a lower bound for the norm of the per-token-class gradient when tokens are not well trained. The result also implies that per-token-class gradient norms can be highly imbalanced across tokens, especially in the early stage of training.

Theorem G.11. *Assume that there exists a constant $0 < c \ll B$ such that*

$$\mathbb{E}\big[p_j(h)^2\|h\|^2\big] \;\leq\; \frac{c^2}{B^2}\,\mathbb{E}\|h\|^2. \tag{28}$$

In addition, define $m_j \coloneqq \mathbb{E}[h \mid y = j]$, $s_j^2 \coloneqq \mathbb{E}[\|h\|^2 \mid y = j]$ and assume $0 < s_j^2 < \infty$ for the tokens under consideration. Then we have

$$\mathbb{E}\|\bar{g}_j\|^2 \;\geq\; \frac{1}{2B}\,q_j\,s_j^2 \;-\; \frac{c^2}{B^2}\,\mathbb{E}\|h\|^2. \tag{29}$$

Consequently, for any two tokens i, j,

$$\frac{\mathbb{E}\|\bar{g}_i\|^2}{\mathbb{E}\|\bar{g}_j\|^2} \;\geq\; \frac{\frac{1}{2B}q_i s_i^2 - \frac{c^2}{B^2}\mathbb{E}\|h\|^2}{\frac{2q_j}{B}\,s_j^2 + 2q_j^2\|m_j\|^2 + \frac{2c^2}{B^2}\,\mathbb{E}\|h\|^2}. \tag{30}$$

Proof. Recall the batch row-gradient decomposition

$$\bar{g}_j = \underbrace{\frac{1}{B}\sum_{b=1}^{B} p_j(h_b)\,h_b}_{=:\,A_j} \;-\; \underbrace{\frac{1}{B}\sum_{b:y_b=j} h_b}_{=:\,B_j}.$$

Therefore,

$$\begin{aligned}\|\bar{g}_j\|^2 = \|A_j - B_j\|^2 &= \|A_j\|^2 + \|B_j\|^2 - 2\langle A_j, B_j\rangle \\ &\geq \|A_j\|^2 + \|B_j\|^2 - \left(2\|A_j\|^2 + \frac{1}{2}\|B_j\|^2\right) = \frac{1}{2}\|B_j\|^2 - \|A_j\|^2,\end{aligned}$$

where the inequality follows from Young's inequality. Taking expectations and using (23) and (28), we obtain

$$\mathbb{E}\|\bar{g}_j\|^2 \geq \frac{1}{2}\mathbb{E}\|B_j\|^2 - \frac{c^2}{B^2}\mathbb{E}\|h\|^2. \tag{31}$$

Next, let $v \coloneqq \mathbf{1}\{y = j\}\,h$. Then

$$\frac{1}{B}\sum_{b:y_b=j} h_b = \frac{1}{B}\sum_{b=1}^{B} v_b$$

with i.i.d. copies v_b of v. Therefore,

$$\mathbb{E}\,\|B_j\|^2 = \frac{1}{B^2}\Big(B\,\mathbb{E}\|v\|^2 + B(B-1)\|\mathbb{E}v\|^2\Big) \geq \frac{1}{B}\,\mathbb{E}\|v\|^2 = \frac{1}{B}\,q_j\,\mathbb{E}[\|h\|^2 \mid y = j] = \frac{1}{B}\,q_j s_j^2. \tag{32}$$

Combining (31) and (32) gives

$$\mathbb{E}\|\bar{g}_j\|^2 \geq \frac{1}{2B}q_j s_j^2 - \frac{c^2}{B^2}\mathbb{E}\|h\|^2. \tag{33}$$

On the other hand, by Proposition G.6, we have

$$\mathbb{E}\|\bar{g}_j\|^2 \;\leq\; \frac{2q_j}{B}s_j^2 \;+\; 2q_j^2\|m_j\|^2 \;+\; \frac{2c^2}{B^2}\,\mathbb{E}\|h\|^2. \tag{34}$$

Combining (33) and (34) gives (30). □

H Characterization of the Intermediate-Layer Gradient Norm

We now extend the result to intermediate layers. The proof follows the same head–tail structure as Theorem G.7, with the output-layer feature vector replaced by the class-dependent intermediate-layer matrix $M_j^{(l)}$, which we will introduce later.

Let $(x, y) \sim \mathcal{D}$, where $y \in \mathcal{V} := \{1, \dots, V\}$. For an intermediate layer l with weight parameter $W^{(l)}$, let

$$a^{(l)} = W^{(l)} h^{(l-1)}, h^{(l)} = \phi(a^{(l)}),$$

where ϕ is the activation. Write

$$h_l^- := h^{(l-1)}.$$

We view the final softmax probability as a function of h_l^-. Specifically, for each token $j \in \mathcal{V}$, define the logits and output probability with respect to token j as

$$z_j := \left[F_{\geq l}(h_l^-)\right]_j, \qquad p_j^{(l)}(h_l^-) := \left[\mathrm{softmax}\left(F_{\geq l}(h_l^-)\right)\right]_j,$$

where $F_{\geq l}$ denotes the remaining network mapping from the layer-$(l-1)$ representation h_l^- to the final logits. Define also

$$q_j^{(l)}(h_l^-) := \mathbb{P}(y = j \mid h^{(l-1)} = h_l^-), \qquad q_j := \mathbb{P}(y = j).$$

For each token $j \in \mathcal{V}$, define

$$M_j^{(l)} := D_\phi \big(J_j^{(l)}\big)^\top \big(h^{(l-1)}\big)^\top,$$

where $J_j^{(l)}$ is the j-th row of the logit Jacobian

$$J^{(l)} = \frac{\partial z}{\partial h^{(l)}},$$

and

$$D_\phi = \mathrm{Diag}\big(\phi'(a^{(l)})\big).$$

Given an i.i.d. mini-batch $\{(x_b, y_b)\}_{b=1}^B$, let

$$h_{b,l}^- := h_b^{(l-1)}, \qquad M_{b,j}^{(l)} := M_j^{(l)}(x_b), \qquad p_{b,j}^{(l)} := p_j^{(l)}(h_b^{(l-1)}).$$

The class-j contribution to the batch-averaged gradient of $W^{(l)}$ is

$$\bar{g}_j^{(l)} := \frac{1}{B} \sum_{b=1}^{B} \big(p_{b,j}^{(l)} - \mathbf{1}\{y_b = j\}\big) M_{b,j}^{(l)}.$$

The full intermediate-layer gradient is

$$G^{(l)} := \nabla_{W^{(l)}} L = \sum_{j=1}^{V} \bar{g}_j^{(l)}.$$

For each token j, define

$$m_{j,l} := \mathbb{E}\Big[M_j^{(l)} \mid y = j\Big], \qquad s_{j,l}^2 := \mathbb{E}\Big[\|M_j^{(l)}\|_F^2 \mid y = j\Big].$$

Assumption H.1. Let $\mathcal{H}, \mathcal{T}$ be a partition of $\mathcal{V}$, where $\mathcal{H}$ is the head set and $\mathcal{T}$ is the tail set. Assume the following.

1. **Well-trained head tokens.** For each $j \in \mathcal{H}$, there exists $\varepsilon_{j,l} \geq 0$ such that

$$\mathbb{E}\left[\left\|\left(p_j^{(l)}(h^{(l-1)}) - q_j^{(l)}(h^{(l-1)})\right) M_j^{(l)}\right\|_F^2\right] \leq \varepsilon_{j,l}^2.$$

2. **Rare tail tokens.** There exists $c_q \in (0,1)$ such that
$$q_j \le c_q, \qquad \forall j \in \mathcal{T}.$$

3. **Finite second moments.** For all $j \in \mathcal{V}$,
$$\mathbb{E}\|M_j^{(l)}\|_F^2 < \infty.$$

Theorem H.2. *Under Assumption H.1,*
$$\frac{1}{V}\mathbb{E}\|G^{(l)}\|_F^2 \le \left(1+\frac{1}{B}\right)\sum_{j\in\mathcal{H}}\varepsilon_{j,l}^2 + \frac{1}{B}\sum_{j\in\mathcal{H}}\mathbb{E}\|M_j^{(l)}\|_F^2 + \left(\frac{2}{B}+2c_q\right)\sum_{j\in\mathcal{T}} q_j s_{j,l}^2 + 2\sum_{j\in\mathcal{T}}\mathbb{E}\Big[\big(p_j^{(l)}(h^{(l-1)})\big)^2\|M_j^{(l)}\|_F^2\Big].$$

Equivalently, if denoting
$$\Gamma_j^{(l)} := \Big\|D_\phi\big(J_j^{(l)}\big)^\top\Big\|_2^2,$$

then
$$\frac{1}{V}\mathbb{E}\|G^{(l)}\|_F^2 \le \left(1+\frac{1}{B}\right)\sum_{j\in\mathcal{H}}\varepsilon_{j,l}^2 + \frac{1}{B}\sum_{j\in\mathcal{H}}\mathbb{E}\Big[\Gamma_j^{(l)}\|h^{(l-1)}\|_2^2\Big] + \left(\frac{2}{B}+2c_q\right)\sum_{j\in\mathcal{T}} q_j s_{j,l}^2 + 2\sum_{j\in\mathcal{T}}\mathbb{E}\Big[\big(p_j^{(l)}(h^{(l-1)})\big)^2\Gamma_j^{(l)}\|h^{(l-1)}\|_2^2\Big].$$

Remark: Compared to the unified formulation in Theorem 3.2, in this case, we take
$$\xi = h^{(l-1)}, \qquad U_j(\xi) = M_j^{(l)} = D_\phi(J_j^{(l)})^\top (h^{(l-1)})^\top,$$
and choose κ as the trivial upper bound V.

Proof. For notational simplicity, throughout the proof we write
$$h_l^- := h^{(l-1)}.$$
Recall that
$$p_j(h_l^-)$$
denotes the final softmax probability of token j, viewed as a function of the layer-$(l-1)$ representation h_l^-. Similarly,
$$q_j(h_l^-) := \mathbb{P}(y = j \mid h^{(l-1)} = h_l^-).$$
For each token $j \in \mathcal{V}$, define the class-wise contribution to the mini-batch stochastic gradient of the l-th layer by
$$\bar{g}_j^{(l)} := \frac{1}{B}\sum_{b=1}^{B}\Big(p_j(h_b^{(l-1)}) - \mathbf{1}\{y_b = j\}\Big) M_{b,j}^{(l)}.$$
Then the full gradient can be decomposed as
$$G^{(l)} = \sum_{j=1}^{V}\bar{g}_j^{(l)}.$$
Unlike the output-layer case, the matrices $\bar{g}_j^{(l)}$ are not supported on disjoint rows. Therefore we use the universal Cauchy bound
$$\left\|\sum_{j=1}^{V}\bar{g}_j^{(l)}\right\|_F^2 \le V\sum_{j=1}^{V}\|\bar{g}_j^{(l)}\|_F^2.$$
Consequently,
$$\frac{1}{V}\mathbb{E}\|G^{(l)}\|_F^2 \le \sum_{j=1}^{V}\mathbb{E}\|\bar{g}_j^{(l)}\|_F^2.$$
It remains to bound the head-token and tail-token contributions separately.

Head tokens. Fix $j \in \mathcal{H}$. Define the single-sample random matrix

$$R_j^{(l)} := \left(p_j(h^{(l-1)}) - \mathbf{1}\{y = j\}\right) M_j^{(l)}.$$

Then

$$\bar{g}_j^{(l)} = \frac{1}{B} \sum_{b=1}^{B} R_{b,j}^{(l)},$$

where $R_{1,j}^{(l)}, \ldots, R_{B,j}^{(l)}$ are i.i.d. copies of $R_j^{(l)}$.

Let

$$\mu_j^{(l)} := \mathbb{E} R_j^{(l)}.$$

By the standard mean–variance decomposition for i.i.d. averages,

$$\mathbb{E}\|\bar{g}_j^{(l)}\|_F^2 = \|\mu_j^{(l)}\|_F^2 + \frac{1}{B} \operatorname{tr}\left(\operatorname{Cov}\left(\operatorname{vec}(R_j^{(l)})\right)\right).$$

Equivalently,

$$\mathbb{E}\|\bar{g}_j^{(l)}\|_F^2 = \|\mu_j^{(l)}\|_F^2 + \frac{1}{B}\left(\mathbb{E}\|R_j^{(l)}\|_F^2 - \|\mu_j^{(l)}\|_F^2\right) \le \|\mu_j^{(l)}\|_F^2 + \frac{1}{B}\mathbb{E}\|R_j^{(l)}\|_F^2. \tag{35}$$

We now bound the two terms on the right-hand side. Conditioning on $h^{(l-1)}$, the matrix $M_j^{(l)}$ and the probability $p_j(h^{(l-1)})$ are fixed, while

$$\mathbb{E}[\mathbf{1}\{y = j\} \mid h^{(l-1)}] = q_j(h^{(l-1)}).$$

Therefore,

$$\mu_j^{(l)} = \mathbb{E}\left[\left(p_j(h^{(l-1)}) - q_j(h^{(l-1)})\right) M_j^{(l)}\right].$$

By Jensen's inequality and the definition of $\varepsilon_{j,l}^2$,

$$\|\mu_j^{(l)}\|_F^2 \le \mathbb{E}\left[\left\|\left(p_j(h^{(l-1)}) - q_j(h^{(l-1)})\right) M_j^{(l)}\right\|_F^2\right] = \varepsilon_{j,l}^2. \tag{36}$$

Next, conditioning again on $h^{(l-1)}$, we have

$$\begin{aligned}
&\mathbb{E}\left[\left(p_j(h^{(l-1)}) - \mathbf{1}\{y = j\}\right)^2 \,\middle|\, h^{(l-1)}\right] \\
=&q_j(h^{(l-1)})\left(1 - q_j(h^{(l-1)})\right) + \left(p_j(h^{(l-1)}) - q_j(h^{(l-1)})\right)^2.
\end{aligned}$$

Hence

$$\begin{aligned}
\mathbb{E}\|R_j^{(l)}\|_F^2 =&\mathbb{E}\left[q_j(h^{(l-1)})\left(1 - q_j(h^{(l-1)})\right)\|M_j^{(l)}\|_F^2\right] \\
&+ \mathbb{E}\left[\left\|\left(p_j(h^{(l-1)}) - q_j(h^{(l-1)})\right) M_j^{(l)}\right\|_F^2\right].
\end{aligned}$$

Using the definition of $\varepsilon_{j,l}^2$, this gives

$$\mathbb{E}\|R_j^{(l)}\|_F^2 \le \mathbb{E}\left[q_j(h^{(l-1)})\left(1 - q_j(h^{(l-1)})\right)\|M_j^{(l)}\|_F^2\right] + \varepsilon_{j,l}^2. \tag{37}$$

Combining (35), (36) and (37), we have

$$\mathbb{E}\|\bar{g}_j^{(l)}\|_F^2 \le \left(1 + \frac{1}{B}\right)\varepsilon_{j,l}^2 + \frac{1}{B}\mathbb{E}\left[q_j(h^{(l-1)})\left(1 - q_j(h^{(l-1)})\right)\|M_j^{(l)}\|_F^2\right].$$

Since

$$0 \le q_j(h^{(l-1)})\left(1 - q_j(h^{(l-1)})\right) \le 1,$$

we further obtain

$$\mathbb{E}\|\bar{g}_j^{(l)}\|_F^2 \le \left(1+\frac{1}{B}\right)\varepsilon_{j,l}^2 + \frac{1}{B}\mathbb{E}\|M_j^{(l)}\|_F^2.$$

Noticing that $\|M_j^{(l)}\|_F^2 = \Gamma_j^{(l)}\|h^{(l-1)}\|^2$, we get

$$\mathbb{E}\|\bar{g}_j^{(l)}\|_F^2 \le \left(1+\frac{1}{B}\right)\varepsilon_{j,l}^2 + \frac{1}{B}\mathbb{E}\left[\Gamma_j^{(l)}\|h^{(l-1)}\|^2\right].$$

Summing over $j \in \mathcal{H}$ yields

$$\sum_{j\in\mathcal{H}}\mathbb{E}\|\bar{g}_j^{(l)}\|_F^2 \le \left(1+\frac{1}{B}\right)\sum_{j\in\mathcal{H}}\varepsilon_{j,l}^2 + \frac{1}{B}\sum_{j\in\mathcal{H}}\mathbb{E}\left[\Gamma_j^{(l)}\|h^{(l-1)}\|^2\right]. \tag{38}$$

Tail tokens. Fix $j \in \mathcal{T}$. The class-wise contribution admits the background–hit decomposition

$$\bar{g}_j^{(l)} = \underbrace{\frac{1}{B}\sum_{b=1}^{B} p_j(h_b^{(l-1)})M_{b,j}^{(l)}}_{=:A_j^{(l)}} - \underbrace{\frac{1}{B}\sum_{b:y_b=j} M_{b,j}^{(l)}}_{=:B_j^{(l)}}.$$

By the elementary inequality

$$\|A-B\|_F^2 \le 2\|A\|_F^2 + 2\|B\|_F^2,$$

we have

$$\mathbb{E}\|\bar{g}_j^{(l)}\|_F^2 \le 2\mathbb{E}\|A_j^{(l)}\|_F^2 + 2\mathbb{E}\|B_j^{(l)}\|_F^2.$$

We first bound the background term. Let

$$X_j^{(l)} := p_j(h^{(l-1)})M_j^{(l)}.$$

Then

$$A_j^{(l)} = \frac{1}{B}\sum_{b=1}^{B} X_{b,j}^{(l)}.$$

Since the samples are i.i.d.,

$$\mathbb{E}\|A_j^{(l)}\|_F^2 = \frac{1}{B}\mathbb{E}\|X_j^{(l)}\|_F^2 + \frac{B-1}{B}\|\mathbb{E}X_j^{(l)}\|_F^2 \le \mathbb{E}\|X_j^{(l)}\|_F^2.$$

Thus

$$\mathbb{E}\|A_j^{(l)}\|_F^2 \le \mathbb{E}\left[p_j(h^{(l-1)})^2\|M_j^{(l)}\|_F^2\right] = \mathbb{E}\left[p_j(h^{(l-1)})^2\Gamma_j^{(l)}\|h^{(l-1)}\|^2\right]. \tag{39}$$

We next bound the label-hit term. Define

$$Y_j^{(l)} := \mathbf{1}\{y=j\}M_j^{(l)}.$$

Then

$$B_j^{(l)} = \frac{1}{B}\sum_{b=1}^{B} Y_{b,j}^{(l)}.$$

Therefore,

$$\mathbb{E}\|B_j^{(l)}\|_F^2 = \frac{1}{B}\mathbb{E}\|Y_j^{(l)}\|_F^2 + \frac{B-1}{B}\|\mathbb{E}Y_j^{(l)}\|_F^2.$$

By definition,

$$\mathbb{E}\|Y_j^{(l)}\|_F^2 = \mathbb{E}\left[\mathbf{1}\{y=j\}\|M_j^{(l)}\|_F^2\right] = q_j\mathbb{E}\left[\|M_j^{(l)}\|_F^2 \mid y=j\right] = q_j s_{j,l}^2.$$

Moreover,

$$\mathbb{E}Y_j^{(l)} = \mathbb{E}\left[\mathbf{1}\{y=j\}M_j^{(l)}\right] = q_j\mathbb{E}\left[M_j^{(l)} \mid y=j\right].$$

Hence

$$\|\mathbb{E}Y_j^{(l)}\|_F^2 = q_j^2 \left\|\mathbb{E}\left[M_j^{(l)} \mid y = j\right]\right\|_F^2.$$

By Jensen's inequality,

$$\left\|\mathbb{E}\left[M_j^{(l)} \mid y = j\right]\right\|_F^2 \le \mathbb{E}\left[\|M_j^{(l)}\|_F^2 \mid y = j\right] = s_{j,l}^2.$$

Therefore,

$$\|\mathbb{E}Y_j^{(l)}\|_F^2 \le q_j^2 s_{j,l}^2.$$

Since $j \in \mathcal{T}$ and $c_q = \max_{j\in\mathcal{T}} q_j$, we have $q_j \le c_q$. Thus

$$q_j^2 s_{j,l}^2 \le c_q q_j s_{j,l}^2.$$

Consequently,

$$\mathbb{E}\|B_j^{(l)}\|_F^2 \le \frac{q_j}{B} s_{j,l}^2 + c_q q_j s_{j,l}^2 = \left(\frac{1}{B} + c_q\right) q_j s_{j,l}^2. \tag{40}$$

Combining (39) and (40) gives

$$\mathbb{E}\|\bar{g}_j^{(l)}\|_F^2 \le 2\mathbb{E}\left[p_j (h^{(l-1)})^2 \Gamma_j^{(l)} \|h^{(l-1)}\|^2\right] + \left(\frac{2}{B} + 2c_q\right) q_j s_{j,l}^2.$$

Summing over $j \in \mathcal{T}$, we obtain

$$\sum_{j\in\mathcal{T}} \mathbb{E}\|\bar{g}_j^{(l)}\|_F^2 \le 2\sum_{j\in\mathcal{T}} \mathbb{E}\left[p_j (h^{(l-1)})^2 \Gamma_j^{(l)} \|h^{(l-1)}\|^2\right] + \left(\frac{2}{B} + 2c_q\right) \sum_{j\in\mathcal{T}} q_j s_{j,l}^2. \tag{41}$$

Combining the head and tail bounds. Putting the head and tail estimates (38) and (41) together, we have

$$\begin{aligned}
\frac{1}{V}\mathbb{E}\|G^{(l)}\|_F^2 &\le \sum_{j=1}^{V} \mathbb{E}\|\bar{g}_j^{(l)}\|_F^2 = \sum_{j\in\mathcal{H}} \mathbb{E}\|\bar{g}_j^{(l)}\|_F^2 + \sum_{j\in\mathcal{T}} \mathbb{E}\|\bar{g}_j^{(l)}\|_F^2 \\
&\le \left(1 + \frac{1}{B}\right) \sum_{j\in\mathcal{H}} \varepsilon_{j,l}^2 + \frac{1}{B} \sum_{j\in\mathcal{H}} \mathbb{E}\left[\Gamma_j^{(l)} \|h^{(l-1)}\|^2\right] \\
&\quad + \left(\frac{2}{B} + 2c_q\right) \sum_{j\in\mathcal{T}} q_j s_{j,l}^2 + 2\sum_{j\in\mathcal{T}} \mathbb{E}\left[p_j (h^{(l-1)})^2 \Gamma_j^{(l)} \|h^{(l-1)}\|^2\right].
\end{aligned}$$

The proof is complete. □

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 in the abstract and introduction are detailedly discussed in the remaining sections of the paper.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: Limitations are discussed in the conclusion section.

   Guidelines:

   - The answer [N/A] 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: The assumptions and complete proof are provided in Appendix G and Appendix H.

Guidelines:

- The answer [N/A] 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: The algorithms we test are fully described in the main body, while the detailed experimental settings are in Appendix C. We further use open-source models and dataset (LLaMA and C4) that are easily accessible.

   Guidelines:

   - The answer [N/A] 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: The C4 training data and the LLaMA architectures used are widely available online. We release our training code in this camera-ready version (GitHub link in the abstract).

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

Answer: [Yes]

Justification: We include our experimental settings in Appendix C.

Guidelines:

- The answer [N/A] 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: We have pre-training experiments with 4 randoms seeds to test the statistical significance of the results in D.6. We observe small variance between Adam and SGD-LL, while SGD is more unstable (can give even worse results). This suggests that our main results and conclusions are consistent and not artifacts of experimental noise. Also note that LLM pre-training experiments are very computationally expensive and thus, it is impractical for us to repeat the large experiments multiple times. This is also standard in the pre-training literature, e.g., see popular recent benchmark paper Wen et al. [2025]. However we do provide detailed information to replicate our experiments.

Guidelines:

- The answer [N/A] 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 report GPU type and memory along with approximate training time in our setup in Appendix C.

   Guidelines:

   - The answer [N/A] 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 paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] 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: [N/A]

    Justification: This is a technical paper and the societal impact is off-topic.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: No new data or models proposed.

    Guidelines:

    - The answer [N/A] 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 all used assets.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] 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: [N/A]

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

    Guidelines:

    - The answer [N/A] 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: [N/A]

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

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.