UNDERSTANDING WHY ADAM OUTPERFORMS SGD: GRADIENT HETEROGENEITY IN TRANSFORMERS

Akiyoshi Tomihari
The University of Tokyo
tomihari@g.ecc.u-tokyo.ac.jp

Issei Sato
The University of Tokyo
sato@g.ecc.u-tokyo.ac.jp

ABSTRACT

Transformers are challenging to optimize with SGD and typically require adaptive optimizers such as Adam. However, the reasons behind the superior performance of Adam over SGD remain unclear. In this study, we investigate the optimization of transformers by focusing on *gradient heterogeneity*, defined as the disparity in gradient norms among parameters. Our analysis shows that gradient heterogeneity hinders gradient-based optimization, including SGD, while sign-based optimization, a simplified variant of Adam, is less affected. We further examine gradient heterogeneity in transformers and show that it is influenced by the placement of layer normalization. Experimental results from fine-tuning transformers in both NLP and vision domains validate our theoretical analyses.

1 INTRODUCTION

Transformers (Vaswani et al., 2017) have achieved significant success across various tasks, especially in natural language processing (NLP). The training of transformers typically relies on adaptive optimization methods such as Adam (Kingma and Ba, 2015), which empirically outperform stochastic gradient descent (SGD) (Schmidt et al., 2021; Choi, 2019), especially for transformers (Zhang et al., 2020b; Kunstner et al., 2023; Zhang et al., 2024a; Kunstner et al., 2024).

Despite the superior performance of Adam, the reasons for its advantage over SGD, particularly during fine-tuning, remain unclear. Adam consistently outperforms SGD, even in full-batch settings, while SignSGD (Bernstein et al., 2018), which behaves similarly to Adam (Li et al., 2025), achieves comparable performance to Adam under the same conditions (Kunstner et al., 2023). This suggests that the performance gap cannot be solely attributed to gradient noise (Zhang et al., 2020b) but rather stems from fundamental differences between SGD and SignSGD, which remain unexplored. The Adam-SGD gap has been partially linked to heavy-tailed label distributions (Kunstner et al., 2024), but this explanation does not fully account for the gap in fine-tuning tasks, where the number of labels can be small. Zhang et al. (2024a) made an important step by associating this gap with *Hessian heterogeneity* in transformers, yet the underlying mechanism remains unclear.

In this study, we take a step toward a better understanding of the performance gap between Adam and SGD by providing a theoretical explanation. Specifically, we analyze the *iteration complexity*—defined as the number of optimization steps required for convergence—and propose that the performance gap arises from parameter-wise heterogeneity. This heterogeneity manifests as both Hessian heterogeneity (Zhang et al., 2024a) and *gradient heterogeneity*, the variation in gradient norms across parameters. Among these, gradient heterogeneity is more tractable for empirical analysis.

We begin by deriving upper bounds on the iteration complexity of gradient-based and sign-based optimization methods in both deterministic and stochastic settings. Our analysis leverages the fact that SGD and SignSGD correspond to steepest descent methods under different norms. The results suggest that gradient-based methods are more sensitive to parameter-wise heterogeneity than their sign-based ones. To further investigate the origin of parameter-wise heterogeneity, we analyze gradient heterogeneity in transformers and examine how it relates to architectural design choices. In particular, we find that applying layer normalization after residual connections significantly amplifies gradient heterogeneity.

Table 1: Comparison with prior studies. ✓: Supported; △: Partially supported; –: Not supported.

Paper	SignSGD	Transformer	Theoretical complexity	Heterogeneity
Zhang et al. (2020b)	–	✓	✓	–
Crawshaw et al. (2022)	✓	△	✓	–
Kunstner et al. (2023)	✓	✓	–	–
Pan and Li (2022)	–	✓	–	–
Kunstner et al. (2024)	✓	△	–	–
Zhang et al. (2024a)	–	✓	–	✓
Ours	✓	✓	✓	✓

Our contributions are summarized as follows. Table 1 compares prior studies with ours.

- We derive upper bounds for the iteration complexity for optimization algorithms in both deterministic and stochastic settings. Our analysis suggests that SGD is highly sensitive to parameter-wise heterogeneity, whereas Adam is less affected (Theorems 4.7 and 4.9).
- To understand parameter-wise heterogeneity, we investigate gradient heterogeneity in transformers, identifying the position of layer normalization as a factor influencing it (Section 4.6).
- Overall, we emphasize that the sign-based nature of Adam helps address optimization challenges caused by parameter-wise heterogeneity, a characteristic of transformer architectures.

2 Related work

Adam in deep learning. Adam (Kingma and Ba, 2015) is a widely used optimization algorithm in deep learning, known for its well-established convergence properties (Zhang et al., 2022). However, the reasons for its superior performance are not yet fully understood. Jiang et al. (2024) empirically observed that Adam tends to converge to parameter regions with uniform diagonal elements in the Hessian, supported by theoretical analysis based on two-layer linear models. Rosenfeld and Risteski (2024) argued that the ability of Adam to handle outliers in features is a critical factor in its effectiveness. Additionally, Kunstner et al. (2024) attributed the performance of Adam in language models to its ability to manage heavy-tailed class imbalance.

Optimization challenges in transformers. A key aspect of transformer optimization is the notable superiority of Adam over SGD. Zhang et al. (2020b) attributed this to the heavy-tailed gradient noise, but Kunstner et al. (2023) later challenged this, arguing that the superior performance of Adam can be attributed to sign-based characteristics rather than gradient noise, supported by full-batch experiments. Li et al. (2025) demonstrated the similarity between Adam and SignSGD in optimization and generalization, and provided a theoretical analysis of SignSGD. Pan and Li (2022) proposed directional sharpness and show that Adam achieves lower values in transformers. Ahn et al. (2024) demonstrated that linear transformers exhibit similar optimization behaviors to standard transformers. Zhang et al. (2024a) revealed that the Hessian spectrum of the loss function with transformers is heterogeneous and suggested that this is one cause of the Adam-SGD performance gap. Zhao et al. (2025b) empirically evaluated various optimization algorithms on transformer architectures. This heterogeneity was later confirmed by Ormaniec et al. (2025), who derived the Hessian of transformers explicitly.

Sign-based optimization and variants. SignSGD, also known as sign descent (Balles and Hennig, 2018), is an optimization method that is computationally efficient and memory-saving, making it suited for distributed training (Bernstein et al., 2018). Through program search, a sign-based optimization algorithm called Lion (evolved sign momentum) was discovered (Chen et al., 2024b), and its effectiveness was shown by Chen et al. (2024a). Adam can be interpreted as a variance-adapted variant of SignSGD. For example, Xie and Li (2024) analyzed the convergence property of Adam by using this property. Similarly, Zhao et al. (2025a) found that sign-based optimizers restore the stability and performance of Adam and proposed using adaptive learning rates for each layer. Additionally, Zhang et al. (2024b) showed that adaptive learning rates do not need to be computed at a coordinate-wise level but can be applied at the level of parameter blocks.

3 PRELIMINARIES

This section introduces the notation and outlines the optimization methods relevant to our study.

3.1 NOTATION

Vectors and matrices. The k-th element of a vector $\boldsymbol{a}$ is denoted by $\boldsymbol{a}_k$, and for a matrix $\boldsymbol{A}$, we use $\boldsymbol{A}_{k,:}$, $\boldsymbol{A}_{:,l}$, and $A_{k,l}$ to denote the k-th row, l-th column, and element at (k, l), respectively. When a vector or matrix is split into blocks, $[\cdot]_b$ denotes the b-th block. The l_q norm is denoted by $\|\cdot\|_q$ for vectors and represents the operator norm for matrices. The all-ones vector and identity matrix of size a are denoted by $\mathbf{1}_a$ and $\boldsymbol{I}_a$, respectively. The operator $\mathrm{blockdiag}(\cdot)$ constructs block diagonal matrices. Derivatives are computed using the numerator layout.

Model. We consider a classification task with C classes and sample space $\mathcal{X}$. The model $\boldsymbol{f}(\cdot; \boldsymbol{\theta}) : \mathcal{X} \to \mathbb{R}^C$ is parameterized by $\boldsymbol{\theta} \in \mathbb{R}^P$, which is divided into B blocks, denoted as $[\boldsymbol{\theta}]_b \in \mathbb{R}^{P_b}$, with $\sum_{b=1}^{B} P_b = P$. It comprises a pre-trained feature extractor $\boldsymbol{\phi}(\cdot) : \mathcal{X} \to \mathbb{R}^h$ and a linear head with weight $\boldsymbol{V} \in \mathbb{R}^{C \times h}$ and bias $\boldsymbol{b} \in \mathbb{R}^C$. The output is given by $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{V}\boldsymbol{\phi}(\boldsymbol{x}) + \boldsymbol{b}$. At the beginning of fine-tuning, $\boldsymbol{\phi}$ remains pre-trained, while $\boldsymbol{V}$ and $\boldsymbol{b}$ are randomly initialized.

Training. The training dataset $\{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$ consists of N samples $\boldsymbol{x}^{(i)} \in \mathcal{X}$ and the corresponding labels $y^{(i)} \in \{1, \ldots, C\}$. The training objective is to minimize the training loss $L(\boldsymbol{\theta}) := \frac{1}{N}\sum_{i=1}^{N} \ell(\boldsymbol{f}(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}), y^{(i)})$. Here, $\ell : \mathbb{R}^C \times \{1, \ldots, C\} \to \mathbb{R}$ denotes the loss function. The element-wise sign function is denoted by $\mathrm{sign}(\cdot)$. The mini-batch loss is denoted by $\widehat{L}(\boldsymbol{\theta})$, and the learning rate at step t is represented by η_t.

3.2 OPTIMIZATION ALGORITHMS

Adam. Adam (Kingma and Ba, 2015) is widely used in deep learning. It uses the first and second moment estimates of the gradient $\nabla\widehat{L}(\boldsymbol{\theta}_t)$, denoted as $\boldsymbol{m}_t$ and $\boldsymbol{v}_t$, computed using an exponential moving average to reduce mini-batch noise. The update is performed coordinate-wise as:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \frac{\widehat{\boldsymbol{m}}_t}{\sqrt{\widehat{\boldsymbol{v}}_t + \epsilon}},$$

where $\widehat{\bullet}$ denotes bias correction and ϵ is a small constant for numerical stability.

Adaptive learning rate and SignSGD. A key feature of Adam is its *adaptive learning rate*, which is computed in a coordinate-wise manner. When the hyperparameter ϵ, which is typically set close to zero, is ignored and the ratio $|\widehat{\boldsymbol{m}}_{t+1}/\sqrt{\widehat{\boldsymbol{v}}_{t+1}}|$ is close to 1, Adam behaves similarly to SignSGD (Balles and Hennig, 2018; Bernstein et al., 2018). SignSGD updates the parameters with momentum $\boldsymbol{m}_t$ as:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \,\mathrm{sign}(\boldsymbol{m}_t).$$

This method has the property that the updates are invariant to the scale of the gradient. In this sense, Adam can be seen as a soft version of SignSGD. Additionally, the optimizer RMSProp (Tieleman and Hinton, 2017), which inspired Adam, was originally motivated by the idea of using the sign of the gradient in a mini-batch setting. RMSProp is similar to Adam but without the momentum term.

SGD and gradient clipping. SGD can also be modified to achieve scale invariance. A simple way to introduce scale invariance is to normalize the learning rate by the gradient norm, a technique known as normalized gradient descent. This method has been shown to be equivalent to gradient clipping up to a constant factor in the learning rate (Zhang et al., 2020a). Gradient clipping is commonly used to stabilize training, particularly in cases where large gradient magnitudes cause instability and is often applied alongside other optimizers. However, a key difference between Adam and SGD is that SGD does not adapt the learning rate in a coordinate-wise manner.

Steepest descent. SGD and SignSGD can be interpreted as updating in the direction of *the steepest descent* (Xie and Li, 2024):

$$\Delta_t \in \arg\min_{\|\Delta\| \le 1} \nabla \widehat{L}(\boldsymbol{\theta}_t)^\top \Delta.$$

The steepest descent direction associated with the norms $\|\cdot\|_2$ and $\|\cdot\|_\infty$ corresponds to the updates of SGD and SignSGD, respectively.

The steepest descent direction satisfies

$$\nabla \widehat{L}(\boldsymbol{\theta}_t)^\top \Delta = -\|\nabla \widehat{L}(\boldsymbol{\theta}_t)\|_*,$$

where $\|\cdot\|_*$ denotes the dual norm of $\|\cdot\|$. Thus, evaluating the gradient norm using the dual norm is a natural choice for analyzing steepest descent algorithms because it measures the largest possible directional derivative within the unit norm constraint.

4 Main results

In this section, we theoretically analyze optimization methods. We first introduce the setting, assumptions (Section 4.1), and complexity measures (Section 4.2), then examine gradient–Hessian correlations (Section 4.3). Next, we derive upper bounds for optimization complexity in deterministic (Section 4.4) and stochastic settings (Section 4.5). Finally, we investigate gradient heterogeneity in transformers (Section 4.6). Our analyses suggest that parameter-wise heterogeneity, which is a characteristic of transformers, contributes to the performance gap between Adam and SGD.

4.1 Setting and assumption

Gradient-based and sign-based sequences. Kunstner et al. (2023) showed that in full-batch settings without gradient noise, SignSGD performs similarly to Adam and outperforms SGD. This suggests that the performance gap between Adam and SGD arises from differences between SignSGD and SGD. Other studies have also used SignSGD as a proxy for Adam in their analyses (Balles and Hennig, 2018; Li et al., 2025; Kunstner et al., 2024).

On the basis of these insights, we analyze the difference between parameter sequences $\{\boldsymbol{\theta}_t^{\text{Grad}}\}_{t=0}^\infty$ and $\{\boldsymbol{\theta}_t^{\text{Sign}}\}_{t=0}^\infty$, referred to as the gradient-based and sign-based sequences, respectively. These sequences correspond to updates performed by gradient-based and sign-based optimization. In deterministic settings, these updates are defined as follows:

$$\boldsymbol{\theta}_{t+1}^{\text{Grad}} = \boldsymbol{\theta}_t^{\text{Grad}} - \eta_t \nabla L(\boldsymbol{\theta}_t^{\text{Grad}}), \quad \boldsymbol{\theta}_{t+1}^{\text{Sign}} = \boldsymbol{\theta}_t^{\text{Sign}} - \eta_t \operatorname{sign}(\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})).$$

In stochastic settings, the loss L is replaced with the mini-batch loss $\widehat{L}$.

Assumptions. We consider fine-tuning settings, in which the parameter $\boldsymbol{\theta}$ can be typically assumed to remain within a region $\mathcal{R}_{\text{FT}}$ throughout training. This assumption restricts $\boldsymbol{\theta}$ to the localized region $\mathcal{R}_{\text{FT}}$, allowing further assumptions to be applied within this region.

Assumption 4.1 (Fine-tuning)**.** The parameter $\boldsymbol{\theta}$ remains within the region $\mathcal{R}_{\text{FT}}$ throughout the training and there exists $\boldsymbol{\theta}_* \in \mathcal{R}_{\text{FT}}$ such that $L_* := L(\boldsymbol{\theta}_*) = \min_{\boldsymbol{\theta} \in \mathcal{R}_{\text{FT}}} L(\boldsymbol{\theta})$.

We assume Hessian Lipschitz continuity, a standard assumption in optimization (Nesterov, 2013).

Assumption 4.2 (Lipschitz continuity (Nesterov, 2013))**.** Within the region $\mathcal{R}_{\text{FT}}$, the loss function L is twice differentiable, and its Hessian matrix is ρ_H-Lipschitz continuous

$$\|\nabla^2 L(\boldsymbol{\theta}) - \nabla^2 L(\boldsymbol{\theta}')\|_2 \le \rho_H \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2.$$

Additionally, Hessians in deep models often exhibit a near block-diagonal structure (Maes et al., 2024; Kunstner et al., 2024; Collobert, 2004; Zhang et al., 2024a), an assumption also adopted in optimization methods (Martens and Grosse, 2015; Zhang et al., 2017). We thus assume a block-diagonal Hessian.

Assumption 4.3 (Near block-diagonal Hessian)**.** Within the region $\mathcal{R}_{\text{FT}}$, the Hessian matrix can be approximated by a block-diagonal matrix with an approximation error δ_D:

$$\|\nabla^2 L(\boldsymbol{\theta}) - \nabla^2 L_D(\boldsymbol{\theta})\|_2 \leq \delta_D, \tag{1}$$

for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathcal{R}_{\text{FT}}$, where

$$\nabla^2 L_D(\boldsymbol{\theta}) \coloneqq \text{blockdiag}(\{[\nabla^2 L(\boldsymbol{\theta})]_b\}_{b=1}^B),$$

represents the block-diagonal approximation.

Note that in Eq. (1), the left-hand side is bounded above by the sum of squared elements in the non-diagonal blocks, following the relationship between $\|\cdot\|_2$ and the Frobenius norm.

4.2 Gradient heterogeneity and complexity measure

We define *gradient heterogeneity* as follows (see Figure 3). Compared with heterogeneity in the Hessian spectrum (Zhang et al., 2024a), it is computationally more tractable to analyze.

Definition 4.4 (Gradient heterogeneity)**.** The gradient heterogeneity is defined as the disparity in gradient norms across different parameter blocks, $\{\|[\frac{\partial L(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}]_b\|_2\}_{b=1}^B$.

Next, we introduce two measures used in our analysis. In the following definitions, Λ_G weights the operator norm of each Hessian block by its corresponding gradient norm, while Λ_P weights it by the parameter dimension. These definitions ensure that the weights of all Hessian blocks sum to 1, as shown by the equalities: $\sum_{b=1}^{B} \|[\nabla L(\boldsymbol{\theta})]_b\|_2^2 / \|\nabla L(\boldsymbol{\theta})\|_2^2 = \sum_{b=1}^{B} P_b/P = 1$.

Definition 4.5 (Weighted Hessian complexity)**.** The gradient-weighted Hessian complexity Λ_G and parameter-weighted Hessian complexity Λ_P are defined as:

$$\Lambda_G \coloneqq \sup_{\boldsymbol{\theta} \in \mathcal{R}_{\text{FT}}} \sum_{b=1}^{B} \frac{\|[\nabla L(\boldsymbol{\theta})]_b\|_2^2}{\|\nabla L(\boldsymbol{\theta})\|_2^2} \|[\nabla^2 L(\boldsymbol{\theta})]_b\|_2, \quad \Lambda_P \coloneqq \sup_{\boldsymbol{\theta} \in \mathcal{R}_{\text{FT}}} \sum_{b=1}^{B} \frac{P_b}{P} \|[\nabla^2 L(\boldsymbol{\theta})]_b\|_2.$$

4.3 Gradient-Hessian correlation

As shown in Figure 1, large Hessian operator norms $\|[\nabla^2 L(\boldsymbol{\theta})]_b\|_2$ are often associated with large gradient magnitudes $\|[\nabla L(\boldsymbol{\theta})]_b\|_2$. In contrast, no such correlation is observed between Hessian $\|[\nabla^2 L(\boldsymbol{\theta})]_b\|_2$ and parameter dimension P_b (Appendix F.2). This gradient-Hessian correlation contributes to an increase in Λ_G under gradient heterogeneity, while Λ_P remains relatively small.

Approximate explanation. If the loss function L is approximated in the region $\mathcal{R}_{\text{FT}}$ by a second-order Taylor expansion around the optimum $\boldsymbol{\theta}_* \in \mathcal{R}_{\text{FT}}$, where $\nabla L(\boldsymbol{\theta}_*)$ is close to $\mathbf{0}$, and the Hessian matrix is assumed to be block-diagonal, the following inequality approximately holds:

$$\|[\nabla L(\boldsymbol{\theta})]_b\|_2 \leq \|[\nabla^2 L(\boldsymbol{\theta}_*)]_b\|_2 \|\delta_{\boldsymbol{\theta}}\|_2,$$

where $\delta_{\boldsymbol{\theta}} = \boldsymbol{\theta} - \boldsymbol{\theta}_*$. This inequality suggests a positive correlation between the gradient norm and the Hessian matrix.

Support from prior studies. This gradient–Hessian correlation has been observed or assumed in prior work. For example, Zhang et al. (2024a); Jiang et al. (2024) demonstrated a relationship between $|\nabla L(\boldsymbol{\theta})i|$ and $|\nabla^2 L(\boldsymbol{\theta})i, i|$. The (L_0, L_1)-smoothness assumption (Zhang et al., 2020a) and its coordinate-wise generalization (Crawshaw et al., 2022) also embody this correlation.

4.4 Complexity bound

To analyze optimization algorithms, we define a complexity measure inspired by Carmon et al. (2020); Zhang et al. (2020a); Crawshaw et al. (2022). This measure reflects the number of parameter updates needed to achieve a sufficiently small gradient norm, with higher complexity indicating slower convergence.

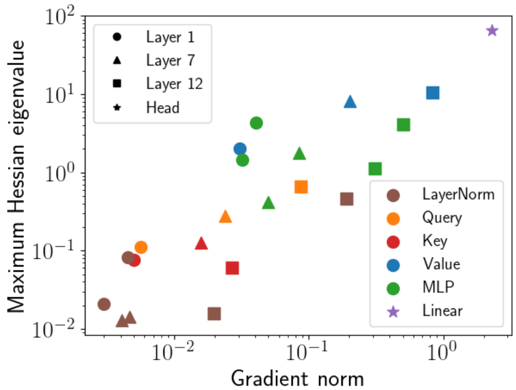


Figure 1: Correlation between the gradient norm and the maximum Hessian eigenvalue. Each point represents the mean value for a parameter block (pre-trained RoBERTa on RTE).

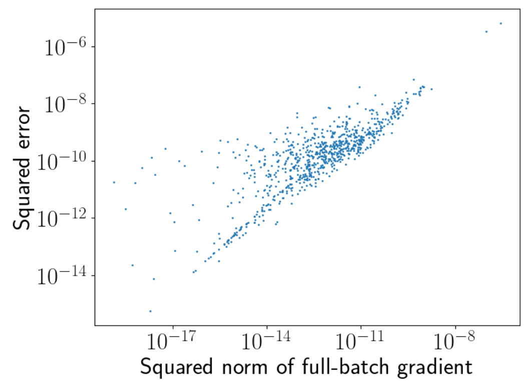


Figure 2: Correlation between the full-batch gradient and gradient error. Each point represents the absolute values of a coordinate (pre-trained RoBERTa on RTE).

Definition 4.6 (Iteration complexity)**.** We define the iteration complexity of a parameter sequence $\{\boldsymbol{\theta}_t\}_{t=0}^{\infty}$ for $\boldsymbol{\theta}_t \in \mathbb{R}^P$ with the loss function L and the norm $\|\cdot\|_q$:

$$\mathcal{T}_\varepsilon(\{\boldsymbol{\theta}_t\}_{t=0}^{\infty}, L, \|\cdot\|_q) \coloneqq \inf\{t \in \mathbb{N} \mid \mathcal{C}_\varepsilon(t)\},$$

where $\mathcal{C}_\varepsilon(t)$ is defined as

$$\text{deterministic:} \quad \|\nabla L(\boldsymbol{\theta}_t)\|_q \leq P^{1/q}\varepsilon, \qquad \text{stochastic:} \quad \mathbb{P}\big(\forall s \leq t,\ \|\nabla L(\boldsymbol{\theta}_s)\|_q \geq P^{1/q}\varepsilon\big) \leq \tfrac{1}{2}.$$

Compared with the complexity definitions in previous studies, we introduce a distinction between norms and a normalization term $P^{\frac{1}{q}}$ to ensure dimensional consistency across different norms.

Using this measure, we show the complexity bound in deterministic, namely full-batch, settings as follows. The parameter $\zeta_0 \in (0, 1)$ controls the range of learning rates.

Theorem 4.7 (Deterministic setting)**.** *Assume $\delta_D < \min(\Lambda_G, \Lambda_P)/3$. Then, the iteration complexities in deterministic settings are bounded as follows.*

For the gradient-based sequence, suppose that $\varepsilon < \frac{\Lambda_G^2}{\rho_H \sqrt{P}}$ holds and that learning rate at time t satisfies $\eta_t = \zeta_t \min(\frac{1}{\Lambda_G}, \frac{1}{\sqrt{\rho_H \|\nabla L(\boldsymbol{\theta}_t^{Grad})\|_2}})$, where $\zeta_t \in [\zeta_0, 1]$, we have

$$\mathcal{T}_\varepsilon(\{\boldsymbol{\theta}_t^{Grad}\}_{t=0}^{\infty}, L, \|\cdot\|_2) \leq \frac{6(L(\boldsymbol{\theta}_0) - L_*)}{P\varepsilon^2\zeta_0}\Lambda_G.$$

For the sign-based sequence, suppose that $\varepsilon < \frac{\Lambda_P^2}{\rho_H \sqrt{P}}$ holds and that the learning rate at time t satisfies $\eta_t = \zeta_t \min(\frac{\|\nabla L(\boldsymbol{\theta}_t^{Sign})\|_1}{\Lambda_P P}, \sqrt{\frac{\|\nabla L(\boldsymbol{\theta}_t^{Sign})\|_1}{\rho_H P^{3/2}}})$, where $\zeta_t \in [\zeta_0, 1]$, we have

$$\mathcal{T}_\varepsilon(\{\boldsymbol{\theta}_t^{Sign}\}_{t=0}^{\infty}, L, \|\cdot\|_1) \leq \frac{6(L(\boldsymbol{\theta}_0) - L_*)}{P\varepsilon^2\zeta_0}\Lambda_P.$$

The iteration complexity of the gradient-based and sign-based sequences is evaluated using the norms $\|\cdot\|_2$ and $\|\cdot\|_1$, respectively. This choice of norms is justified because they correspond to the dual norms that determine the steepest descent direction, as discussed in Section 3.2.

Gradient heterogeneity can increase the complexity of the gradient-based sequence. The theorem indicates that the iteration complexity of the gradient-based and sign-based sequences is characterized by Λ_G and Λ_P, respectively. As discussed earlier, when the gradient is heterogeneous, Λ_G can become large. Consequently, the iteration complexity of the gradient-based sequence may surpass that of the sign-based sequence under such conditions.

4.5 STOCHASTIC SETTING

In practice, the gradient is estimated using a mini-batch. We introduce the following assumptions about the noise, defined as the difference between the full-batch and mini-batch gradients.

Assumption 4.8 (Noise)**.** For all $\boldsymbol{\theta} \in \mathbb{R}^P$, there exist constants $\sigma_3, \sigma_2 \geq 0$ such that:

$$\mathbb{E}[\nabla \widehat{L}(\boldsymbol{\theta})] = \nabla L(\boldsymbol{\theta}), \tag{2}$$

$$\mathbb{E}[\|\nabla \widehat{L}(\boldsymbol{\theta}) - \nabla L(\boldsymbol{\theta})\|_2^3] \leq \sigma_3 \|\nabla L(\boldsymbol{\theta})\|_2^3, \tag{3}$$

and for all $i \in \{1, \ldots, P\}$,

$$\mathbb{E}[|\nabla \widehat{L}(\boldsymbol{\theta})_i - \nabla L(\boldsymbol{\theta})_i|^2] \leq \sigma_2 |\nabla L(\boldsymbol{\theta})_i|^2. \tag{4}$$

The assumption in Eq.(2) is standard in stochastic optimization (Bernstein et al., 2018). We introduce Eq.(3) to bound the third-order moment of the gradient noise norm and Eq.(4) to model its coordinate-wise correlation with the gradient. This correlation is supported by Figure 2 (additional settings in Appendix F.4). The coordinate-wise assumption is needed for analyzing errors in the gradient sign and block-wise gradient. Additionally, bounding the noise is a common practice in stochastic optimization (Crawshaw et al., 2022; Zhang et al., 2020a).

Using these assumptions, we establish the complexity bounds for the stochastic setting, where $\zeta_0 \in (0, 1)$ controls the range of learning rates as in the deterministic setting.

Theorem 4.9 (Stochastic setting)**.** *Assume $\delta_D < \min(\Lambda_G, \Lambda_P)/3$. Then, the iteration complexities in stochastic settings are bounded as follows.*

For the gradient-based sequence, suppose that $\varepsilon < \frac{(1+\sigma_2)^2 \Lambda_G^2}{4(1+\sigma_3)\rho_H \sqrt{P}}$ holds and that the learning rate at time t satisfies $\eta_t = \zeta_t \min(\frac{1}{(1+\sigma_2)\Lambda_G}, \frac{1}{2\sqrt{(1+\sigma_3)\rho_H \|\nabla L(\boldsymbol{\theta}_t^{Grad})\|_2}})$, where $\zeta_t \in [\zeta_0, 1]$, we have

$$\mathcal{T}_\varepsilon(\{\boldsymbol{\theta}_t^{Grad}\}_{t=0}^\infty, L, \|\cdot\|_2) \leq \frac{12(1+\sigma_2)(L(\boldsymbol{\theta}_0) - L_*)}{P\varepsilon^2 \zeta_0} \Lambda_G.$$

For the sign-based sequence, suppose that $\varepsilon < \frac{\Lambda_P^2}{\rho_H \sqrt{P}}$ and $\sigma_2 \leq \frac{1}{24}$ hold and that the learning rate at time t satisfies $\eta_t = \zeta_t \min(\frac{\|\nabla L(\boldsymbol{\theta}_t^{Sign})\|_1}{\Lambda_P P}, \sqrt{\frac{\|\nabla L(\boldsymbol{\theta}_t^{Sign})\|_1}{\rho_H P^{3/2}}})$, where $\zeta_t \in [\zeta_0, 1]$, we have

$$\mathcal{T}_\varepsilon(\{\boldsymbol{\theta}_t^{Sign}\}_{t=0}^\infty, L, \|\cdot\|_1) \leq \frac{12(1+24\sigma_2)(L(\boldsymbol{\theta}_0) - L_*)}{P\varepsilon^2 \zeta_0} \Lambda_P.$$

This theorem shows that the dependence on the noise is the same for the both sequences up to a constant, so the difference in noise dependence may be minor. Therefore, the performance gap is more likely due to the difference between Λ_G and Λ_P, as in the deterministic setting.

4.6 OPTIMIZATION OF TRANSFORMERS

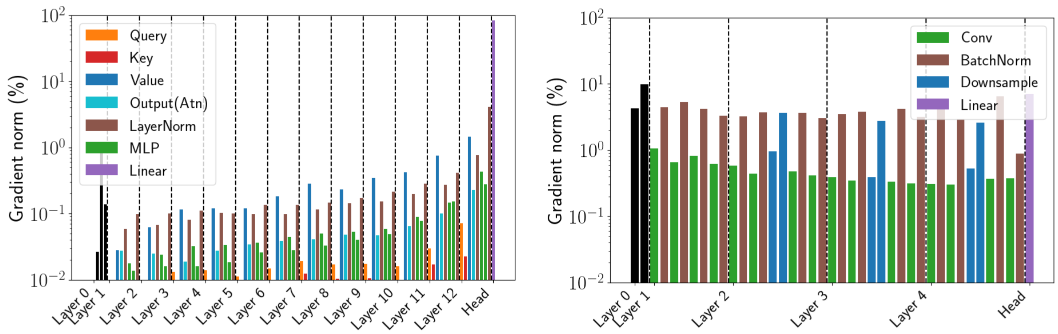
Transformers show much greater parameter heterogeneity than other models (Zhang et al., 2024a; Cui and Wang, 2024). Theorems 4.7 and 4.9, suggest that gradient heterogeneity is a key factor in the performance gap between Adam and SGD in transformers. Here, we discuss the role of layer normalization in transformers.

Post-LN and Pre-LN. Transformers integrate residual connections and layer normalization with multi-head attention and feed-forward networks. The two main transformer architectures are post-layer normalization (Post-LN), where the residual connection is followed by the layer normalization, and pre-layer normalization (Pre-LN), where the layer normalization precedes the residual connection. Pre-LN is known for greater stability (Wang et al., 2019b; Xiong et al., 2020).

Jacobian of transformers. The Jacobians of Pre-LN and Post-LN transformer layers are:

$$\boldsymbol{J}_{\text{Pre-LN}} = \boldsymbol{J}_{\text{FFN}} \left(\boldsymbol{J}_{\text{LN}} + \boldsymbol{I}_{nd}\right) \boldsymbol{J}_{\text{ATT}} \left(\boldsymbol{J}_{\text{LN}} + \boldsymbol{I}_{nd}\right), \tag{5}$$

$$\boldsymbol{J}_{\text{Post-LN}} = \boldsymbol{J}_{\text{LN}} \left(\boldsymbol{J}_{\text{FFN}} + \boldsymbol{I}_{nd}\right) \boldsymbol{J}_{\text{LN}} \left(\boldsymbol{J}_{\text{ATT}} + \boldsymbol{I}_{nd}\right). \tag{6}$$



(a) RoBERTa on RTE

(b) ResNet18 on Flowers102

Figure 3: **Transformers exhibit large gradient heterogeneity.** Gradient norms of individual parameters in pre-trained models.

where $\boldsymbol{J}_{\text{ATT}}$ and $\boldsymbol{J}_{\text{FFN}}$ denote the Jacobians of the self-attention and feed-forward network modules, respectively. For simplicity, the evaluation points of the Jacobians are omitted. The Jacobian of the layer normalization is represented by $\boldsymbol{J}_{\text{LN}}$, calculated for an input $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ as:

$$\boldsymbol{J}_{\text{LN}}(\boldsymbol{X}) = \text{blockdiag}(\{\boldsymbol{L}_i(\boldsymbol{X})\}_{i=1}^{n}), \tag{7}$$

where each block $\boldsymbol{L}_i \in \mathbb{R}^{d \times d}$ is defined as:

$$\boldsymbol{L}_i(\boldsymbol{X}) \coloneqq \frac{\sqrt{d}}{\|\widetilde{\boldsymbol{X}_{i,:}}\|_2}\left(\boldsymbol{I}_d - \frac{\widetilde{\boldsymbol{X}_{i,:}}\widetilde{\boldsymbol{X}_{i,:}}^\top}{\|\widetilde{\boldsymbol{X}_{i,:}}\|_2^2}\right)\left(\boldsymbol{I}_d - \frac{\mathbf{1}\mathbf{1}^\top}{d}\right),$$

and $\widetilde{\boldsymbol{X}_{i,:}} \coloneqq \boldsymbol{X}_{i,:}(\boldsymbol{I}_d - \frac{\mathbf{1}\mathbf{1}^\top}{d})$. These derivations are provided in Appendix D.

Greater gradient heterogeneity in Post-LN. Equation (7) shows that the Jacobian of layer normalization, $\boldsymbol{J}_{\text{LN}}$, depends on the input and varies across layers. As seen in Eqs. (5) and (6), Post-LN is more directly affected by $\boldsymbol{J}_{\text{LN}}$, because $\boldsymbol{J}_{\text{LN}}$ appears multiplicatively and thus its variations scale the entire Jacobian. In contrast, in Pre-LN, $\boldsymbol{J}_{\text{LN}}$ appears additively with the identity matrix, reducing its influence and leading to more uniform gradients across layers. Further discussion, including the role of the attention mechanism, is provided in Appendix H.

5 Numerical evaluation

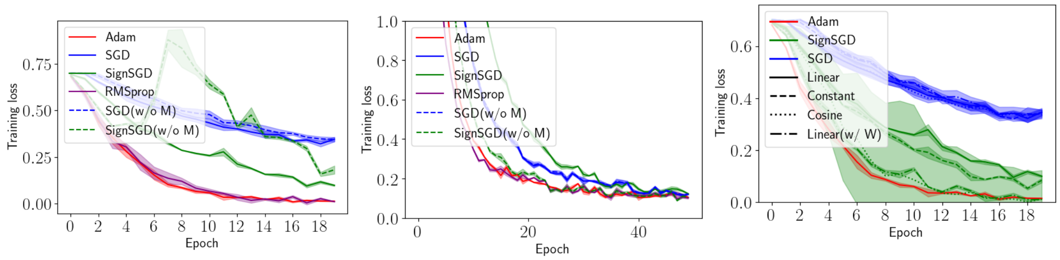
We numerically validate the following claims. Detailed experimental settings can be found in Appendices E and F.

- Gradient heterogeneity is pronounced in transformers and is influenced by the position of layer normalization (Section 5.2).
- SGD encounters greater difficulty in optimization under gradient heterogeneity compared with adaptive optimizers such as Adam (Section 5.3).

5.1 Experimental setup

Datasets and models. We used a total of nine datasets and three pre-trained models obtained from public sources. For NLP tasks, we used four datasets from SuperGLUE (Wang et al., 2019a) (BoolQ, CB, RTE, and WiC) and three datasets from GLUE (Wang et al., 2018) (CoLA, MRPC, and SST-2) with RoBERTa-Base model (Liu et al., 2019). For vision tasks, we used the Flowers102 (Nilsback and Zisserman, 2008) and FGVC-Aircraft (Aircraft) (Maji et al., 2013) datasets with ViT-Base (Dosovitskiy et al., 2021) and ResNet18 (He et al., 2016) models.

Training. Following Kunstner et al. (2023), learning rates were tuned via grid search based on the training loss. Gradient clipping was applied, and the learning rate schedule was fixed for each domain. All models were fine-tuned on each dataset starting from pre-trained weights.



(a) RoBERTa

(b) ResNet18

(c) RoBERTa with learning rate schedules

Figure 4: **RoBERTa is difficult to optimize using SGD.** Lines show training losses, with shaded areas representing interquartile ranges. “w/ W” indicates warmup. (a) and (c): RTE; (b): Flowers102.

5.2 Gradient heterogeneity

Figure 3 shows that RoBERTa exhibits greater gradient heterogeneity than ResNet18 (with ViT falling in between; see Figure 5). This suggests that transformers, and the Post-LN architecture used in RoBERTa, tend to induce more pronounced gradient heterogeneity (Section 4.6).

Additionally, the gradients of the value weight matrix in RoBERTa are consistently larger than those of the query and key, consistent with the findings of Noci et al. (2022) (Appendix H).

Effect of layer normalization. Table 2 shows Gini coefficients for different normalization placements of RoBERTa on RTE. Post-LN shows the highest heterogeneity, followed by Pre-LN and No-LN, consistent with our analysis (Section 4.6). Pre-trained weights are only available for Post-LN.

Table 2: **Post-LN increases gradient heterogeneity.** Higher Gini coefficient indicates greater heterogeneity. “No-LN” means no layer normalization.

Norm Type	Initialization	Gini Coeff.
No-LN	Scratch	0.867 ± 0.006
Pre-LN	Scratch	0.880 ± 0.004
Post-LN	Scratch	0.941 ± 0.012
Post-LN	Pre-trained	0.944 ± 0.005

5.3 Training curves

Limitations of SGD under gradient heterogeneity. As shown in Figure 4(a) and (b), all optimizers successfully train ResNet (and ViT; see Figure 10), but SGD fails to optimize RoBERTa, highlighting the challenge caused by gradient heterogeneity. This aligns with our theoretical analysis in Theorems 4.7 and 4.9. Additionally, the final training losses are similar for SGD and SignSGD (with or without momentum), and Adam performs similarly to RMSProp. This suggests that adaptive learning rates are the primary cause of the performance gap (Kunstner et al., 2023). Note that the RTE dataset with two classes is almost balanced, so the Adam–SGD performance gap cannot be explained by heavy-tailed class imbalance (Kunstner et al., 2024).

Effectiveness of learning rate schedules. In Figure 4(c), we compare training performance of RoBERTa under various learning rate schedules. While scheduling has little effect on SGD, SignSGD benefits substantially from appropriate schedules, achieving performance comparable to Adam when combined with a linear schedule and warmup. These results suggest that it is not the lack of scheduling, but rather gradient heterogeneity, that limits SGD performance, and they demonstrate that SignSGD can match Adam’s performance when properly scheduled.

6 Conclusion

We identify gradient heterogeneity as a key factor underlying the performance gap between Adam and SGD in transformers, supported by derived upper bounds on iteration complexity based on norms in steepest descent. Our analysis shows that gradient heterogeneity is particularly pronounced in Post-LN architectures.Empirical results support these findings: transformer architectures and Post-LN design induce gradient heterogeneity, which significantly impedes SGD, while SignSGD with appropriate scheduling matches Adam’s performance.

References

Kwangjun Ahn, Xiang Cheng, Minhak Song, Chulhee Yun, Ali Jadbabaie, and Suvrit Sra. Linear attention is (maybe) all you need (to understand transformer optimization). In *International Conference on Learning Representations*, 2024.

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. Etc: Encoding long and structured inputs in transformers. In *The 2020 Conference on Empirical Methods in Natural Language Processing*, 2020.

Lukas Balles and Philipp Hennig. Dissecting adam: The sign, magnitude and variance of stochastic gradients. In *International Conference on Machine Learning*, volume 80. PMLR, 2018.

Han Bao, Ryuichiro Hataya, and Ryo Karakida. Self-attention networks localize when qk-eigenspectrum concentrates. In *International Conference on Machine Learning*, 2024.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*. PMLR, 2018.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *Mathematical Programming*, 184(1), 2020.

Guanzheng Chen, Fangyu Liu, Zaiqiao Meng, and Shangsong Liang. Revisiting parameter-efficient tuning: Are we really there yet? In *The 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2022.

Lizhang Chen, Bo Liu, Kaizhao Liang, and qiang liu. Lion secretly solves a constrained optimization: As lyapunov predicts. In *International Conference on Learning Representations*, 2024a.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, et al. Symbolic discovery of optimization algorithms. *Advances in Neural Information Processing Systems*, 36, 2024b.

D Choi. On empirical comparisons of optimizers for deep learning. *arXiv preprint arXiv:1910.05446*, 2019.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *The 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019.

Kevin Clark. What does bert look at? an analysis of bert’ s attention. *arXiv preprint arXiv:1906.04341*, 2019.

Ronan Collobert. *Large scale machine learning*. PhD thesis, Université de Paris VI, 2004.

Michael Crawshaw, Mingrui Liu, Francesco Orabona, Wei Zhang, and Zhenxun Zhuang. Robustness to unbounded smoothness of generalized signsgd. *Advances in Neural Information Processing Systems*, 35, 2022.

Wanyun Cui and Qianle Wang. Cherry on top: Parameter heterogeneity and quantization in large language models. *arXiv preprint arXiv:2404.02837*, 2024.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse. In *Sinn und Bedeutung*, volume 23, 2019.

Bill Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Third international workshop on paraphrasing (IWP2005)*, 2005.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE conference on computer vision and pattern recognition*, 2016.

Ruining He, Anirudh Ravula, Bhargav Kanagal, and Joshua Ainslie. Realformer: Transformer likes residual attention. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021.

Nam Hyeon-Woo, Kim Yu-Ji, Byeongho Heo, Dongyoon Han, Seong Joon Oh, and Tae-Hyun Oh. Scratching visual transformer's back with uniform attention. In *The IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Kaiqi Jiang, Dhruv Malik, and Yuanzhi Li. How does adaptive optimization impact local neural network geometry? *Advances in Neural Information Processing Systems*, 36, 2024.

Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Frederik Kunstner, Jacques Chen, Jonathan Wilder Lavington, and Mark Schmidt. Noise is not the main factor behind the gap between sgd and adam on transformers, but sign descent might be. In *International Conference on Learning Representations*, 2023.

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

Bingrui Li, Wei Huang, Andi Han, Zhanpeng Zhou, Taiji Suzuki, Jun Zhu, and Jianfei Chen. On the optimization and generalization of two-layer transformers with sign gradient descent. In *International Conference on Learning Representations*, 2025.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Lucas Maes, Tianyue H Zhang, Alexia Jolicoeur-Martineau, Ioannis Mitliagkas, Damien Scieur, Simon Lacoste-Julien, and Charles Guille-Escuret. Understanding adam requires better rotation dependent assumptions. *arXiv preprint arXiv:2410.19964*, 2024.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*. PMLR, 2015.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*. IEEE, 2008.

Lorenzo Noci, Sotiris Anagnostidis, Luca Biggio, Antonio Orvieto, Sidak Pal Singh, and Aurelien Lucchi. Signal propagation in transformers: Theoretical perspectives and the role of rank collapse. *Advances in Neural Information Processing Systems*, 35, 2022.

Weronika Ormaniec, Felix Dangel, and Sidak Pal Singh. What does it mean to be a transformer? insights from a theoretical hessian analysis. In *International Conference on Learning Representations*, 2025.

Yan Pan and Yuanzhi Li. Toward understanding why adam converges faster than SGD for transformers. In *OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop)*, 2022.

Namuk Park and Songkuk Kim. How do vision transformers work? In *International Conference on Learning Representations*, 2022.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

Mohammad Taher Pilehvar and Jose Camacho-Collados. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. In *The 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge Belongie. Objects in context. In *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007.

Elan Rosenfeld and Andrej Risteski. Outliers with opposing signals have an outsized effect on neural network optimization. In *International Conference on Learning Representations*, 2024.

Robin M Schmidt, Frank Schneider, and Philipp Hennig. Descending through a crowded valley-benchmarking deep learning optimizers. In *International Conference on Machine Learning*. PMLR, 2021.

Han Shi, JIAHUI GAO, Hang Xu, Xiaodan Liang, Zhenguo Li, Lingpeng Kong, Stephen MS Lee, and James Kwok. Revisiting over-smoothing in bert from the perspective of graph. In *International Conference on Learning Representations*, 2022.

Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International journal of computer vision*, 81, 2009.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *The 2013 conference on Empirical Methods in Natural Language Processing*, 2013.

Tijmen Tieleman and G Hinton. Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. *Technical report*, 2017.

Akiyoshi Tomihari and Issei Sato. Understanding linear probing then fine-tuning language models from ntk perspective. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Antonio Torralba. Contextual priming for object detection. *International journal of computer vision*, 53, 2003.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2018.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in Neural Information Processing Systems*, 32, 2019a.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. Learning deep transformer models for machine translation. In *The 57th Annual Meeting of the Association for Computational Linguistics*, 2019b.

Shulun Wang, Feng Liu, and Bin Liu. Escaping the gradient vanishing: Periodic alternatives of softmax in attention mechanism. *IEEE Access*, 9, 2021.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7, 2019.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *The 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, 2020.

Xinyi Wu, Amir Ajorlou, Yifei Wang, Stefanie Jegelka, and Ali Jadbabaie. On the role of attention masks and layernorm in transformers. In *Advances in Neural Information Processing Systems*, 2024.

Shuo Xie and Zhiyuan Li. Implicit bias of adamw: ℓ_∞-norm constrained optimization. In *International Conference on Machine Learning*. PMLR, 2024.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*. PMLR, 2020.

Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE international conference on big data (Big data)*. IEEE, 2020.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33, 2020.

Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Joshua M. Susskind. Stabilizing transformer training by preventing attention entropy collapse. In *International Conference on Machine Learning*, volume 202. PMLR, 2023.

Huishuai Zhang, Caiming Xiong, James Bradbury, and Richard Socher. Block-diagonal hessian-free optimization for training neural networks. *arXiv preprint arXiv:1712.07296*, 2017.

Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2020a.

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, 2020b.

Yushun Zhang, Congliang Chen, Naichen Shi, Ruoyu Sun, and Zhi-Quan Luo. Adam can converge without any modification on update rules. *Advances in Neural Information Processing Systems*, 35, 2022.

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, 2024a.

Yushun Zhang, Congliang Chen, Ziniu Li, Tian Ding, Chenwei Wu, Diederik P. Kingma, Yinyu Ye, Zhi-Quan Luo, and Ruoyu Sun. Adam-mini: Use fewer learning rates to gain more, 2024b.

Rosie Zhao, Depen Morwani, David Brandfonbrener, Nikhil Vyas, and Sham M. Kakade. Deconstructing what makes a good optimizer for autoregressive language models. In *International Conference on Learning Representations*, 2025a.

Rosie Zhao, Depen Morwani, David Brandfonbrener, Nikhil Vyas, and Sham M. Kakade. Deconstructing what makes a good optimizer for autoregressive language models. In *International Conference on Learning Representations*, 2025b.

Table of contents

A Extended related work

Adam in deep learning. Adam (Kingma and Ba, 2015) is a widely used optimization algorithm in deep learning, known for its well-established convergence properties (Zhang et al., 2022). However, the reasons for its superior performance are not yet fully understood. Jiang et al. (2024) empirically observed that Adam tends to converge to parameter regions with uniform diagonal elements in the Hessian, supported by theoretical analysis based on two-layer linear models. Rosenfeld and Risteski (2024) argued that the ability of Adam to handle outliers in features is a critical factor in its effectiveness. Additionally, Kunstner et al. (2024) attributed the performance of Adam in language models to its ability to manage heavy-tailed class imbalance.

Optimization challenges in transformers. A key aspect of transformer optimization is the notable superiority of Adam over SGD. Zhang et al. (2020b) attributed this to the heavy-tailed gradient noise, but Kunstner et al. (2023) later challenged this, arguing that the superior performance of Adam can be attributed to sign-based characteristics rather than gradient noise, supported by full-batch experiments. Li et al. (2025) demonstrated the similarity between Adam and SignSGD in optimization and generalization, and provided a theoretical analysis of SignSGD. Pan and Li (2022) proposed directional sharpness and show that Adam achieves lower values in transformers. Ahn et al. (2024) demonstrated that linear transformers exhibit similar optimization behaviors to standard transformers. Zhang et al. (2024a) revealed that the Hessian spectrum of the loss function with transformers is heterogeneous and suggested that this is one cause of the Adam-SGD performance gap. This heterogeneity was later confirmed by Ormaniec et al. (2025), who derived the Hessian of transformers explicitly.

Sign-based optimization and variants. SignSGD, also known as sign descent (Balles and Hennig, 2018), is an optimization method that is computationally efficient and memory-saving, making it suited for distributed training (Bernstein et al., 2018). Through program search, a sign-based optimization algorithm called Lion (evolved sign momentum) was discovered (Chen et al., 2024b), and its effectiveness was shown by Chen et al. (2024a). Adam can be interpreted as a variance-adapted variant of SignSGD. For example, Xie and Li (2024) analyzed the convergence property of Adam by using this property. Similarly, Zhao et al. (2025a) found that sign-based optimizers restore the stability and performance of Adam and proposed using adaptive learning rates for each layer. Additionally, Zhang et al. (2024b) showed that adaptive learning rates do not need to be computed at a coordinate-wise level but can be applied at the level of parameter blocks.

Transformer architecture and layer normalization. The original transformer architecture (Vaswani et al., 2017), referred to as Post-LN, applies layer normalization after the residual connection. In contrast, the Pre-LN architecture places layer normalization before the residual connection. Wang et al. (2019b) demonstrated that Post-LN transformers are difficult to train when the number of layers is large, a finding later theoretically confirmed by Xiong et al. (2020) using mean field theory. Other architectures such as Reformer (He et al., 2021) were also introduced. Shi et al. (2022) showed that a large standard deviation in layer normalization leads to rank collapse in Post-LN transformers. Furthermore, Wu et al. (2024) observed that sparse masked attention mitigates rank collapse in the absence of layer normalization and that layer normalization induces equilibria ranging from rank one to full rank.

Attention sparsity. Sparse attention mechanisms have been proposed to reduce the computational costs of transformers. For example, ETC (Ainslie et al., 2020) introduces efficient sparse attention, and Zaheer et al. (2020) proposed BigBird, which they theoretically demonstrated to be as expressive as full attention. These sparse attention mechanisms are widely used in language models with large context windows, such as Longformer (Beltagy et al., 2020) and Mistral 7B (Jiang et al., 2023). In NLP, Clark (2019) found that attention of pre-trained BERT focuses on specific tokens. In vision, Hyeon-Woo et al. (2023) showed that while uniform attention is challenging to learn with the softmax function, ViT successfully learns uniform attention, which is key to its success. Additionally, Zhai et al. (2023) suggested that low attention entropy contributes to training instability in transformers, a phenomenon they termed *entropy collapse*. Furthermore, Bao et al. (2024) demonstrated that a small eigenspectrum variance of query and key matrices leads to localized attention and mitigates both rank and entropy collapse.

A.1 Comparison with Zhang et al. (2024a)

Zhang et al. (2024a) experimentally observed significant differences in the Hessian spectra of transformers and attributed these differences to the Adam–SGD performance gap. They also provided "Initial Theoretical Results", but as the term "Initial" suggests, their explanation does not fully capture the optimization of transformers because:

- They analyze only a single update step.
- They focus solely on convex quadratic problems with a positive definite matrix.

While these observations are insightful, the above limitations are far from the settings relevant to deep learning optimization in Transformers.

We improve upon their theoretical explanation by:

- Introducing iteration complexity to capture long-term training dynamics.
- Considering more realistic deep learning functions under appropriate assumptions.

Overall, we do not challenge Zhang et al. (2024a) but rather supplement their findings by introducing the concept of gradient heterogeneity. We theoretically argue that the Adam–SGD performance gap arises from the combined effect of Hessian and gradient heterogeneity.

B Abbreviation and notation

Table 3 and Table 4 show our abbreviations and notations, respectively.

Table 3: Table of abbreviations.

Abbreviation	Definition
natural language processing	NLP
stochastic gradient descent	SGD
post-layer normalization	Post-LN
pre-layer normalization	Pre-LN

Table 4: Table of notations.

Variable	Definition
a_k	k-th element of vector $\boldsymbol{a}$
$\boldsymbol{A}_{k,:}, \boldsymbol{A}_{:,j}, A_{k,j}$	k-th row, j-th column, and (k, j)-th element of matrix $\boldsymbol{A}$
$[\boldsymbol{A}]_b, [\boldsymbol{a}]_b$	b-th block of matrix $\boldsymbol{A}$ and vector $\boldsymbol{a}$
B	number of blocks in parameters
$\mathbf{1}_a$	all-ones vector of size a
$\boldsymbol{I}_a$	identity matrix of size $a \times a$
$\mathrm{vec}(\cdot), \mathrm{blockdiag}(\cdot)$	row-wise vectorization, block diagonal matrix
$\otimes$	Kronecker product
C, N	number of classes and training samples
P, P_b	dimensions of model parameters, and b-th block of parameters
$\mathcal{X}$	sample space
$\boldsymbol{\theta}$	model parameter
$\boldsymbol{f}(\cdot), \boldsymbol{\phi}(\cdot)$	model, feature extractor
$\boldsymbol{V}, \boldsymbol{b}$	weight matrix and bias of the linear head
h, d	dimensions of features and tokens
$\boldsymbol{x}^{(i)}, y^{(i)}$	i-th training sample and label
$L(\cdot)$	training loss
$\widehat{L}(\cdot)$	mini-batch loss
η_t	learning rate at iteration t
$\ell(\cdot, \cdot)$	cross entropy loss function
$\mathrm{softmax}(\cdot), \mathrm{sign}(\cdot)$	softmax and sign function
$\mathcal{R}_{\mathrm{FT}}$	parameter region of fine-tuning
$L_* = L(\boldsymbol{\theta}_*)$	local minimum of training loss
ρ_H	Lipschitz constant of the Hessian matrix
L_D	block-diagonal approximation of the Hessian matrix
δ_D	upper bound of the approximation of L_D
σ_2, σ_3	constants in the upper bound of the gradient error
$\mathrm{SA}(\cdot)$	single-head self-attention
$\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V$	query, key, and value weight matrix
d_k, d_v	dimensions of key/query and value

C PROOF

C.1 LEMMA

Lemma C.1. *Under assumption 4.3, for any $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^P$, the following inequality holds:*

$$L(\boldsymbol{\theta}') - L(\boldsymbol{\theta}) \le \nabla L(\boldsymbol{\theta})^\top(\boldsymbol{\theta}' - \boldsymbol{\theta}) + \frac{1}{2}(\boldsymbol{\theta}' - \boldsymbol{\theta})^\top \nabla^2 L(\boldsymbol{\theta})(\boldsymbol{\theta}' - \boldsymbol{\theta}) + \frac{\rho_H}{6}\|\boldsymbol{\theta}' - \boldsymbol{\theta}\|_2^3.$$

Proof. Define $\boldsymbol{\nu}(\alpha) \coloneqq \boldsymbol{\theta} + \alpha(\boldsymbol{\theta}' - \boldsymbol{\theta})$. Then we have:

$$\begin{aligned}
&(\nabla L(\boldsymbol{\theta}') - \nabla L(\boldsymbol{\theta}))^\top(\boldsymbol{\theta}' - \boldsymbol{\theta}) \\
&= \int_0^1 (\boldsymbol{\theta}' - \boldsymbol{\theta})^\top \nabla^2 L(\boldsymbol{\nu}(\alpha))(\boldsymbol{\theta}' - \boldsymbol{\theta}) d\alpha \\
&= (\boldsymbol{\theta}' - \boldsymbol{\theta})^\top \nabla^2 L(\boldsymbol{\theta})(\boldsymbol{\theta}' - \boldsymbol{\theta}) + \int_0^1 (\boldsymbol{\theta}' - \boldsymbol{\theta})^\top(\nabla^2 L(\boldsymbol{\nu}(\alpha)) - \nabla^2 L(\boldsymbol{\theta}))(\boldsymbol{\theta}' - \boldsymbol{\theta}) d\alpha \\
&\le (\boldsymbol{\theta}' - \boldsymbol{\theta})^\top \nabla^2 L(\boldsymbol{\theta})(\boldsymbol{\theta}' - \boldsymbol{\theta}) + \int_0^1 \|\nabla^2 L(\boldsymbol{\nu}(\alpha)) - \nabla^2 L(\boldsymbol{\theta})\|_2 \|\boldsymbol{\theta}' - \boldsymbol{\theta}\|_2^2 d\alpha \\
&\le (\boldsymbol{\theta}' - \boldsymbol{\theta})^\top \nabla^2 L(\boldsymbol{\theta})(\boldsymbol{\theta}' - \boldsymbol{\theta}) + \int_0^1 \rho_H \alpha \|\boldsymbol{\theta}' - \boldsymbol{\theta}\|_2^3 d\alpha \quad \text{(Because Hessian matrix is } \rho_H\text{-Lipschitz continuous)} \\
&= (\boldsymbol{\theta}' - \boldsymbol{\theta})^\top \nabla^2 L(\boldsymbol{\theta})(\boldsymbol{\theta}' - \boldsymbol{\theta}) + \frac{\rho_H}{2}\|\boldsymbol{\theta}' - \boldsymbol{\theta}\|_2^3. \qquad (8)
\end{aligned}$$

Using this inequality, we obtain:

$$\begin{aligned}
&L(\boldsymbol{\theta}') - L(\boldsymbol{\theta}) \\
&= \int_0^1 \nabla L(\boldsymbol{\nu}(\alpha))^\top(\boldsymbol{\theta}' - \boldsymbol{\theta}) d\alpha \\
&= \nabla L(\boldsymbol{\theta})^\top(\boldsymbol{\theta}' - \boldsymbol{\theta}) + \int_0^1 (\nabla L(\boldsymbol{\nu}(\alpha)) - \nabla L(\boldsymbol{\theta}))^\top(\boldsymbol{\theta}' - \boldsymbol{\theta}) d\alpha \\
&= \nabla L(\boldsymbol{\theta})^\top(\boldsymbol{\theta}' - \boldsymbol{\theta}) + \int_0^1 (\nabla L(\boldsymbol{\nu}(\alpha)) - \nabla L(\boldsymbol{\theta}))^\top \frac{1}{\alpha}(\boldsymbol{\nu}(\alpha) - \boldsymbol{\theta}) d\alpha \\
&\le \nabla L(\boldsymbol{\theta})^\top(\boldsymbol{\theta}' - \boldsymbol{\theta}) + \int_0^1 \frac{1}{\alpha}\left((\boldsymbol{\nu}(\alpha) - \boldsymbol{\theta})^\top \nabla^2 L(\boldsymbol{\theta})(\boldsymbol{\nu}(\alpha) - \boldsymbol{\theta}) + \frac{\rho_H}{2}\|\boldsymbol{\nu}(\alpha) - \boldsymbol{\theta}\|_2^3\right) d\alpha \quad \text{(From } Eq.\text{(8))} \\
&= \nabla L(\boldsymbol{\theta})^\top(\boldsymbol{\theta}' - \boldsymbol{\theta}) + \int_0^1 \left((\boldsymbol{\theta}' - \boldsymbol{\theta})^\top \nabla^2 L(\boldsymbol{\theta})(\boldsymbol{\theta}' - \boldsymbol{\theta})\alpha + \frac{\rho_H}{2}\|\boldsymbol{\theta}' - \boldsymbol{\theta}\|_2^3 \alpha^2\right) d\alpha \\
&= \nabla L(\boldsymbol{\theta})^\top(\boldsymbol{\theta}' - \boldsymbol{\theta}) + \frac{1}{2}(\boldsymbol{\theta}' - \boldsymbol{\theta})^\top \nabla^2 L(\boldsymbol{\theta})(\boldsymbol{\theta}' - \boldsymbol{\theta}) + \frac{\rho_H}{6}\|\boldsymbol{\theta}' - \boldsymbol{\theta}\|_2^3.
\end{aligned}$$

□

Lemma C.2. *For any $a, b \ge 0$, the following inequality holds:*

$$(a+b)^3 \le 4(a^3 + b^3).$$

Proof. Calculating the difference between the right-hand and left-hand side, we obtain:

$$\begin{aligned}
4(a^3 + b^3) - (a+b)^3 &= 4(a^3 + b^3) - (a^3 + 3a^2 b + 3ab^2 + b^3) \\
&= 3(a^3 + b^3) - 3a^2 b - 3ab^2 \\
&= 3(a+b)(a-b)^2 \ge 0.
\end{aligned}$$

□

C.2 Proof of Theorem 4.7

Theorem 4.7 is restated. *Assume $\delta_D < \min(\Lambda_G, \Lambda_P)/3$. Then, the iteration complexities in deterministic settings are bounded as follows.*

For the gradient-based sequence, suppose that $\varepsilon < \frac{\Lambda_G^2}{\rho_H\sqrt{P}}$ holds and that learning rate at time t satisfies $\eta_t = \zeta_t \min(\frac{1}{\Lambda_G}, \frac{1}{\sqrt{\rho_H \|\nabla L(\boldsymbol{\theta}_t^{Grad})\|_2}})$, where $\zeta_t \in [\zeta_0, 1]$, we have

$$\mathcal{T}_\varepsilon(\{\boldsymbol{\theta}_t^{Grad}\}_{t=0}^\infty, L, \|\cdot\|_2) \leq \frac{6(L(\boldsymbol{\theta}_0) - L_*)}{P\varepsilon^2\zeta_0}\Lambda_G.$$

For the sign-based sequence, suppose that $\varepsilon < \frac{\Lambda_P^2}{\rho_H\sqrt{P}}$ holds and that the learning rate at time t satisfies $\eta_t = \zeta_t \min(\frac{\|\nabla L(\boldsymbol{\theta}_t^{Sign})\|_1}{\Lambda_P P}, \sqrt{\frac{\|\nabla L(\boldsymbol{\theta}_t^{Sign})\|_1}{\rho_H P^{3/2}}})$, where $\zeta_t \in [\zeta_0, 1]$, we have

$$\mathcal{T}_\varepsilon(\{\boldsymbol{\theta}_t^{Sign}\}_{t=0}^\infty, L, \|\cdot\|_1) \leq \frac{6(L(\boldsymbol{\theta}_0) - L_*)}{P\varepsilon^2\zeta_0}\Lambda_P.$$

Proof of gradient-based sequence. The update rule of the gradient-based sequence in deterministic setting is $\boldsymbol{\theta}_{t+1}^{\text{Grad}} = \boldsymbol{\theta}_t^{\text{Grad}} - \eta_t \nabla L(\boldsymbol{\theta}_t^{\text{Grad}})$. Thus, we obtain:

$$\begin{aligned}
& L(\boldsymbol{\theta}_{t+1}^{\text{Grad}}) - L(\boldsymbol{\theta}_t^{\text{Grad}}) \\
\leq\ & \nabla L(\boldsymbol{\theta}_t^{\text{Grad}})^\top (\boldsymbol{\theta}_{t+1}^{\text{Grad}} - \boldsymbol{\theta}_t^{\text{Grad}}) + \frac{1}{2}(\boldsymbol{\theta}_{t+1}^{\text{Grad}} - \boldsymbol{\theta}_t^{\text{Grad}})^\top \nabla^2 L(\boldsymbol{\theta}_t^{\text{Grad}})(\boldsymbol{\theta}_{t+1}^{\text{Grad}} - \boldsymbol{\theta}_t^{\text{Grad}}) + \frac{\rho_H}{6}\|\boldsymbol{\theta}_{t+1}^{\text{Grad}} - \boldsymbol{\theta}_t^{\text{Grad}}\|_2^3 \\
& \quad \text{(From } Lemma\ C.1) \\
=\ & -\eta_t \|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 + \frac{\eta_t^2}{2}\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})^\top \nabla^2 L(\boldsymbol{\theta}_t^{\text{Grad}}) \nabla L(\boldsymbol{\theta}_t^{\text{Grad}}) + \eta_t^3 \frac{\rho_H}{6}\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^3 \\
=\ & -\eta_t \|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 + \frac{\eta_t^2}{2}\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})^\top \nabla^2 L_D(\boldsymbol{\theta}_t^{\text{Grad}}) \nabla L(\boldsymbol{\theta}_t^{\text{Grad}}) \\
& + \frac{\eta_t^2}{2}\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})^\top (\nabla^2 L(\boldsymbol{\theta}_t^{\text{Grad}}) - \nabla^2 L_D(\boldsymbol{\theta}_t^{\text{Grad}})) \nabla L(\boldsymbol{\theta}_t^{\text{Grad}}) + \eta_t^3 \frac{\rho_H}{6}\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^3 \\
=\ & -\eta_t \|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 + \frac{\eta_t^2}{2}\sum_b [\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})]_b^\top [\nabla^2 L(\boldsymbol{\theta}_t^{\text{Grad}})]_b [\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})]_b \\
& + \frac{\eta_t^2}{2}\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})^\top (\nabla^2 L(\boldsymbol{\theta}_t^{\text{Grad}}) - \nabla^2 L_D(\boldsymbol{\theta}_t^{\text{Grad}})) \nabla L(\boldsymbol{\theta}_t^{\text{Grad}}) + \eta_t^3 \frac{\rho_H}{6}\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^3 \\
\leq\ & -\eta_t \|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 + \frac{\eta_t^2}{2}\sum_b \|[\nabla^2 L(\boldsymbol{\theta}_t^{\text{Grad}})]_b\|_2 \|[\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})]_b\|_2^2 \\
& + \frac{\eta_t^2}{2}\|\nabla^2 L(\boldsymbol{\theta}_t^{\text{Grad}}) - \nabla^2 L_D(\boldsymbol{\theta}_t^{\text{Grad}})\|_2 \|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 + \eta_t^3 \frac{\rho_H}{6}\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^3 \\
\leq\ & -\eta_t \|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 + \frac{\eta_t^2}{2}\Lambda_G \|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 + \frac{\eta_t^2}{2}\delta_D \|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 + \eta_t^3 \frac{\rho_H}{6}\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^3 \\
\leq\ & -\eta_t \|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 + \frac{\eta_t}{2}\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 + \frac{\eta_t}{6}\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 + \frac{\eta_t}{6}\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 \\
& \text{(From } \eta_t \leq \min(\frac{1}{\Lambda_G}, \frac{1}{\sqrt{\rho_H \|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2}}) \text{ and } \delta_D < \Lambda_G/3) \\
=\ & -\frac{\eta_t}{6}\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2.
\end{aligned}$$

Taking the telescoping sum, and noting that $\boldsymbol{\theta}_0 = \boldsymbol{\theta}_0^{\text{Grad}}$, we have:

$$\begin{aligned}
L(\boldsymbol{\theta}_T^{\text{Grad}}) - L(\boldsymbol{\theta}_0) &\le -\frac{1}{6}\sum_{t=0}^{T-1} \eta_t \|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 \\
&\le -\frac{\zeta_0}{6}\sum_{t=0}^{T-1} \min\Big(\frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2}{\Lambda_G}, \frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^{3/2}}{\sqrt{\rho_H}}\Big) \\
&\quad \text{(From } \eta_t \ge \zeta_0 \min\Big(\frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2}{\Lambda_G}, \frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^{3/2}}{\sqrt{\rho_H}}\Big)\text{)}
\end{aligned}$$

Assume that $\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2 \ge \sqrt{P}\varepsilon$ holds for all $0 \le t < T$. Then, we have

$$\begin{aligned}
L(\boldsymbol{\theta}_T^{\text{Grad}}) - L(\boldsymbol{\theta}_0) &\le -\frac{T\zeta_0}{6}\min\Big(\frac{P\varepsilon^2}{\Lambda_G}, \frac{P^{3/4}\varepsilon^{3/2}}{\sqrt{\rho_H}}\Big) \\
&= -\frac{TP\varepsilon^2\zeta_0}{6\Lambda_G} \quad \text{(From } \varepsilon < \frac{\Lambda_G^2}{\rho_H\sqrt{P}}\text{)}.
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
T &\le \frac{6(L(\boldsymbol{\theta}_0) - L(\boldsymbol{\theta}_T^{\text{Grad}}))}{P\varepsilon^2\zeta_0}\Lambda_G \\
&\le \frac{6(L(\boldsymbol{\theta}_0) - L_*)}{P\varepsilon^2\zeta_0}\Lambda_G.
\end{aligned}$$

This means

$$\mathcal{T}_\varepsilon(\{\boldsymbol{\theta}_t^{\text{Grad}}\}_{t=0}^{\infty}, L, \|\cdot\|_2) \le \frac{6(L(\boldsymbol{\theta}_0) - L_*)}{P\varepsilon^2\zeta_0}\Lambda_G.$$

□

Proof of sign-based sequence. The update rule of the sign-based sequence in deterministic setting is $\boldsymbol{\theta}_{t+1}^{\text{Sign}} = \boldsymbol{\theta}_t^{\text{Sign}} - \eta_t \operatorname{sign}(\nabla L(\boldsymbol{\theta}_t^{\text{Sign}}))$. Thus, we obtain:

$$
\begin{aligned}
& L(\boldsymbol{\theta}_{t+1}^{\text{Sign}}) - L(\boldsymbol{\theta}_t^{\text{Sign}}) \\
\leq\ & \nabla L(\boldsymbol{\theta}_t^{\text{Sign}})^\top (\boldsymbol{\theta}_{t+1}^{\text{Sign}} - \boldsymbol{\theta}_t^{\text{Sign}}) + \frac{1}{2}(\boldsymbol{\theta}_{t+1}^{\text{Sign}} - \boldsymbol{\theta}_t^{\text{Sign}})^\top \nabla^2 L(\boldsymbol{\theta}_t^{\text{Sign}})(\boldsymbol{\theta}_{t+1}^{\text{Sign}} - \boldsymbol{\theta}_t^{\text{Sign}}) + \frac{\rho_H}{6}\|\boldsymbol{\theta}_{t+1}^{\text{Sign}} - \boldsymbol{\theta}_t^{\text{Sign}}\|_2^3 \\
& \qquad \text{(From } Lemma\ C.1\text{)} \\
=\ & -\eta_t \|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 + \frac{\eta_t^2}{2} \operatorname{sign}(\nabla L(\boldsymbol{\theta}_t^{\text{Sign}}))^\top \nabla^2 L(\boldsymbol{\theta}_t^{\text{Sign}}) \operatorname{sign}(\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})) + \eta_t^3 \frac{\rho_H}{6} \|\operatorname{sign}(\nabla L(\boldsymbol{\theta}_t^{\text{Sign}}))\|_2^3 \\
=\ & -\eta_t \|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 + \frac{\eta_t^2}{2} \operatorname{sign}(\nabla L(\boldsymbol{\theta}_t^{\text{Sign}}))^\top \nabla^2 L_D(\boldsymbol{\theta}_t^{\text{Sign}}) \operatorname{sign}(\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})) \\
& + \frac{\eta_t^2}{2} \operatorname{sign}(\nabla L(\boldsymbol{\theta}_t^{\text{Sign}}))^\top (\nabla^2 L(\boldsymbol{\theta}_t^{\text{Sign}}) - \nabla^2 L_D(\boldsymbol{\theta}_t^{\text{Sign}})) \operatorname{sign}(\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})) + \eta_t^3 \frac{\rho_H}{6} P^{3/2} \\
=\ & -\eta_t \|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 + \frac{\eta_t^2}{2} \sum_b [\operatorname{sign}(\nabla L(\boldsymbol{\theta}_t^{\text{Sign}}))]_b^\top [\nabla^2 L(\boldsymbol{\theta}_t^{\text{Sign}})]_b [\operatorname{sign}(\nabla L(\boldsymbol{\theta}_t^{\text{Sign}}))]_b \\
& + \frac{\eta_t^2}{2} \operatorname{sign}(\nabla L(\boldsymbol{\theta}_t^{\text{Sign}}))^\top (\nabla^2 L(\boldsymbol{\theta}_t^{\text{Sign}}) - \nabla^2 L_D(\boldsymbol{\theta}_t^{\text{Sign}})) \operatorname{sign}(\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})) + \eta_t^3 \frac{\rho_H}{6} P^{3/2} \\
\leq\ & -\eta_t \|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 + \frac{\eta_t^2}{2} \sum_b \|[\nabla^2 L(\boldsymbol{\theta}_t^{\text{Sign}})]_b\|_2 P_b + \frac{\eta_t^2}{2} \|\nabla^2 L(\boldsymbol{\theta}_t^{\text{Sign}}) - \nabla^2 L_D(\boldsymbol{\theta}_t^{\text{Sign}})\|_2 P + \eta_t^3 \frac{\rho_H}{6} P^{3/2} \\
\leq\ & -\eta_t \|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 + \frac{\eta_t^2}{2} \Lambda_P P + \frac{\eta_t^2}{2} \delta_D P + \eta_t^3 \frac{\rho_H}{6} P^{3/2} \\
\leq\ & -\eta_t \|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 + \frac{\eta_t}{2} \|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 + \frac{\eta_t}{6} \|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 + \frac{\eta_t}{6} \|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 \\
& \text{(From } \eta_t \leq \min(\frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1}{\Lambda_P P}, \sqrt{\frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1}{\rho_H P^{3/2}}}) \text{ and } \delta_D < \Lambda_P/3\text{)} \\
=\ & -\frac{\eta_t}{6} \|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1.
\end{aligned}
$$

Taking the telescoping sum, and noting that $\boldsymbol{\theta}_0 = \boldsymbol{\theta}_0^{\text{Sign}}$, we have:

$$
\begin{aligned}
L(\boldsymbol{\theta}_T^{\text{Sign}}) - L(\boldsymbol{\theta}_0) &\leq -\frac{1}{6} \sum_{t=0}^{T-1} \eta_t \|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 \\
&\leq -\frac{\zeta_0}{6} \sum_{t=0}^{T-1} \min(\frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1}{P \Lambda_P}, \sqrt{\frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1}{\rho_H P^{3/2}}}) \|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 \\
&\qquad \text{(From } \eta_t \geq \zeta_0 \min(\frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1}{\Lambda_P P}, \sqrt{\frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1}{\rho_H P^{3/2}}})\text{)}
\end{aligned}
$$

Assume that $\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 \geq P\varepsilon$ holds for all $0 \leq t < T$. Then, we have

$$
\begin{aligned}
L(\boldsymbol{\theta}_T^{\text{Sign}}) - L(\boldsymbol{\theta}_0) &\leq -\frac{TP\varepsilon\zeta_0}{6} \min(\frac{\varepsilon}{\Lambda_P}, \sqrt{\frac{\varepsilon}{\rho_H P^{1/2}}}) \\
&= -\frac{TP\varepsilon^2\zeta_0}{6\Lambda_P} \quad \text{(From } \varepsilon < \frac{\Lambda_P^2}{\rho_H \sqrt{P}}\text{)}.
\end{aligned}
$$

Therefore, we have:

$$
\begin{aligned}
T &\leq \frac{6(L(\boldsymbol{\theta}_0) - L(\boldsymbol{\theta}_T^{\text{Sign}}))}{P\varepsilon^2\zeta_0} \Lambda_P \\
&\leq \frac{6(L(\boldsymbol{\theta}_0) - L_*)}{P\varepsilon^2\zeta_0} \Lambda_P.
\end{aligned}
$$

This means:

$$\mathcal{T}_\varepsilon(\{\boldsymbol{\theta}_t^{\text{Sign}}\}_{t=0}^{\infty}, L, \|\cdot\|_1) \leq \frac{6(L(\boldsymbol{\theta}_0) - L_*)}{P\varepsilon^2\zeta_0}\Lambda_P.$$

□

C.3 Proof of Theorem 4.9

Theorem 4.9 is restated. *Assume $\delta_D < \min(\Lambda_G, \Lambda_P)/3$. Then, the iteration complexities in stochastic settings are bounded as follows.*

For the gradient-based sequence, suppose that $\varepsilon < \frac{(1+\sigma_2)^2\Lambda_G^2}{4(1+\sigma_3)\rho_H\sqrt{P}}$ holds and that the learning rate at time t satisfies $\eta_t = \zeta_t \min(\frac{1}{(1+\sigma_2)\Lambda_G}, \frac{1}{2\sqrt{(1+\sigma_3)\rho_H\|\nabla L(\boldsymbol{\theta}_t^{\mathit{Grad}})\|_2}})$, where $\zeta_t \in [\zeta_0, 1]$, we have

$$\mathcal{T}_\varepsilon(\{\boldsymbol{\theta}_t^{\mathit{Grad}}\}_{t=0}^{\infty}, L, \|\cdot\|_2) \leq \frac{12(1+\sigma_2)(L(\boldsymbol{\theta}_0) - L_*)}{P\varepsilon^2\zeta_0}\Lambda_G.$$

For the sign-based sequence, suppose that $\varepsilon < \frac{\Lambda_P^2}{\rho_H\sqrt{P}}$ and $\sigma_2 \leq \frac{1}{24}$ hold and that the learning rate at time t satisfies $\eta_t = \zeta_t \min(\frac{\|\nabla L(\boldsymbol{\theta}_t^{\mathit{Sign}})\|_1}{\Lambda_P P}, \sqrt{\frac{\|\nabla L(\boldsymbol{\theta}_t^{\mathit{Sign}})\|_1}{\rho_H P^{3/2}}})$, where $\zeta_t \in [\zeta_0, 1]$, we have

$$\mathcal{T}_\varepsilon(\{\boldsymbol{\theta}_t^{\mathit{Sign}}\}_{t=0}^{\infty}, L, \|\cdot\|_1) \leq \frac{12(1+24\sigma_2)(L(\boldsymbol{\theta}_0) - L_*)}{P\varepsilon^2\zeta_0}\Lambda_P.$$

Proof of gradient-based sequence. The update rule of the gradient-based sequence in stochastic setting is $\boldsymbol{\theta}_{t+1}^{\text{Grad}} = \boldsymbol{\theta}_t^{\text{Grad}} - \eta_t\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}})$. Thus, we obtain:

$$\begin{aligned}
&\mathbb{E}\left[L(\boldsymbol{\theta}_{t+1}^{\text{Grad}}) - L(\boldsymbol{\theta}_t^{\text{Grad}}) \mid \boldsymbol{\theta}_t^{\text{Grad}}\right] \\
&\leq \mathbb{E}\left[\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})^\top(\boldsymbol{\theta}_{t+1}^{\text{Grad}} - \boldsymbol{\theta}_t^{\text{Grad}}) + \frac{1}{2}(\boldsymbol{\theta}_{t+1}^{\text{Grad}} - \boldsymbol{\theta}_t^{\text{Grad}})^\top\nabla^2 L(\boldsymbol{\theta}_t^{\text{Grad}})(\boldsymbol{\theta}_{t+1}^{\text{Grad}} - \boldsymbol{\theta}_t^{\text{Grad}}) + \frac{\rho_H}{6}\|\boldsymbol{\theta}_{t+1}^{\text{Grad}} - \boldsymbol{\theta}_t^{\text{Grad}}\|_2^3 \mid \boldsymbol{\theta}_t^{\text{Grad}}\right] \\
&\qquad \text{(From } Lemma\ C.1\text{)} \\
&= -\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 + \mathbb{E}\left[\frac{\eta_t^2}{2}\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}})^\top\nabla^2 L(\boldsymbol{\theta}_t^{\text{Grad}})\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}}) + \eta_t^3\frac{\rho_H}{6}\|\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^3 \mid \boldsymbol{\theta}_t^{\text{Grad}}\right] \\
&\qquad \text{(From } \mathbb{E}[\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}})] = \nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\text{)} \\
&= -\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 + \mathbb{E}\left[\frac{\eta_t^2}{2}\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}})^\top\nabla^2 L_D(\boldsymbol{\theta}_t^{\text{Grad}})\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}}) \mid \boldsymbol{\theta}_t^{\text{Grad}}\right] \\
&\quad + \mathbb{E}\left[\frac{\eta_t^2}{2}\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}})^\top(\nabla^2 L(\boldsymbol{\theta}_t^{\text{Grad}}) - \nabla^2 L_D(\boldsymbol{\theta}_t^{\text{Grad}}))\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}}) + \eta_t^3\frac{\rho_H}{6}\|\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^3 \mid \boldsymbol{\theta}_t^{\text{Grad}}\right] \\
&= -\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 + \mathbb{E}\left[\frac{\eta_t^2}{2}\sum_b[\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}})]_b^\top[\nabla^2 L(\boldsymbol{\theta}_t^{\text{Grad}})]_b[\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}})]_b \mid \boldsymbol{\theta}_t^{\text{Grad}}\right] \\
&\quad + \mathbb{E}\left[\frac{\eta_t^2}{2}\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}})^\top(\nabla^2 L(\boldsymbol{\theta}_t^{\text{Grad}}) - \nabla^2 L_D(\boldsymbol{\theta}_t^{\text{Grad}}))\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}}) + \eta_t^3\frac{\rho_H}{6}\|\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^3 \mid \boldsymbol{\theta}_t^{\text{Grad}}\right] \\
&\leq -\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 + \mathbb{E}\left[\frac{\eta_t^2}{2}\sum_b\|[\nabla^2 L(\boldsymbol{\theta}_t^{\text{Grad}})]_b\|_2\|[\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}})]_b\|_2^2 \mid \boldsymbol{\theta}_t^{\text{Grad}}\right] \\
&\quad + \mathbb{E}\left[\frac{\eta_t^2}{2}\|\nabla^2 L(\boldsymbol{\theta}_t^{\text{Grad}}) - \nabla^2 L_D(\boldsymbol{\theta}_t^{\text{Grad}})\|_2\|\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 \mid \boldsymbol{\theta}_t^{\text{Grad}}\right] + \mathbb{E}\left[\eta_t^3\frac{\rho_H}{6}\|\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^3 \mid \boldsymbol{\theta}_t^{\text{Grad}}\right].
\end{aligned} \tag{9}$$

For the second and third term, we can derive an upper bound as follows:

$$
\begin{aligned}
&\mathbb{E}\left[\frac{\eta_t^2}{2}\sum_b \|[\nabla^2 L(\boldsymbol{\theta}_t^{\text{Grad}})]_b\|_2 \|[\nabla \widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}})]_b\|_2^2 \mid \boldsymbol{\theta}_t^{\text{Grad}}\right] + \mathbb{E}\left[\frac{\eta_t^2}{2}\|\nabla^2 L(\boldsymbol{\theta}_t^{\text{Grad}}) - \nabla^2 L_D(\boldsymbol{\theta}_t^{\text{Grad}})\|_2 \|\nabla \widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 \mid \boldsymbol{\theta}_t^{\text{Grad}}\right] \\
&\leq \mathbb{E}\left[\frac{\eta_t^2}{2}\sum_b \|[\nabla^2 L(\boldsymbol{\theta}_t^{\text{Grad}})]_b\|_2 \|[\nabla \widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}})]_b\|_2^2 \mid \boldsymbol{\theta}_t^{\text{Grad}}\right] + \mathbb{E}\left[\frac{\eta_t^2}{2}\delta_D \|\nabla \widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 \mid \boldsymbol{\theta}_t^{\text{Grad}}\right] \\
&= \frac{\eta_t^2}{2}\sum_b \|[\nabla^2 L(\boldsymbol{\theta}_t^{\text{Grad}})]_b\|_2 \sum_i \mathbb{E}\left[(([\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})]_b)_i + ([\nabla \widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}})]_b)_i - ([\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})]_b)_i)^2 \mid \boldsymbol{\theta}_t^{\text{Grad}}\right] \\
&\quad + \frac{\eta_t^2}{2}\delta_D \sum_i \mathbb{E}\left[(\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})_i + \nabla \widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}})_i - \nabla L(\boldsymbol{\theta}_t^{\text{Grad}})_i)^2 \mid \boldsymbol{\theta}_t^{\text{Grad}}\right] \\
&\leq \frac{\eta_t^2}{2}\sum_b \|[\nabla^2 L(\boldsymbol{\theta}_t^{\text{Grad}})]_b\|_2 (1+\sigma_2)([\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})]_b)_i^2 + \frac{\eta_t^2}{2}\delta_D \sum_i (1+\sigma_2)\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})_i^2 \quad (\text{From } Eqs.(2)(4)) \\
&\leq \frac{\eta_t^2}{2}(1+\sigma_2)\Lambda_G \|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 + \frac{\eta_t^2}{2}(1+\sigma_2)\delta_D \|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 \\
&\leq \frac{2\eta_t^2}{3}(1+\sigma_2)\Lambda_G \|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 \quad (\text{From } \delta_D < \Lambda_G/3).
\end{aligned} \tag{10}
$$

For the fourth term, we can derive an upper bound as follows:

$$
\begin{aligned}
&\mathbb{E}\left[\eta_t^3 \frac{\rho_H}{6}\|\nabla \widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^3 \mid \boldsymbol{\theta}_t^{\text{Grad}}\right] \\
&\leq \eta_t^3 \frac{\rho_H}{6}\mathbb{E}\left[(\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2 + \|\nabla \widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}}) - \nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2)^3 \mid \boldsymbol{\theta}_t^{\text{Grad}}\right] \\
&\leq \frac{2\eta_t^3 \rho_H}{3}\mathbb{E}\left[\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^3 + \|\nabla \widehat{L}(\boldsymbol{\theta}_t^{\text{Grad}}) - \nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^3 \mid \boldsymbol{\theta}_t^{\text{Grad}}\right] \quad (\text{From } Lemma\ C.2) \\
&\leq \frac{2\eta_t^3 \rho_H}{3}(1+\sigma_3)\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^3 \quad (\text{From } Eq.(3)).
\end{aligned} \tag{11}
$$

Combining Eqs.(9)(10)(11), we have:

$$
\begin{aligned}
&\mathbb{E}\left[L(\boldsymbol{\theta}_{t+1}^{\text{Grad}}) - L(\boldsymbol{\theta}_t^{\text{Grad}}) \mid \boldsymbol{\theta}_t^{\text{Grad}}\right] \\
&\leq -\eta_t \|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 + \frac{2\eta_t^2}{3}(1+\sigma_2)\Lambda_G \|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 + \frac{2\eta_t^3 \rho_H}{3}(1+\sigma_3)\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^3 \\
&\leq -\frac{\eta_t}{6}\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 \quad (\text{From } \eta_t \leq \min(\frac{1}{(1+\sigma_2)\Lambda_G}, \frac{1}{2\sqrt{(1+\sigma_3)\rho_H \|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2}}))
\end{aligned}
$$

Assume that the probability of the event $\mathcal{E}(T) = \{\forall s \leq T,\ \|\nabla L(\boldsymbol{\theta}_s^{\text{Grad}})\|_2 \geq \sqrt{P}\varepsilon\}$ satisfies $\mathbb{P}(\mathcal{E}(T)) \geq \frac{1}{2}$. By applying the telescoping sum and taking expectations, and noting that $\boldsymbol{\theta}_0 = \boldsymbol{\theta}_0^{\text{Grad}}$, we have:

$$
\begin{aligned}
&\mathbb{E}\left[L(\boldsymbol{\theta}_T^{\text{Grad}})\right] - L(\boldsymbol{\theta}_0) \\
\le& -\frac{1}{6}\sum_{t=0}^{T-1}\mathbb{E}\left[\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2\right] \\
=& -\frac{1}{6}\sum_{t=0}^{T-1}\left(\mathbb{E}\left[\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 \mid \mathcal{E}(T)\right]\mathbb{P}\left(\mathcal{E}(T)\right) + \mathbb{E}\left[\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 \mid \overline{\mathcal{E}(T)}\right]\mathbb{P}\left(\overline{\mathcal{E}(T)}\right)\right) \\
\le& -\frac{1}{6}\sum_{t=0}^{T-1}\mathbb{E}\left[\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 \mid \mathcal{E}(T)\right]\mathbb{P}\left(\mathcal{E}(T)\right) \\
\le& -\frac{1}{12}\sum_{t=0}^{T-1}\mathbb{E}\left[\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2 \mid \mathcal{E}(T)\right] \\
\le& -\frac{\zeta_0}{12}\sum_{t=0}^{T-1}\mathbb{E}\left[\min\left(\frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^2}{(1+\sigma_2)\Lambda_G}, \frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2^{3/2}}{2\sqrt{(1+\sigma_3)\rho_H}}\right) \mid \mathcal{E}(T)\right] \\
&\quad \text{(From } \eta_t \ge \zeta_0\min\left(\frac{1}{(1+\sigma_2)\Lambda_G}, \frac{1}{2\sqrt{(1+\sigma_3)\rho_H\|\nabla L(\boldsymbol{\theta}_t^{\text{Grad}})\|_2}}\right)) \\
\le& -\frac{T\zeta_0}{12}\min\left(\frac{P\varepsilon^2}{(1+\sigma_2)\Lambda_G}, \frac{P^{3/4}\varepsilon^{3/2}}{2\sqrt{(1+\sigma_3)\rho_H}}\right) \\
=& -\frac{TP\varepsilon^2\zeta_0}{12(1+\sigma_2)\Lambda_G} \quad \text{(From } \varepsilon < \frac{(1+\sigma_2)^2\Lambda_G^2}{4(1+\sigma_3)\rho_H\sqrt{P}}\text{)}.
\end{aligned}
$$

Therefore, we have

$$
\begin{aligned}
T &\le \frac{12(1+\sigma_2)(L(\boldsymbol{\theta}_0) - \mathbb{E}\left[L(\boldsymbol{\theta}_T^{\text{Grad}})\right])}{P\varepsilon^2\zeta_0}\Lambda_G \\
&\le \frac{12(1+\sigma_2)(L(\boldsymbol{\theta}_0) - L_*)}{P\varepsilon^2\zeta_0}\Lambda_G.
\end{aligned}
$$

This means that when we take $T > \frac{12(1+\sigma_2)(L(\boldsymbol{\theta}_0)-L_*)}{P\varepsilon^2\zeta_0}\Lambda_G$, we have $\mathbb{P}\left(\mathcal{E}(T)\right) < \frac{1}{2}$. Therefore, we have

$$
\mathcal{T}_\varepsilon(\{\boldsymbol{\theta}_t^{\text{Grad}}\}_{t=0}^{\infty}, L, \|\cdot\|_2) \le \frac{12(1+\sigma_2)(L(\boldsymbol{\theta}_0) - L_*)}{P\varepsilon^2\zeta_0}\Lambda_G.
$$

□

Proof of sign-based sequence. The update rule of the sign-based sequence in stochastic setting is $\boldsymbol{\theta}_{t+1}^{\text{Sign}} = \boldsymbol{\theta}_t^{\text{Sign}} - \eta_t \operatorname{sign}(\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}}))$. Thus, we obtain:

$$
\begin{aligned}
&\mathbb{E}\left[L(\boldsymbol{\theta}_{t+1}^{\text{Sign}}) - L(\boldsymbol{\theta}_t^{\text{Sign}}) \mid \boldsymbol{\theta}_t^{\text{Sign}}\right] \\
\le& \mathbb{E}\left[\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})^\top(\boldsymbol{\theta}_{t+1}^{\text{Sign}} - \boldsymbol{\theta}_t^{\text{Sign}}) + \frac{1}{2}(\boldsymbol{\theta}_{t+1}^{\text{Sign}} - \boldsymbol{\theta}_t^{\text{Sign}})^\top\nabla^2 L(\boldsymbol{\theta}_t^{\text{Sign}})(\boldsymbol{\theta}_{t+1}^{\text{Sign}} - \boldsymbol{\theta}_t^{\text{Sign}}) + \frac{\rho_H}{6}\|\boldsymbol{\theta}_{t+1}^{\text{Sign}} - \boldsymbol{\theta}_t^{\text{Sign}}\|_2^3 \mid \boldsymbol{\theta}_t^{\text{Sign}}\right] \\
&\quad \text{(From } Lemma\ C.1\text{)} \\
=& -\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 + \mathbb{E}\left[\frac{\eta_t^2}{2}\operatorname{sign}(\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}}))^\top\nabla^2 L(\boldsymbol{\theta}_t^{\text{Sign}})\operatorname{sign}(\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}})) + \eta_t^3\frac{\rho_H}{6}\|\operatorname{sign}(\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}}))\|_2^3 \mid \boldsymbol{\theta}_t^{\text{Sign}}\right] \\
&+ \mathbb{E}\left[-\eta_t\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})^\top(\operatorname{sign}(\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}})) - \operatorname{sign}(\nabla L(\boldsymbol{\theta}_t^{\text{Sign}}))) \mid \boldsymbol{\theta}_t^{\text{Sign}}\right]. \qquad (12)
\end{aligned}
$$

For the second term, we can derive an upper bound in the same way as in the deterministic case:

$$
\begin{aligned}
&\mathbb{E}\left[\frac{\eta_t^2}{2}\operatorname{sign}(\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}}))^\top\nabla^2 L(\boldsymbol{\theta}_t^{\text{Sign}})\operatorname{sign}(\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}})) + \eta_t^3\frac{\rho_H}{6}\|\operatorname{sign}(\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}}))\|_2^3 \mid \boldsymbol{\theta}_t^{\text{Sign}}\right] \\
&= \mathbb{E}\left[\frac{\eta_t^2}{2}\operatorname{sign}(\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}}))^\top\nabla^2 L_D(\boldsymbol{\theta}_t^{\text{Sign}})\operatorname{sign}(\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}})) \mid \boldsymbol{\theta}_t^{\text{Sign}}\right] \\
&\quad + \mathbb{E}\left[\frac{\eta_t^2}{2}\operatorname{sign}(\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}}))^\top(\nabla^2 L(\boldsymbol{\theta}_t^{\text{Sign}}) - \nabla^2 L_D(\boldsymbol{\theta}_t^{\text{Sign}}))\operatorname{sign}(\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}})) \mid \boldsymbol{\theta}_t^{\text{Sign}}\right] + \eta_t^3\frac{\rho_H}{6}P^{3/2} \\
&= \mathbb{E}\left[\frac{\eta_t^2}{2}\sum_b [\operatorname{sign}(\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}}))]_b^\top[\nabla^2 L(\boldsymbol{\theta}_t^{\text{Sign}})]_b[\operatorname{sign}(\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}}))]_b \mid \boldsymbol{\theta}_t^{\text{Sign}}\right] \\
&\quad + \mathbb{E}\left[\frac{\eta_t^2}{2}\operatorname{sign}(\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}}))^\top(\nabla^2 L(\boldsymbol{\theta}_t^{\text{Sign}}) - \nabla^2 L_D(\boldsymbol{\theta}_t^{\text{Sign}}))\operatorname{sign}(\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}})) \mid \boldsymbol{\theta}_t^{\text{Sign}}\right] + \eta_t^3\frac{\rho_H}{6}P^{3/2} \\
&\le \frac{\eta_t^2}{2}\sum_b \|[\nabla^2 L(\boldsymbol{\theta}_t^{\text{Sign}})]_b\|_2 P_b + \frac{\eta_t^2}{2}\|\nabla^2 L(\boldsymbol{\theta}_t^{\text{Sign}}) - \nabla^2 L_D(\boldsymbol{\theta}_t^{\text{Sign}})\|_2 P + \eta_t^3\frac{\rho_H}{6}P^{3/2} \\
&\le \frac{\eta_t^2}{2}\Lambda_P P + \frac{\eta_t^2}{2}\delta_D P + \eta_t^3\frac{\rho_H}{6}P^{3/2}. \qquad (13)
\end{aligned}
$$

For the third term, we can derive an upper bound as follows:

$$
\begin{aligned}
&\mathbb{E}\left[-\eta_t\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})^\top(\operatorname{sign}(\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}})) - \operatorname{sign}(\nabla L(\boldsymbol{\theta}_t^{\text{Sign}}))) \mid \boldsymbol{\theta}_t^{\text{Sign}}\right] \\
&= \eta_t\sum_{i=1}^P \nabla L(\boldsymbol{\theta}_t^{\text{Sign}})_i\mathbb{E}\left[\operatorname{sign}(\nabla L(\boldsymbol{\theta}_t^{\text{Sign}}))_i - \operatorname{sign}(\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}}))_i \mid \boldsymbol{\theta}_t^{\text{Sign}}\right] \\
&= \eta_t\sum_{i=1}^P |\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})_i| 2\mathbb{E}\left[\mathbb{1}[\operatorname{sign}(\nabla L(\boldsymbol{\theta}_t^{\text{Sign}}))_i \ne \operatorname{sign}(\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}}))_i] \mid \boldsymbol{\theta}_t^{\text{Sign}}\right] \\
&= \eta_t\sum_{i=1}^P |\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})_i| 2\mathbb{P}\left(\operatorname{sign}(\nabla L(\boldsymbol{\theta}_t^{\text{Sign}}))_i \ne \operatorname{sign}(\nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}}))_i \mid \boldsymbol{\theta}_t^{\text{Sign}}\right) \\
&\le \eta_t\sum_{i=1}^P |\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})_i| 2\mathbb{P}\left(|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})_i - \nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}})_i| \ge |\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})_i| \mid \boldsymbol{\theta}_t^{\text{Sign}}\right) \\
&\le \eta_t\sum_{i=1}^P |\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})_i| 2\frac{\mathbb{E}\left[|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})_i - \nabla\widehat{L}(\boldsymbol{\theta}_t^{\text{Sign}})_i|^2 \mid \boldsymbol{\theta}_t^{\text{Sign}}\right]}{|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})_i|^2} \quad \text{(From Chebyshev's inequality)} \\
&\le \eta_t\sum_{i=1}^P |\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})_i| 2\sigma_2 \quad \text{(From } Eq.(4)\text{)} \\
&= 2\sigma_2\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1. \qquad (14)
\end{aligned}
$$

Combining Eqs.(12)(13)(14), we have:

$$
\begin{aligned}
&\mathbb{E}\left[L(\boldsymbol{\theta}_{t+1}^{\text{Sign}}) - L(\boldsymbol{\theta}_t^{\text{Sign}}) \mid \boldsymbol{\theta}_t^{\text{Sign}}\right] \\
&\le -\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 + \frac{\eta_t^2}{2}\Lambda_P P + \frac{\eta_t^2}{2}\delta_D P + \eta_t^3\frac{\rho_H}{6}P^{3/2} + 2\sigma_2\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 \qquad (15) \\
&\le -\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 + \frac{\eta_t}{2}\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 + \frac{\eta_t}{6}\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 + \frac{\eta_t}{6}\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 + 2\sigma_2\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 \\
&\qquad \text{(From } \eta_t \le \min(\frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1}{\Lambda_P P}, \sqrt{\frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1}{\rho_H P^{3/2}}}) \text{ and } \delta_D < \Lambda_P/3\text{)} \\
&= -\frac{(1-12\sigma_2)\eta_t}{6}\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 \\
&\le -\frac{\eta_t}{6(1+24\sigma_2)}\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 \quad \text{(From } \sigma_2 \le \frac{1}{24}\text{)}
\end{aligned}
$$

Assume that the probability of the event $\mathcal{E}(T) = \{\forall s \leq T,\ \|\nabla L(\boldsymbol{\theta}_s^{\text{Sign}})\|_1 \geq P\varepsilon\}$ satisfies $\mathbb{P}\left(\mathcal{E}(T)\right) \geq \frac{1}{2}$. By applying the telescoping sum and taking expectations, and noting that $\boldsymbol{\theta}_0 = \boldsymbol{\theta}_0^{\text{Sign}}$, we have:

$$
\begin{aligned}
&\mathbb{E}\left[L(\boldsymbol{\theta}_T^{\text{Sign}})\right] - L(\boldsymbol{\theta}_0) \\
&\leq -\frac{1}{6(1+24\sigma_2)} \sum_{t=0}^{T-1} \mathbb{E}\left[\eta_t \|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1\right] \\
&= -\frac{1}{6(1+24\sigma_2)} \sum_{t=0}^{T-1} \left(\mathbb{E}\left[\eta_t \|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 \mid \mathcal{E}(T)\right] \mathbb{P}\left(\mathcal{E}(T)\right) + \mathbb{E}\left[\overline{\eta}_t \|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 \mid \overline{\mathcal{E}(T)}\right] \mathbb{P}\left(\overline{\mathcal{E}(T)}\right)\right) \\
&\leq -\frac{1}{6(1+24\sigma_2)} \sum_{t=0}^{T-1} \mathbb{E}\left[\eta_t \|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 \mid \mathcal{E}(T)\right] \mathbb{P}\left(\mathcal{E}(T)\right) \\
&\leq -\frac{1}{12(1+24\sigma_2)} \sum_{t=0}^{T-1} \mathbb{E}\left[\eta_t \|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 \mid \mathcal{E}(T)\right] \\
&\leq -\frac{\zeta_0}{12(1+24\sigma_2)} \sum_{t=0}^{T-1} \mathbb{E}\left[\min\left(\frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1^2}{\Lambda_P P}, \frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1^{3/2}}{\sqrt{\rho_H P^{3/2}}}\right) \mid \mathcal{E}(T)\right] \\
&\qquad \left(\text{From } \eta_t \geq \zeta_0 \min\left(\frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1}{\Lambda_P P}, \sqrt{\frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1}{\rho_H P^{3/2}}}\right)\right) \\
&\leq -\frac{\zeta_0}{12(1+24\sigma_2)} \sum_{t=0}^{T-1} \min\left(\frac{P\varepsilon^2}{\Lambda_P}, P\varepsilon\sqrt{\frac{\varepsilon}{\rho_H P^{1/2}}}\right) \\
&= -\frac{TP\varepsilon^2\zeta_0}{12(1+24\sigma_2)\Lambda_P} \quad \left(\text{From } \varepsilon < \frac{\Lambda_P^2}{\rho_H\sqrt{P}}\right).
\end{aligned}
$$

Therefore, we have:

$$
\begin{aligned}
T &\leq \frac{12(1+24\sigma_2)(L(\boldsymbol{\theta}_0) - \mathbb{E}\left[L(\boldsymbol{\theta}_T^{\text{Sign}})\right])}{P\varepsilon^2\zeta_0}\Lambda_P \\
&\leq \frac{12(1+24\sigma_2)(L(\boldsymbol{\theta}_0) - L_*)}{P\varepsilon^2\zeta_0}\Lambda_P.
\end{aligned}
$$

This means that when we take $T > \frac{12(1+24\sigma_2)(L(\boldsymbol{\theta}_0)-L_*)}{P\varepsilon^2\zeta_0}\Lambda_P$, we have $\mathbb{P}\left(\mathcal{E}(T)\right) < \frac{1}{2}$. Therefore, we have

$$
\mathcal{T}_\varepsilon(\{\boldsymbol{\theta}_t^{\text{Sign}}\}_{t=0}^{\infty}, L, \|\cdot\|_1) \leq \frac{12(1+24\sigma_2)(L(\boldsymbol{\theta}_0) - L_*)}{P\varepsilon^2\zeta_0}\Lambda_P.
$$

□

D DERIVATION OF JACOBIAN MATRIX IN SECTION 4.6

D.1 JACOBIAN OF TRANSFORMER LAYER

The output of a transformer layer for an input $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ is given by $\mathcal{M}(\mathcal{A}(\boldsymbol{X}))$, where $\mathcal{A}(\cdot)$is the attention layer and $\mathcal{M}(\cdot)$ is the feed-forward layer. In the foll owing, we denote the Jacobian of the self-attention module, the feed-forward module, and the layer normalization as $\boldsymbol{J}_{\text{ATT}}$, $\boldsymbol{J}_{\text{FFN}}$, and $\boldsymbol{J}_{\text{LN}}$, respectively.

In Pre-LN. The self-attention and feed-forward layers in the Pre-LN architecture are given by

$$
\begin{aligned}
\mathcal{A}(\boldsymbol{X}) &= \text{ATT}(\text{LN}(\boldsymbol{X})) + \boldsymbol{X}, \\
\mathcal{M}(\boldsymbol{Y}) &= \text{FFN}(\text{LN}(\boldsymbol{Y})) + \boldsymbol{Y}.
\end{aligned}
$$

The Jacobian of these modules are as follows:

$$\begin{aligned}\frac{\partial \mathcal{A}(\boldsymbol{X})}{\partial \boldsymbol{X}} &= \frac{\partial\, \mathrm{ATT}(\boldsymbol{Z})}{\boldsymbol{Z}}\bigg|_{\boldsymbol{Z}=\mathrm{LN}(\boldsymbol{X})} \frac{\partial\, \mathrm{LN}(\boldsymbol{X})}{\partial \boldsymbol{X}} + \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{X}} \\ &= \boldsymbol{J}_{\mathrm{ATT}}(\mathrm{LN}(\boldsymbol{X}))\boldsymbol{J}_{\mathrm{LN}}(\boldsymbol{X}) + \boldsymbol{I}_{nd}, \\ \frac{\partial \mathcal{M}(\boldsymbol{Y})}{\partial \boldsymbol{Y}} &= \frac{\partial\, \mathrm{FFN}(\boldsymbol{Y})}{\boldsymbol{Y}}\bigg|_{\boldsymbol{Y}=\mathrm{LN}(\boldsymbol{Y})} \frac{\partial\, \mathrm{LN}(\boldsymbol{Y})}{\partial \boldsymbol{Y}} + \frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{Y}} \\ &= \boldsymbol{J}_{\mathrm{FFN}}(\mathrm{LN}(\boldsymbol{Y}))\boldsymbol{J}_{\mathrm{LN}}(\boldsymbol{Y}) + \boldsymbol{I}_{nd}.\end{aligned}$$

Therefore, the Jacobian of the Pre-LN layer is given by

$$\begin{aligned}\boldsymbol{J}_{\text{Pre-LN}}(\boldsymbol{X}) &= \frac{\partial \mathcal{M}(\boldsymbol{Y})}{\partial \boldsymbol{Y}}\bigg|_{\boldsymbol{Y}=\mathcal{A}(\boldsymbol{X})} \frac{\partial \mathcal{A}(\boldsymbol{X})}{\partial \boldsymbol{X}} \\ &= \left(\boldsymbol{J}_{\mathrm{FFN}}(\mathrm{LN}(\mathcal{A}(\boldsymbol{X})))\boldsymbol{J}_{\mathrm{LN}}(\mathcal{A}(\boldsymbol{X})) + \boldsymbol{I}_{nd}\right)\left(\boldsymbol{J}_{\mathrm{ATT}}(\mathrm{LN}(\boldsymbol{X}))\boldsymbol{J}_{\mathrm{LN}}(\boldsymbol{X}) + \boldsymbol{I}_{nd}\right)\end{aligned}$$

and with omitting the evaluation point, we can write the Jacobian as

$$\boldsymbol{J}_{\text{Pre-LN}} = \left(\boldsymbol{J}_{\mathrm{FFN}}\boldsymbol{J}_{\mathrm{LN}} + \boldsymbol{I}_{nd}\right)\left(\boldsymbol{J}_{\mathrm{ATT}}\boldsymbol{J}_{\mathrm{LN}} + \boldsymbol{I}_{nd}\right).$$

In Post-LN. The self-attention and feed-forward layers in the Post-LN layer are given by

$$\begin{aligned}\mathcal{A}(\boldsymbol{X}) &= \mathrm{LN}(\mathrm{ATT}(\boldsymbol{X}) + \boldsymbol{X}), \\ \mathcal{M}(\boldsymbol{Y}) &= \mathrm{LN}(\mathrm{FFN}(\boldsymbol{Y}) + \boldsymbol{Y}).\end{aligned}$$

The Jacobian of these modules are as follows:

$$\begin{aligned}\frac{\partial \mathcal{A}(\boldsymbol{X})}{\partial \boldsymbol{X}} &= \frac{\partial\, \mathrm{LN}(\boldsymbol{Z})}{\boldsymbol{Z}}\bigg|_{\boldsymbol{Z}=\mathrm{ATT}(\boldsymbol{X})+\boldsymbol{X}} \left(\frac{\partial\, \mathrm{ATT}(\boldsymbol{X})}{\partial \boldsymbol{X}} + \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{X}}\right) \\ &= \boldsymbol{J}_{\mathrm{LN}}(\mathrm{ATT}(\boldsymbol{X}) + \boldsymbol{X})\left(\boldsymbol{J}_{\mathrm{ATT}}(\boldsymbol{X}) + \boldsymbol{I}_{nd}\right), \\ \frac{\partial \mathcal{M}(\boldsymbol{Y})}{\partial \boldsymbol{Y}} &= \frac{\partial\, \mathrm{LN}(\boldsymbol{Z})}{\boldsymbol{Z}}\bigg|_{\boldsymbol{Z}=\mathrm{FFN}(\boldsymbol{Y})+\boldsymbol{Y}} \left(\frac{\partial\, \mathrm{FFN}(\boldsymbol{Y})}{\partial \boldsymbol{Y}} + \frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{Y}}\right) \\ &= \boldsymbol{J}_{\mathrm{LN}}(\mathrm{FFN}(\boldsymbol{Y}) + \boldsymbol{Y})\left(\boldsymbol{J}_{\mathrm{FFN}}(\boldsymbol{Y}) + \boldsymbol{I}_{nd}\right).\end{aligned}$$

Therefore, the Jacobian of the Post-LN layer is given by

$$\begin{aligned}\boldsymbol{J}_{\text{Post-LN}}(\boldsymbol{X}) &= \frac{\partial \mathcal{M}(\boldsymbol{Y})}{\partial \boldsymbol{Y}}\bigg|_{\boldsymbol{Y}=\mathcal{A}(\boldsymbol{X})} \frac{\partial \mathcal{A}(\boldsymbol{X})}{\partial \boldsymbol{X}} \\ &= \boldsymbol{J}_{\mathrm{LN}}(\mathrm{FFN}(\mathcal{A}(\boldsymbol{X})) + \mathcal{A}(\boldsymbol{X}))\left(\boldsymbol{J}_{\mathrm{FFN}}(\mathcal{A}(\boldsymbol{X})) + \boldsymbol{I}_{nd}\right)\boldsymbol{J}_{\mathrm{LN}}(\mathrm{ATT}(\boldsymbol{X}) + \boldsymbol{X})\left(\boldsymbol{J}_{\mathrm{ATT}}(\boldsymbol{X}) + \boldsymbol{I}_{nd}\right)\end{aligned}$$

and with omitting the evaluation point, we can write the Jacobian as

$$\boldsymbol{J}_{\text{Post-LN}} = \boldsymbol{J}_{\mathrm{LN}}\left(\boldsymbol{J}_{\mathrm{FFN}} + \boldsymbol{I}_{nd}\right)\boldsymbol{J}_{\mathrm{LN}}\left(\boldsymbol{J}_{\mathrm{ATT}} + \boldsymbol{I}_{nd}\right).$$

D.2 JACOBIAN OF LAYER NORMALIZATION

Since the layer normalization is a raw-wise operation, the Jacobian of the layer normalization for the input matrix $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ is given by

$$\boldsymbol{J}_{\mathrm{LN}}(X) = \mathrm{blockdiag}(\{\frac{\partial\, \mathrm{LN}(\boldsymbol{X})_{i,:}}{\partial \boldsymbol{X}_{i,:}}\}_{i=1}^{n}).$$

where $\frac{\partial\, \mathrm{LN}(\boldsymbol{X})_{i,:}}{\partial \boldsymbol{X}_{i,:}}$ is the Jacobian of the layer normalization for the i-th row of the input matrix $\boldsymbol{X}$. The layer normalization for the i-th row of the input matrix $\boldsymbol{X}$ is given by

$$\mathrm{LN}(\boldsymbol{X})_{i,:} = \frac{\sqrt{d}\widetilde{\boldsymbol{X}_{i,:}}}{\|\widetilde{\boldsymbol{X}_{i,:}}\|},$$

where $\widetilde{\boldsymbol{X}_{i,:}} \coloneqq \boldsymbol{X}_{i,:}(\boldsymbol{I}_d - \frac{1}{d}\mathbf{1}\mathbf{1}^\top)$. Therefore, the i-th block of the Jacobian of the layer normali zation is given by

$$\begin{aligned}
\frac{\partial \operatorname{LN}(\boldsymbol{X})_{i,:}}{\partial \boldsymbol{X}_{i,:}} &= \frac{\partial \operatorname{LN}(\boldsymbol{X})_{i,:}}{\partial \widetilde{\boldsymbol{X}_{i,:}}} \frac{\partial \widetilde{\boldsymbol{X}_{i,:}}}{\partial \boldsymbol{X}_{i,:}} \\
&= \sqrt{d}\left(\frac{1}{\|\widetilde{\boldsymbol{X}_{i,:}}\|}\boldsymbol{I}_d - \widetilde{\boldsymbol{X}_{i,:}}\frac{\widetilde{\boldsymbol{X}_{i,:}}^\top}{\|\widetilde{\boldsymbol{X}_{i,:}}\|^3}\right)(\boldsymbol{I}_d - \frac{1}{d}\mathbf{1}\mathbf{1}^\top) \\
&= \frac{\sqrt{d}}{\|\widetilde{\boldsymbol{X}_{i,:}}\|_2}\left(\boldsymbol{I}_d - \frac{\widetilde{\boldsymbol{X}_{i,:}}\widetilde{\boldsymbol{X}_{i,:}}^\top}{\|\widetilde{\boldsymbol{X}_{i,:}}\|_2^2}\right)\left(\boldsymbol{I}_d - \frac{\mathbf{1}\mathbf{1}^\top}{d}\right).
\end{aligned}$$

Therefore, we can write the Jacobian of the layer normalization as

$$\boldsymbol{J}_{\mathrm{LN}}(\boldsymbol{X}) = \operatorname{blockdiag}(\{\boldsymbol{L}_i(\boldsymbol{X})\}_{i=1}^n),$$

where

$$\boldsymbol{L}_i(\boldsymbol{X}) = \frac{\sqrt{d}}{\|\widetilde{\boldsymbol{X}_{i,:}}\|_2}\left(\boldsymbol{I}_d - \frac{\widetilde{\boldsymbol{X}_{i,:}}\widetilde{\boldsymbol{X}_{i,:}}^\top}{\|\widetilde{\boldsymbol{X}_{i,:}}\|_2^2}\right)\left(\boldsymbol{I}_d - \frac{\mathbf{1}\mathbf{1}^\top}{d}\right).$$

E EXPERIMENTAL DETAILS

E.1 IMPLEMENTATION AND TRAINING DETAILS

Our implementation, based on PyTorch (Paszke et al., 2019), uses the HuggingFace Transformers library (Wolf et al., 2020) for NLP tasks and primarily follows Tomihari and Sato (2024). All experiments were conducted on a single NVIDIA A100 GPU. The reported results are averages over one tuning seed and five training seeds. We used the cross-entropy loss, defined as $\ell(\boldsymbol{f}(\boldsymbol{x}), y) := -\log(\mathrm{softmax}(\boldsymbol{f}(\boldsymbol{x}))_y)$, where the function $\mathrm{softmax} : \mathbb{R}^C \to \mathbb{R}^C$ represents the softmax operation.

Following the methodology of Kunstner et al. (2023), we optimized the learning rate via grid search based on the training loss, while keeping other hyperparameters, such as batch size and the number of epochs, fixed. Momentum was set to 0.9 as the default configuration for both SGD and SignSGD, and gradient clipping with a threshold of 1.0 was applied. For NLP tasks, we used linear learning rate scheduling, whereas for vision tasks, a warmup schedule was applied.

Other hyperparameters followed the default values provided by PyTorch, including Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-8$) and RMSProp ($\alpha = 0.99$, $\epsilon = 1e-8$). For NLP tasks, the original training set was split into a 9:1 training-to-validation ratio, with the original validation set used as the test set, following Chen et al. (2022); Tomihari and Sato (2024).

We provide dataset statistics and hyperparameter configurations in Table 5 and Tables 6–8, respectively.

E.2 DETAILS OF EACH EXPERIMENT AND FIGURE

Correlation between Hessian and gradient. In Figure 1, we show the correlation between the Hessian and the gradient. The maximum eigenvalue of the Hessian was computed using power iteration, as described in Park and Kim (2022), with the PyHessian implementation (Yao et al., 2020). To estimate the maximum eigenvectors of the block-diagonal elements of the Hessian, we calculated the product of the Hessian and a random vector for each parameter. The batch size used for these computations was the same as the training batch size. The maximum eigenvalue and the gradient were computed for each batch across all training data.

Correlation between full-batch gradient and gradient error. In Figure 2, we show the correlation between the full-batch gradient and the gradient error in a coordinate-wise manner. We randomly sampled $1,000$ coordinates from the parameters and computed the squared norm of the full-batch gradient and the gradient error for each coordinate. The gradient error is defined as the difference between the full-batch gradient and the gradient computed with a mini-batch. The batch size was the same as the training batch size. The gradient error was computed for each batch across all training data.

Gradient heterogeneity. In Figure 3, we show the ratio of the gradient norm for each parameter relative to the sum of the gradient norms. Specifically, we plot:

$$\frac{G_\theta/\sqrt{P_\theta}}{\sum_{\theta'} G_{\theta'}/\sqrt{P_{\theta'}}},$$

for each parameter $\boldsymbol{\theta}$, where G_θ is the full-batch gradient norm of parameter $\boldsymbol{\theta}$, and P_θ is its dimension. To compare gradient norms across different parameters, we normalize each gradient norm by the square root of its parameter dimension. Bias parameters are omitted in these plots.

Effect of layer normalization. In Tables 2 and 12, all models share the same RoBERTa backbone and differ only in the placement of the normalization layer. To minimize the effect of initialization, we trained scratch-initialized models for 1000 iterations. Note that pre-trained weights are available only for the Post-LN variant.

Training Curve. In Figure 4, we show training runs with the median final loss value among the five training seeds. The shaded area represents the interquartile range across the five seeds. This approach is used to reduce the influence of outliers on the reported results.

Table 5: Dataset statistics, including the number of classes and counts of training (Train), validation (Val), and test samples for each dataset.

Domain	Dataset	Classes	Train	Val	Test
NLP	CB (De Marneffe et al., 2019)	3	225	25	57
	RTE (Wang et al., 2018)	2	2,241	249	277
	BoolQ (Clark et al., 2019)	2	8,484	943	3,270
	WiC (Pilehvar and Camacho-Collados, 2019)	2	5,400	600	638
	CoLA (Warstadt et al., 2019)	2	7,695	855	1,040
	SST-2 (Socher et al., 2013)	2	60,614	6,735	872
	MRPC (Dolan and Brockett, 2005)	2	3,301	367	408
Vision	Flowers102 (Nilsback and Zisserman, 2008)	102	1,632	408	6,149
	Aircraft (Maji et al., 2013)	100	5,334	1,333	3,333

Table 6: Hyperparameter configurations for RoBERTa-Base. The settings include batch size (bs), learning rate (lr), and the number of epochs (epochs). "w/o M" denotes optimizers without momentum and "Const", "Cos", and "Lin-W" denote constant, cosine, and linear with warm-up learning rate schedules, respectively.

Optimizer	Param	CB	RTE	BoolQ	WiC	CoLA	SST-2	MRPC
Common	bs	8	8	32	32	32	32	32
	epochs	20	20	20	20	20	10	20
Adam	lr	$1e-4$	$1e-5$	$1e-5$	$1e-5$	$1e-5$	$1e-5$	$1e-5$
SGD		$1e-2$	$1e-3$	$1e-2$	$1e-3$	$1e-3$	$1e-2$	$1e-2$
SGD (w/o M)		$1e-1$	$1e-2$	$1e-1$	$1e-2$	$1e-2$	$1e-1$	$1e-1$
SignSGD		$1e-5$	$1e-6$	$1e-5$	$1e-5$	$1e-5$	$1e-5$	$1e-5$
SignSGD (w/o M)		$1e-4$	$1e-5$	$1e-5$	$1e-5$	$1e-4$	$1e-5$	$1e-5$
RMSProp		$1e-5$	$1e-5$	$1e-5$	$1e-5$	$1e-5$	$1e-5$	$1e-5$
SGD (Const)		$1e-2$	$1e-3$	-	-	-	-	-
SGD (Cos)		$1e-2$	$1e-3$	-	-	-	-	-
SGD (Lin-W)		$1e-2$	$1e-3$	-	-	-	-	-
SignSGD (Const)		$1e-6$	$1e-6$	-	-	-	-	-
SignSGD (Cos)		$1e-5$	$1e-5$	-	-	-	-	-
SignSGD (Lin-W)		$1e-5$	$1e-5$	-	-	-	-	-

Table 7: Hyperparameter configurations for ResNet18. The settings include batch size (bs), learning rate (lr), and the number of epochs (epochs). "w/o M" denotes optimizers without momentum.

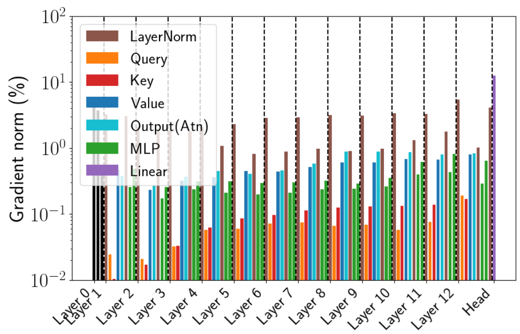
Optimizer	Param	Flowers102	Aircraft
Common	bs	32	32
	epochs	50	100
Adam	lr	$1e-4$	$1e-4$
SGD		$1e-2$	$1e-2$
SGD (w/o M)		$1e-1$	$1e-1$
SignSGD		$1e-5$	$1e-5$
SignSGD (w/o M)		$1e-4$	$1e-4$
RMSProp		$1e-4$	$1e-4$

Table 8: Hyperparameter configurations for ViT-Base. The settings include batch size (bs), learning rate (lr), and the number of epochs (epochs).“w/o M” denotes optimizers without momentum.

Optimizer	Param	Flowers102	Aircraft
Common	bs	32	32
	epochs	50	100
Adam	lr	$1e-5$	$1e-5$
SGD		$1e-2$	$1e-2$
SGD (w/o M)		$1e-1$	$5e-1$
SignSGD		$1e-5$	$1e-5$
SignSGD (w/o M)		$1e-4$	$1e-5$
RMSProp		$1e-5$	$1e-5$

F Additional experimental results

F.1 Gradient heterogeneity of ViT

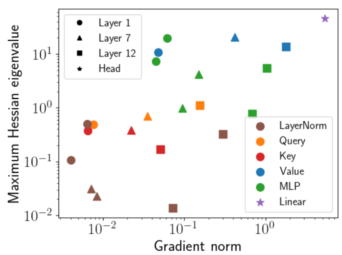


(a) ViT on Flowers102

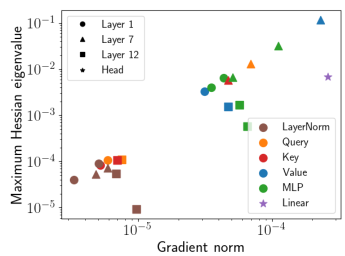
Figure 5: Gradient norms for each parameter of pre-trained models.

F.2 CORRELATION BETWEEN HESSIAN AND GRADIENT

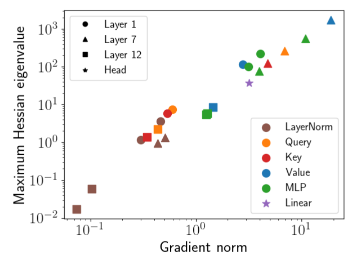
We show the correlation between the Hessian and the gradient in Figure 6. The Hessian and gradient are computed using the pre-trained models or the trained models corresponding to the median final loss value among the five training seeds shown in Figures 4 and 10 and Appendix F.7.



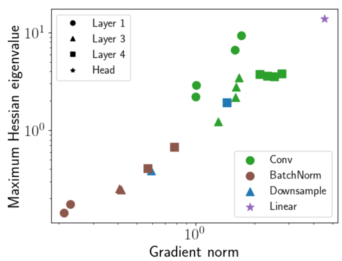
(a) Pre-trained RoBERTa on CB



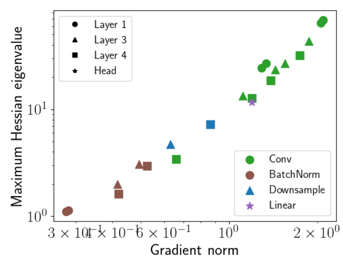
(b) RoBERTa fine-tuned with Adam on RTE



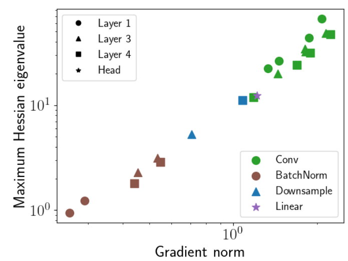
(c) RoBERTa fine-tuned with SGD on RTE



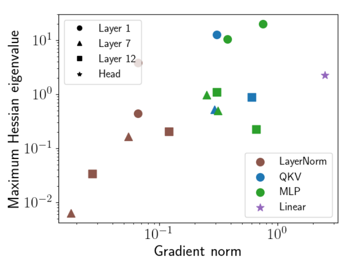
(d) Pre-trained ResNet18 on Flowers102



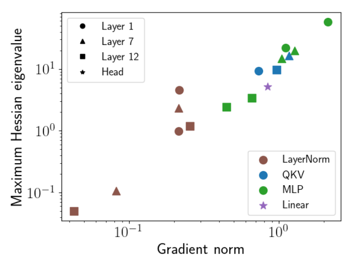
(e) ResNet18 fine-tuned with Adam on Flowers102



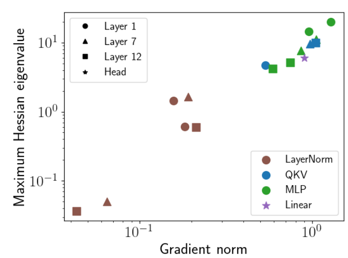
(f) ResNet18 fine-tuned with SGD on Flowers102



(g) Pre-trained ViT on Aircraft



(h) ViT fine-tuned with Adam on Aircraft

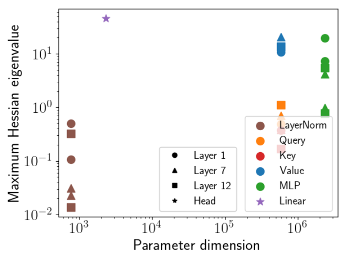


(i) ViT fine-tuned with SGD on Aircraft

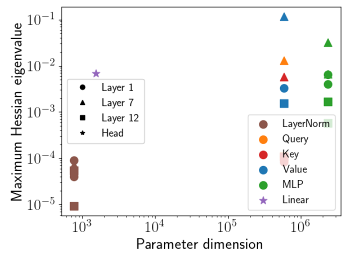
Figure 6: Gradient vs. Hessian matrix.

F.3 Correlation between Hessian and parameter dimension

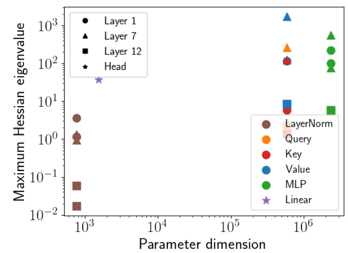
We show the correlation between the Hessian and the parameter in Figure 7. The Hessian and parameter dimension do not show a clear correlation.



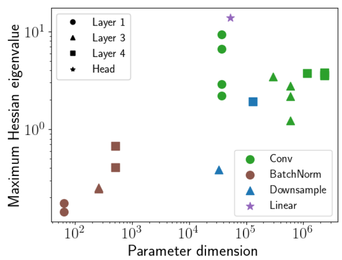
(a) Pre-trained RoBERTa on CB



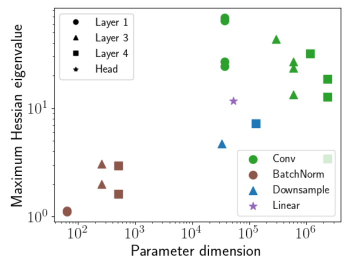
(b) RoBERTa fine-tuned with Adam on RTE



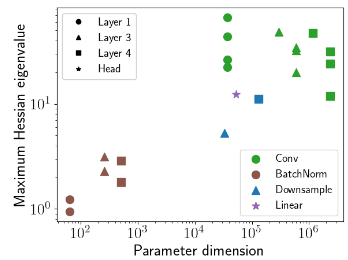
(c) RoBERTa fine-tuned with SGD on RTE



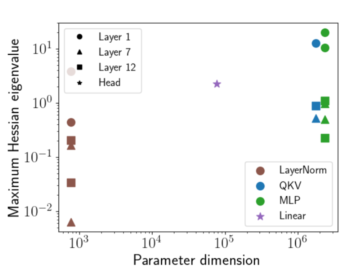
(d) Pre-trained ResNet18 on Flowers102



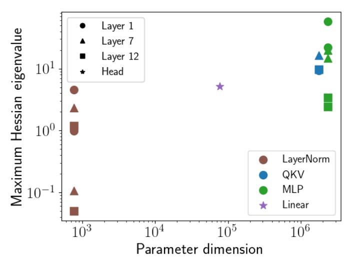
(e) ResNet18 fine-tuned with Adam on Flowers102



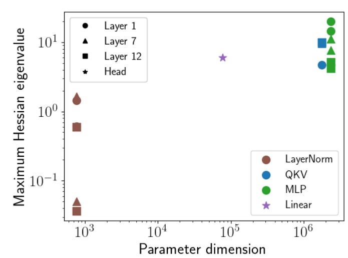
(f) ResNet18 fine-tuned with SGD on Flowers102



(g) Pre-trained ViT on Aircraft



(h) ViT fine-tuned with Adam on Aircraft

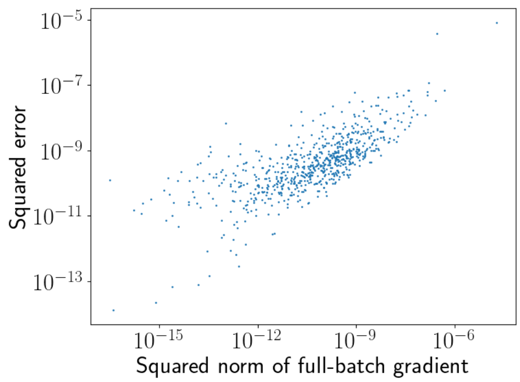


(i) ViT fine-tuned with SGD on Aircraft

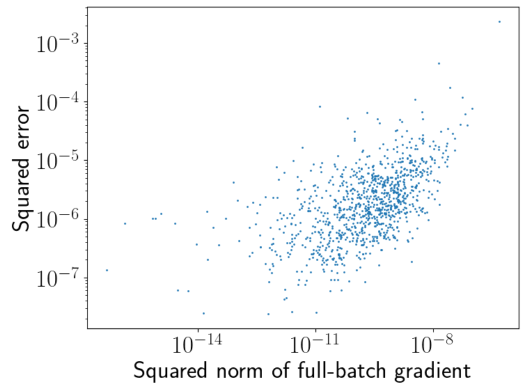
Figure 7: Parameter dimension vs. Hessian matrix.

F.4 Correlation between full-batch gradient and gradient error

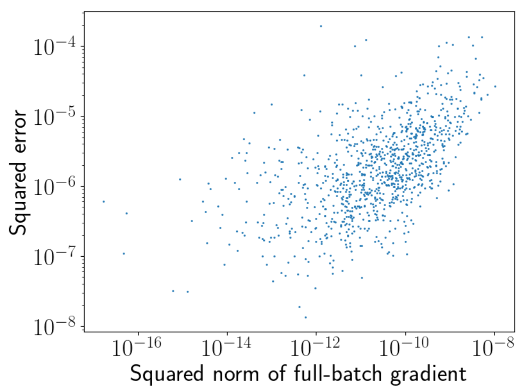
We show the correlation between the full-batch gradient and the gradient error in Figure 8.



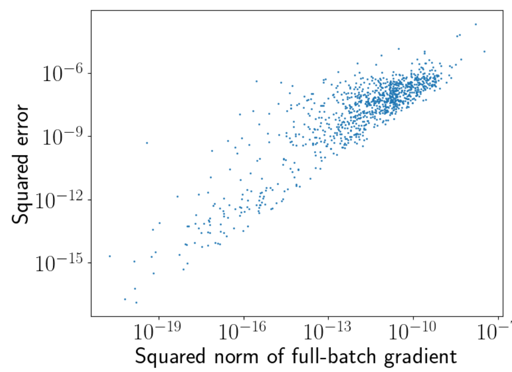
(a) Pre-trained RoBERTa on CB



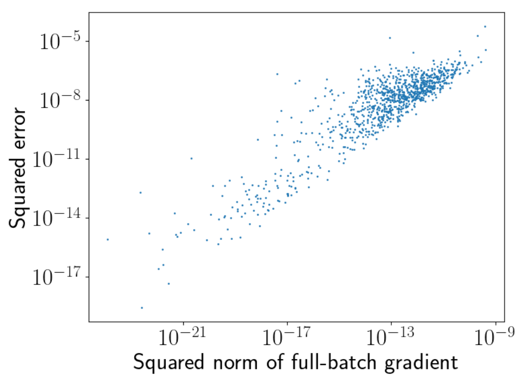
(b) Pre-trained ResNet18 on Flowers102



(c) Pre-trained ResNet18 on Aircraft



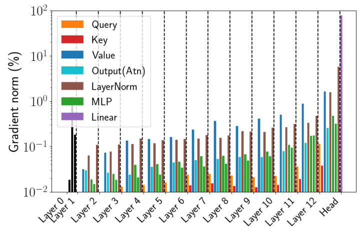
(d) Pre-trained ViT on Flowers102



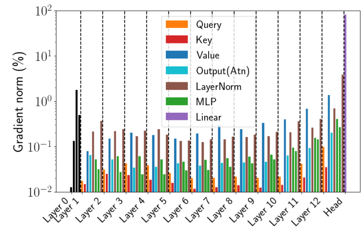
(e) Pre-trained ViT on Aircraft

Figure 8: coordinate-wise full-batch gradient vs. gradient error.

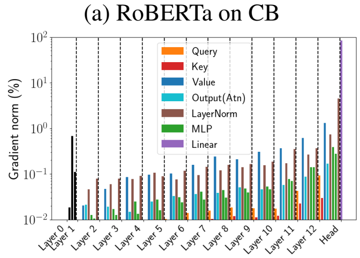
F.5 Gradient per parameter



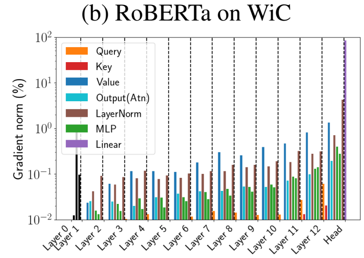
(a) RoBERTa on CB



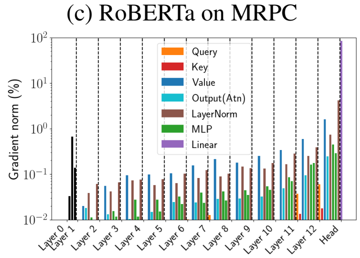
(b) RoBERTa on WiC



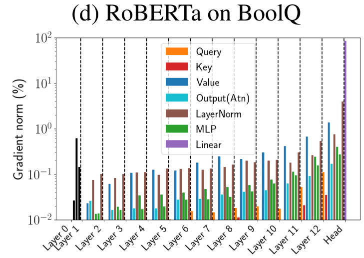
(c) RoBERTa on MRPC



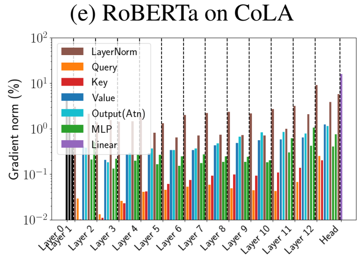
(d) RoBERTa on BoolQ



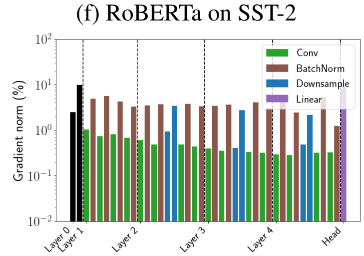
(e) RoBERTa on CoLA



(f) RoBERTa on SST-2



(g) ViT on Aircraft



(h) ResNet18 on Aircraft

Figure 9: Gradient norm of each parameter of pre-trained model.

F.6 QUANTITATIVE MEASURES OF GRADIENT HETEROGENEITY

Gini coefficient. In Table 9, we provide the Gini coefficient of the normalized gradients.

Gini coefficient is a measure of statistical dispersion intended to represent the inequality of a distribution, which ranges from 0 to 1 and the higher value indicates more heterogeneity.

Given a set of values $\{x_1, x_2, \ldots, x_n\}$ sorted in non-decreasing order, the Gini coefficient is defined as:

$$G = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |x_i - x_j|}{2n^2 \bar{x}},$$

where $\bar{x}$ is the mean of the values.

Layer-wise gradient norm ratio. In Table 10, we present the ratio of the gradient norm for each layer, computed as:

$$\frac{G_l}{\sum_{l'} G_{l'}},$$

where G_l represents the sum of the normalized full-batch gradient norms of the parameters in layer l. Since all layers contain the same number of parameters, this comparison is valid.

Model (Dataset)	Gini coefficient
RoBERTa-Base (CB)	0.932 ± 0.006
RoBERTa-Base (RTE)	0.944 ± 0.005
RoBERTa-Base (WiC)	0.931 ± 0.004
RoBERTa-Base (BoolQ)	0.944 ± 0.001
RoBERTa-Base (CoLA)	0.954 ± 0.003
RoBERTa-Base (MRPC)	0.951 ± 0.001
RoBERTa-Base (SST-2)	0.930 ± 0.032
ResNet-18 (Flowers102)	0.407 ± 0.013
ResNet-18 (Aircraft)	0.433 ± 0.005
ViT-Base (Flowers102)	0.539 ± 0.004
ViT-Base (Aircraft)	0.598 ± 0.009

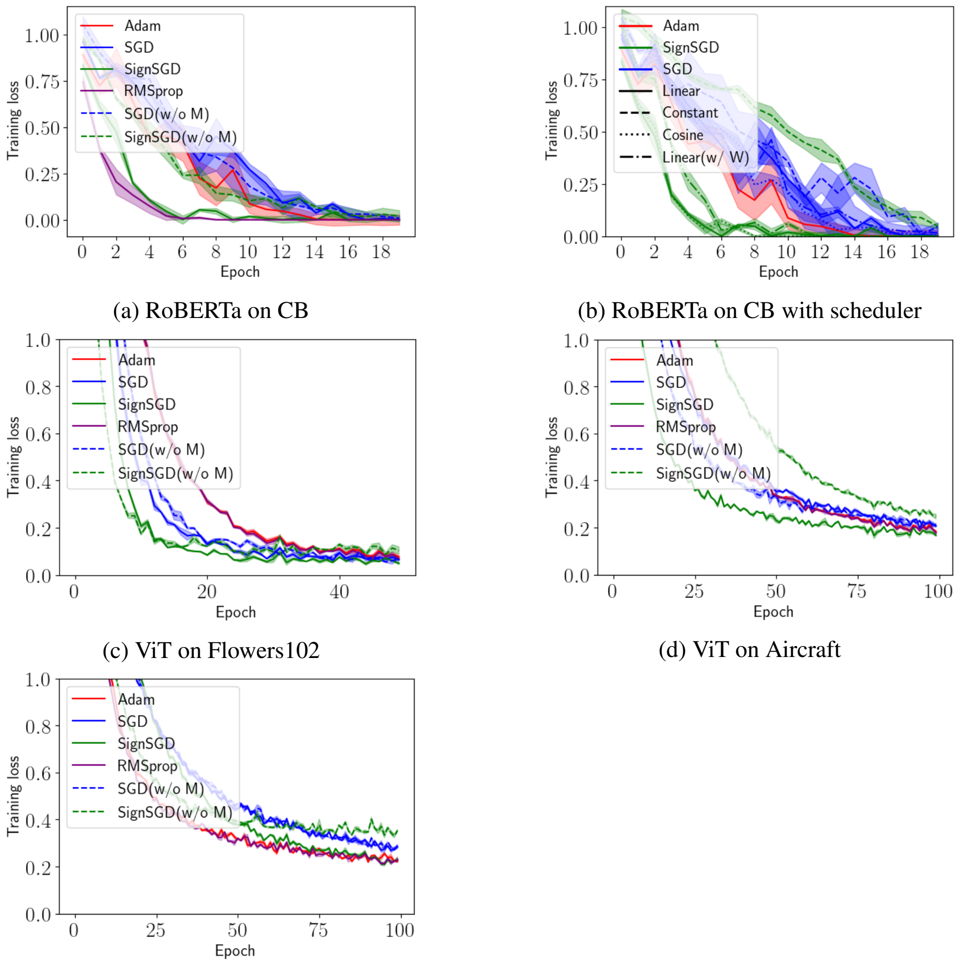
Table 9: Gini coefficient of normalized gradients. ± represents standard deviation.

Layer	1	2	3	4	5	6	7	8	9	10	11	12
RoBERTa-Base (CB)	0.021 ± 0.001	0.022 ± 0.001	0.027 ± 0.002	0.031 ± 0.002	0.036 ± 0.002	0.045 ± 0.002	0.054 ± 0.002	0.060 ± 0.003	0.070 ± 0.004	0.092 ± 0.005	0.156 ± 0.015	0.387 ± 0.027
RoBERTa-Base (RTE)	0.023 ± 0.003	0.024 ± 0.003	0.028 ± 0.003	0.030 ± 0.003	0.034 ± 0.002	0.042 ± 0.002	0.051 ± 0.004	0.058 ± 0.003	0.068 ± 0.003	0.093 ± 0.008	0.163 ± 0.014	0.387 ± 0.023
RoBERTa-Base (WiC)	0.047 ± 0.014	0.042 ± 0.010	0.041 ± 0.005	0.040 ± 0.003	0.036 ± 0.002	0.040 ± 0.003	0.049 ± 0.004	0.055 ± 0.004	0.063 ± 0.003	0.086 ± 0.006	0.145 ± 0.009	0.355 ± 0.035
RoBERTa-Base (BoolQ)	0.023 ± 0.001	0.024 ± 0.001	0.028 ± 0.001	0.031 ± 0.002	0.034 ± 0.002	0.043 ± 0.002	0.055 ± 0.003	0.062 ± 0.004	0.073 ± 0.004	0.098 ± 0.007	0.157 ± 0.010	0.370 ± 0.034
RoBERTa-Base (CoLA)	0.017 ± 0.001	0.018 ± 0.001	0.023 ± 0.003	0.025 ± 0.002	0.029 ± 0.002	0.037 ± 0.003	0.042 ± 0.002	0.048 ± 0.002	0.058 ± 0.003	0.083 ± 0.006	0.169 ± 0.013	0.451 ± 0.027
RoBERTa-Base (MRPC)	0.019 ± 0.002	0.020 ± 0.002	0.024 ± 0.002	0.028 ± 0.002	0.032 ± 0.002	0.040 ± 0.002	0.049 ± 0.003	0.057 ± 0.004	0.067 ± 0.004	0.089 ± 0.007	0.155 ± 0.010	0.421 ± 0.037
RoBERTa-Base (SST-2)	0.025 ± 0.010	0.026 ± 0.010	0.032 ± 0.012	0.036 ± 0.012	0.040 ± 0.013	0.046 ± 0.012	0.054 ± 0.014	0.061 ± 0.014	0.070 ± 0.009	0.087 ± 0.008	0.148 ± 0.022	0.373 ± 0.086
ViT-Base (Flowers102)	0.093 ± 0.004	0.065 ± 0.002	0.073 ± 0.002	0.071 ± 0.004	0.069 ± 0.003	0.071 ± 0.005	0.075 ± 0.005	0.079 ± 0.003	0.083 ± 0.005	0.094 ± 0.002	0.105 ± 0.005	0.122 ± 0.004
ViT-Base (Aircraft)	0.083 ± 0.005	0.058 ± 0.003	0.067 ± 0.003	0.063 ± 0.003	0.058 ± 0.002	0.063 ± 0.003	0.068 ± 0.001	0.073 ± 0.002	0.077 ± 0.003	0.090 ± 0.001	0.119 ± 0.005	0.181 ± 0.011

Table 10: Layer-wise ratio of gradient norms in transformers. ± represents standard deviation.

F.7 TRAIN CURVES

We show the training curves on different datasets from that in the main text. On the CB dataset, the final train loss is similar among all optimizers, but the convergence speed of SGD is slower than other optimizers. This is consistent with our analysis suggesting the difficulty of training of RoBERTa with SGD.



(a) RoBERTa on CB

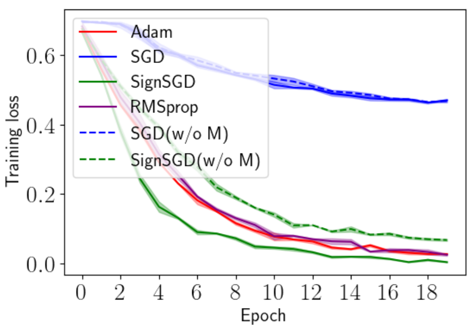
(b) RoBERTa on CB with scheduler

(c) ViT on Flowers102

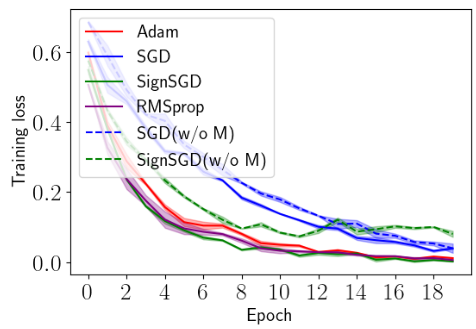
(d) ViT on Aircraft

(e) ResNet18 on Aircraft

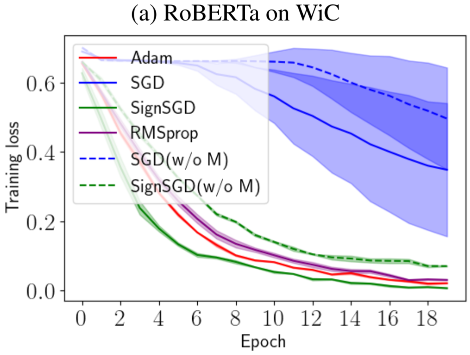
Figure 10: Training curve with different optimizers. w/ W indicates “with warmup”.



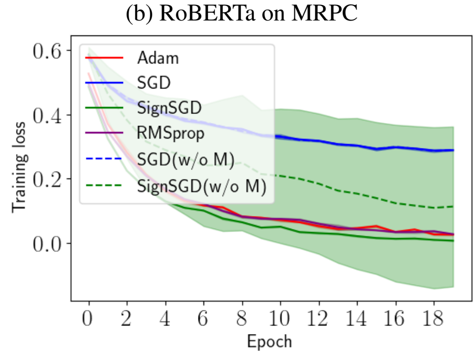
(a) RoBERTa on WiC



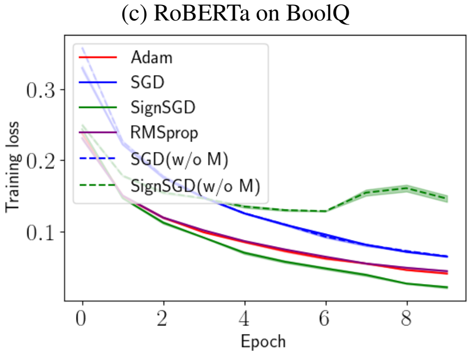
(b) RoBERTa on MRPC



(c) RoBERTa on BoolQ



(d) RoBERTa on CoLA



(e) RoBERTa on SST-2

Figure 11: Training curve with different optimizers.

F.8 TEST RESULTS

Table 11: Test results corresponding to the training curves shown in Figures 4 and 10. We report the accuracy and its standard deviation.

Model	Dataset	Adam	RMSprop	SGD	SignSGD	SGD(w/o M)	SignSGD(w/o M)
ViT-Base	Flowers102	95.06 ± 0.34	95.15 ± 0.41	94.22 ± 0.54	94.01 ± 0.98	94.49 ± 0.62	92.45 ± 1.35
	Aircraft	74.28 ± 0.59	74.86 ± 0.87	71.33 ± 0.27	73.96 ± 0.73	55.25 ± 0.67	75.21 ± 0.88
ResNet18	Flowers102	93.33 ± 0.62	93.27 ± 0.71	93.40 ± 0.47	94.43 ± 0.54	93.03 ± 0.62	93.10 ± 0.37
	Aircraft	71.95 ± 0.69	70.53 ± 0.42	72.66 ± 0.71	72.01 ± 0.40	72.16 ± 0.41	70.87 ± 0.35
RoBERTa-Base	CB	76.43 ± 7.41	84.29 ± 4.96	78.21 ± 6.36	83.21 ± 2.71	71.79 ± 12.46	77.86 ± 2.99
	RTE	75.88 ± 1.56	74.66 ± 2.89	75.31 ± 3.12	75.02 ± 2.30	73.21 ± 1.83	75.74 ± 2.74

F.9 EFFECT OF LAYER NORMALIZATION

Table 12: Gini coefficients of gradient norms for different normalization. A higher Gini coefficient indicates greater heterogeneity. “No-LN” refers to the architecture without layer normalization.

Norm Type	Init	Dataset	Gini Coefficient
No-LN	Scratch	RTE	0.867 ± 0.006
Pre-LN	Scratch	RTE	0.880 ± 0.004
Post-LN	Scratch	RTE	0.941 ± 0.012
Post-LN	Pre-trained	RTE	0.944 ± 0.005
No-LN	Scratch	CB	0.850 ± 0.049
Pre-LN	Scratch	CB	0.873 ± 0.017
Post-LN	Scratch	CB	0.899 ± 0.018
Post-LN	Pre-trained	CB	0.932 ± 0.006

F.10 APPLICABILITY BEYOND FINE-TUNING SETTINGS

To examine whether our findings generalize beyond fine-tuning scenarios, we conducted language modeling from scratch using nanoGPT on the Shakespeare dataset. The results indicate that Adam achieves better performance than SGD, while SignSGD also performs competitively. We additionally observed that gradient heterogeneity in nanoGPT is higher than in ViT and ResNet, but lower than in RoBERTa. Although this setup differs from fine-tuning, the findings are consistent with our overall analysis.

Table 13: Training loss for nanoGPT trained from scratch on the Shakespeare dataset. “Min” denotes the lowest observed loss during training, and “Last” denotes the final loss at the end of training.

Optimizer	Min	Last
Adam	0.658 ± 0.009	0.687 ± 0.019
SGD	0.928 ± 0.120	0.964 ± 0.122
SignSGD	0.791 ± 0.011	0.820 ± 0.017

Table 14: Gini coefficient of gradient norms for nanoGPT on the Shakespeare dataset. A higher Gini coefficient indicates greater gradient heterogeneity.

Model (Dataset)	Gini Coefficient
nanoGPT (Shakespeare)	0.609 ± 0.004

G Discussion on momentum in SignSGD

The impact of the momentum term used in Adam has not been considered in the analysis so far. However, in sample-wise training, the presence of a momentum term significantly affects the updates of the linear head, particularly for the bias term.

Proposition G.1 (SignSGD without momentum)**.** *Let $\Delta^S\theta$ and $\Delta^F\theta$ denote the one-epoch updates of a parameter θ during sample-wise and full-batch training, respectively. For a linear head trained using the cross-entropy loss and SignSGD with a learning rate η, the updates are as follows:*

For the bias term b_k:

$$\Delta^S b_k = -\frac{\eta}{N}\sum_{i=1}^{N}(1 - 2\cdot\mathbb{1}[y^{(i)} = k]), \quad \Delta^F b_k = -\eta\,\mathrm{sign}\left(\sum_{i=1}^{N}\delta_{p_k}^{(i)}\right),$$

and for the weight matrix $V_{k,l}$:

$$\Delta^S V_{k,l} = -\frac{\eta}{N}\left(\sum_{y^{(i)}\neq k} s_l^{(i)} - \sum_{y^{(i)}=k} s_l^{(i)}\right), \quad \Delta^F V_{k,l} = -\eta\,\mathrm{sign}\left(\sum_{i=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}^{(i)})_l\delta_{p_k}^{(i)}\right),$$

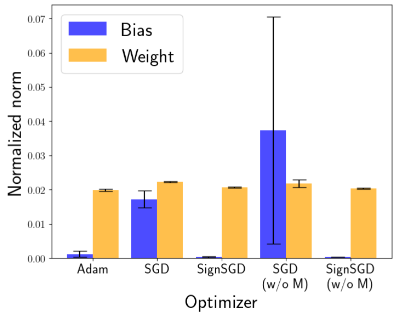
where $\delta_{p_k}^{(i)} := \mathrm{softmax}(\boldsymbol{f}(\boldsymbol{x}^{(i)}))_k - \mathbb{1}[k = y^{(i)}]$ represents the prediction error for the i-th sample and class k and $s_l^{(i)} := \mathrm{sign}\left(\boldsymbol{\phi}(\boldsymbol{x}^{(i)})_l\right)$ is the sign of the l-th element of the feature embedding $\boldsymbol{\phi}(\boldsymbol{x}^{(i)})_l$.

Sign-alignment causes large updates. In full-batch training, the updates $\Delta^{\mathrm{F}} b_k$ and $\Delta^{\mathrm{F}} V_{k,l}$ depend on the model predictions. Because the signs of these updates vary across epochs, these updates remain small. In contrast, in sample-wise training, update signs can align across epochs, resulting in disproportionately large updates. This effect is particularly pronounced for the bias term $\Delta^{\mathrm{S}} b_k$, which is independent of model predictions and grows with the number of classes. Similarly, the sign of $\Delta^{\mathrm{S}} V_{k,l}$, which depends on the feature extractor output $\boldsymbol{\phi}(\boldsymbol{x}^{(i)})$, may align across epochs.

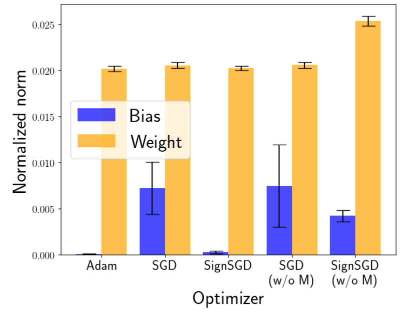
Momentum resolves the issue. Excessively large updates can cause training instability and incorrect predictions. Although the proposition specifically addresses sample-wise updates, similar challenges can arise in batch training. Momentum, which estimates the full-batch gradient using exponential moving averages, effectively mitigates this problem.

G.1 Experimental results

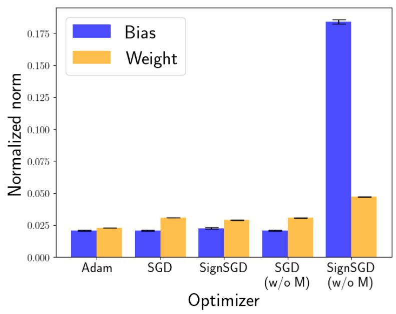
We show the norm of the linear head for different datasets, models, and optimizers. The results indicate that when the number of classes is large, the bias term of the linear head exhibits a larger norm with SignSGD without momentum compared to other optimizers. In contrast, the weight norm does not necessarily increase under the same conditions, even with SignSGD without momentum. This observation aligns with the theoretical analysis in Proposition G.1, which suggests that a large number of classes leads to an increase in the bias term norm, while the weight norm is influenced by the sign of the feature extractor outputs.



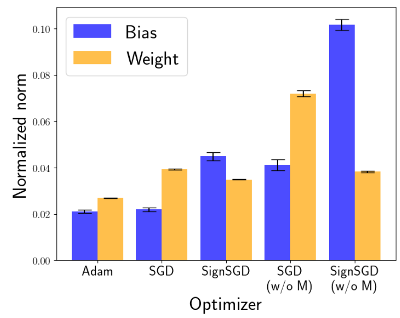
(a) RoBERTa on CB



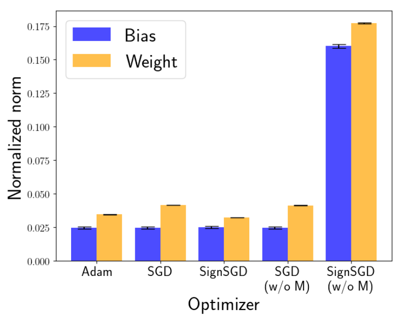
(b) RoBERTa on RTE



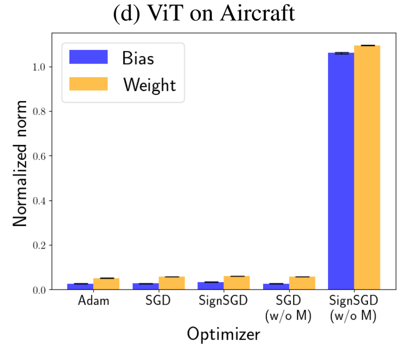
(c) ViT on Flowers102



(d) ViT on Aircraft



(e) ResNet18 on Flowers102



(f) ResNet18 on Aircraft

Figure 12: Norm of the linear head.

G.2 Proof of Proposition G.1

Proof. The partial derivative of the bias and the weight matrix with the cross-entropy loss is given by:

$$
\begin{aligned}
\frac{\partial \ell(\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}))}{\partial b_k} &= \frac{\partial \ell(\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}))}{\partial \boldsymbol{f}(\boldsymbol{x}^{(i)})} \frac{\partial \boldsymbol{f}(\boldsymbol{x}^{(i)})}{\partial b_k} \\
&= \frac{\partial \ell(\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}))}{\partial \boldsymbol{f}(\boldsymbol{x}^{(i)})} \frac{\partial \boldsymbol{V}\boldsymbol{\phi}(\boldsymbol{x}^{(i)}) + \boldsymbol{b}}{\partial b_k} \\
&= (\mathrm{softmax}(\boldsymbol{f}(\boldsymbol{x}^{(i)})) - \boldsymbol{e}^{(y^{(i)})})^\top \boldsymbol{e}^{(k)} \\
&= \mathrm{softmax}(\boldsymbol{f}(\boldsymbol{x}^{(i)}))_k - \mathbb{1}[k = y^{(i)}] \\
\frac{\partial \ell(\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}))}{\partial V_{k,l}} &= \frac{\partial \ell(\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}))}{\partial \boldsymbol{f}(\boldsymbol{x}^{(i)})} \frac{\partial \boldsymbol{V}\boldsymbol{\phi}(\boldsymbol{x}^{(i)}) + \boldsymbol{b}}{\partial V_{k,l}} \\
&= (\mathrm{softmax}(\boldsymbol{f}(\boldsymbol{x}^{(i)})) - \boldsymbol{e}^{(y^{(i)})})^\top \boldsymbol{\phi}(\boldsymbol{x}^{(i)})_l \boldsymbol{e}^{(k)} \\
&= \boldsymbol{\phi}(\boldsymbol{x}^{(i)})_l (\mathrm{softmax}(\boldsymbol{f}(\boldsymbol{x}^{(i)}))_k - \mathbb{1}[k = y^{(i)}])
\end{aligned}
$$

The one-epoch updates of the bias and the weight matrix with the sample-wise training are given by:

$$\begin{aligned}
\Delta^{\mathrm{S}} b_k &= -\frac{\eta}{N}\sum_{i=1}^{N}\operatorname{sign}\left(\frac{\partial \ell(\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}))}{\partial b_k}\right) \\
&= -\frac{\eta}{N}\sum_{i=1}^{N}\operatorname{sign}\left(\operatorname{softmax}(\boldsymbol{f}(\boldsymbol{x}^{(i)}))_k - \mathbb{1}[k = y^{(i)}]\right) \\
&= -\frac{\eta}{N}\sum_{i=1}^{N}(1 - 2\cdot \mathbb{1}[y^{(i)} = k])
\end{aligned}$$

and

$$\begin{aligned}
\Delta^{\mathrm{S}} V_{k,l} &= -\frac{\eta}{N}\sum_{i=1}^{N}\operatorname{sign}\left(\frac{\partial \ell(\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}))}{\partial V_{k,l}}\right) \\
&= -\frac{\eta}{N}\sum_{i=1}^{N}\operatorname{sign}\left(\boldsymbol{\phi}(\boldsymbol{x}^{(i)})_l(\operatorname{softmax}(\boldsymbol{f}(\boldsymbol{x}^{(i)}))_k - \mathbb{1}[k = y^{(i)}])\right) \\
&= -\frac{\eta}{N}\sum_{i=1}^{N}\operatorname{sign}\left(\boldsymbol{\phi}(\boldsymbol{x}^{(i)})_l\right)\operatorname{sign}\left(\operatorname{softmax}(\boldsymbol{f}(\boldsymbol{x}^{(i)}))_k - \mathbb{1}[k = y^{(i)}]\right) \\
&= -\frac{\eta}{N}\left(\sum_{y^{(i)}\neq k}\operatorname{sign}\left(\boldsymbol{\phi}(\boldsymbol{x}^{(i)})_l\right) - \sum_{y^{(i)}= k}\operatorname{sign}\left(\boldsymbol{\phi}(\boldsymbol{x}^{(i)})_l\right)\right)
\end{aligned}$$

The one-epoch updates of the bias and the weight matrix with the full-batch training are given by:

$$\begin{aligned}
\Delta^{\mathrm{F}} b_k &= -\eta\operatorname{sign}\left(\frac{1}{N}\sum_{i=1}^{N}\frac{\partial \ell(\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}))}{\partial b_k}\right) \\
&= -\eta\operatorname{sign}\left(\frac{1}{N}\sum_{i=1}^{N}\left(\operatorname{softmax}(\boldsymbol{f}(\boldsymbol{x}^{(i)}))_k - \mathbb{1}[k = y^{(i)}]\right)\right) \\
&= -\eta\operatorname{sign}\left(\sum_{i=1}^{N}\delta_{p_k}^{(i)}\right)
\end{aligned}$$

and

$$\begin{aligned}
\Delta^{\mathrm{F}} V_{k,l} &= -\eta\operatorname{sign}\left(\frac{1}{N}\sum_{i=1}^{N}\frac{\partial \ell(\boldsymbol{f}(\boldsymbol{x}^{(i)}, y^{(i)}))}{\partial V_{k,l}}\right) \\
&= -\eta\operatorname{sign}\left(\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}^{(i)})_l(\operatorname{softmax}(\boldsymbol{f}(\boldsymbol{x}^{(i)}))_k - \mathbb{1}[k = y^{(i)}])\right) \\
&= -\eta\operatorname{sign}\left(\sum_{i=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}^{(i)})_l\delta_{p_k}^{(i)}\right).
\end{aligned}$$

□

H MORE DISCUSSION ON TRANSFORMERS

In this section, we provide additional discussion on the gradient heterogeneity in transformers, focusing on the self-attention mechanism.

Additional notation. The k-th standard basis vector is denoted by $\boldsymbol{e}^{(k)}$ with $\boldsymbol{e}_l^{(k)} = \delta_{kl}$, where δ_{kl} is the Kronecker delta. Function $\text{vec}(\cdot)$ denotes row-wise vectorization. Frobenius norm and the Kronecker product is denoted by $\|\cdot\|_F$ and $\otimes$, respectively.

H.1 TRANSFORMER ARCHITECTURE

The transformer architecture (Vaswani et al., 2017) relies on the self-attention mechanism, which assigns importance to each token in the input sequence.

For an input sequence of n tokens, each of dimension d, represented by $\boldsymbol{X} \in \mathbb{R}^{n\times d}$, single-head self-attention is defined as:

$$\text{SA}(\boldsymbol{X}) \coloneqq \text{softmax}\left(\frac{\boldsymbol{X}\boldsymbol{W}_Q(\boldsymbol{X}\boldsymbol{W}_K)^\top}{\sqrt{d_k}}\right)\boldsymbol{X}\boldsymbol{W}_V,$$

where $\boldsymbol{W}_Q, \boldsymbol{W}_K \in \mathbb{R}^{d\times d_k}$ and $\boldsymbol{W}_V \in \mathbb{R}^{d\times d_v}$ are learnable projection matrices for queries, keys, and values, respectively. Multi-head attention concatenates the outputs of parallel single-head self-attention mechanisms and applies a linear transformation, followed by a feed-forward network.

H.2 GRADIENT OF SELF-ATTENTION MECHANISM

We analyze the gradients in self-attention, focusing on the value and query/key weight matrices. Using Lemma A.2 from Noci et al. (2022), the Frobenius norms of these gradients are:

$$\begin{aligned}\|\frac{\partial\,\text{SA}(\boldsymbol{X})}{\partial \boldsymbol{W}_V}\|_F &= \|\boldsymbol{P}\boldsymbol{X}\otimes \boldsymbol{I}_{d_v}\|_F \\ &\le \underbrace{\sqrt{d_v}\|\boldsymbol{P}\|_F\|\boldsymbol{X}\|_F}_{=:\mathcal{U}_V},\end{aligned} \tag{16}$$

$$\begin{aligned}&\|\frac{\partial\,\text{SA}(\boldsymbol{X})}{\partial \boldsymbol{W}_Q}\|_F \\ =&\|(\boldsymbol{I}_n\otimes \boldsymbol{W}_V\boldsymbol{X}^\top)\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{M}}\frac{\boldsymbol{X}\otimes\boldsymbol{X}\boldsymbol{W}_K}{\sqrt{d_k}}\|_F \\ \le& \underbrace{\sqrt{n}\|\boldsymbol{W}_V\boldsymbol{X}^\top\|_F\|\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{M}}\|_F\frac{\|\boldsymbol{X}\|_F\|\boldsymbol{X}\boldsymbol{W}_K\|_F}{\sqrt{d_k}}}_{=:\mathcal{U}_Q},\end{aligned} \tag{17}$$

where $\boldsymbol{M} \coloneqq \boldsymbol{X}\boldsymbol{W}_Q\boldsymbol{W}_K^\top\boldsymbol{X}^\top/\sqrt{d_k}$, $\boldsymbol{P} \coloneqq \text{softmax}(\boldsymbol{M})$, and $\mathcal{U}_V$ and $\mathcal{U}_Q$ represent the upper bounds for the gradients of the value and query weight matrices, respectively. The derivation of the gradient for the key weight matrix is omitted, as it is analogous to that of the query weight matrix.

Focusing on the attention matrix $\boldsymbol{P}$, we derive the following result.

Proposition H.1 (Gradients and attention matrices). *In transformers, one-hot attention matrices uniquely maximize the upper bound of the Frobenius norm of the gradient with respect to the value weight matrix $\mathcal{U}_V$ and uniquely minimize that with respect to the query weight matrix $\mathcal{U}_Q$, as follows:*

$$\arg\max_{\boldsymbol{P}}\mathcal{U}_V = \arg\min_{\boldsymbol{P}}\mathcal{U}_Q = \mathcal{P}_{\textit{one-hot}},$$

where

$$\mathcal{P}_{\textit{one-hot}} \coloneqq \{\boldsymbol{P} \mid \forall i,\ \exists k_i\ s.t.\ \boldsymbol{P}_{i,:} = \boldsymbol{e}^{(k_i)}\}$$

is the set of one-hot matrices.

The proof of the proposition is provided in Appendix H.4. The statement about the query weight matrix also applies to the key weight matrix due to their analogous gradients. The proposition demonstrates that the gradients of the value and query/key weight matrices exhibit opposing behaviors with respect to one-hot attention matrices: the gradient of the value weight matrix is maximized, while those of the query/key weight matrices are minimized.

Previous studies (Noci et al., 2022; Wang et al., 2021) observed that the gradient of the value weight matrix is typically larger than those of the query/key weight matrices, consistent with our experimental findings in Section 5.2. Together with Proposition H.1, these results suggest that attention matrices close to one-hot amplify gradient heterogeneity in the self-attention mechanism.

H.3 UNIFORMITY OF THE ATTENTION MATRIX

In Figure 13, we compare the attention matrices of pre-trained RoBERTa and ViT. The attention matrix of ViT is more uniform than that of RoBERTa, reflecting the differences between NLP and vision tasks. In NLP, the use of special tokens and stronger interrelations between input tokens lead to less uniform attention, with only a few tokens receiving attention (Clark, 2019). Conversely, vision tasks, which prioritize holistic information (Torralba, 2003; Rabinovich et al., 2007; Shotton et al., 2009), produce more uniform attention matrices, where all tokens are attended to. This observation aligns with Hyeon-Woo et al. (2023), who also reported uniform attention matrices in ViT. Notably, more uniform attention matrices are farther from one-hot matrices, indicating reduced dominance by individual tokens.

Combined with the analysis in Appendix H.2, which shows that attention matrices closer to one-hot matrices amplify gradient heterogeneity, this suggests that gradient heterogeneity in the self-attention mechanism is more pronounced in NLP tasks than in vision tasks.

H.4 PROOF OF PROPOSITION H.1

Proof of $\mathcal{U}_V$. As defined in Eq.(16), the upper bound of the gradient is given by:

$$\mathcal{U}_V = \sqrt{d_v}\|\boldsymbol{P}\|_F\|\boldsymbol{X}\|_F.$$

We observe that:

$$\begin{aligned}\arg\max_{\boldsymbol{P}} \mathcal{U}_V &= \arg\max_{\boldsymbol{P}} \|\boldsymbol{P}\|_F \\ &= \arg\max_{\boldsymbol{P}} \|\boldsymbol{P}\|_F^2 \\ &= \arg\max_{\boldsymbol{P}} \sum_{i=1}^{n} \|\boldsymbol{P}_{i,:}\|_2^2.\end{aligned}$$

Since the rows of the attention matrix are independent, we focus on the i-th row. The i-th row of the attention matrix satisfies the following constraints:

$$1 \le j \le n, \quad P_{i,j} \ge 0, \quad \sum_{j=1}^{n} P_{i,j} = 1.$$

We define the Lagrangian function as:

$$\mathcal{L}_V = -\sum_{j=1}^{n} P_{i,j}^2 - \sum_{j=1}^{n} \mu_j P_{i,j} + \lambda(\sum_{j=1}^{n} P_{i,j} - 1),$$

where λ and μ_j are the Lagrange multipliers. To minimize the Lagrangian function, the solution must satisfy the following KKT conditions:

$$\frac{\partial \mathcal{L}_V}{\partial P_{i,j}} = -2P_{i,j} - \mu_j + \lambda = 0, \quad 1 \leq j \leq n, \tag{18}$$

$$\sum_{j=1}^{n} P_{i,j} - 1 = 0, \tag{19}$$

$$P_{i,j} \geq 0, \quad 1 \leq j \leq n, \tag{20}$$

$$\mu_j \geq 0, \quad 1 \leq j \leq n, \tag{21}$$

$$\mu_j P_{i,j} = 0, \quad 1 \leq j \leq n. \tag{22}$$

From Equations (19) and (20), it follows that $P_{i,j} > 0$ for some j. Let k $(1 \leq k \leq n)$ denote the number of non-zero elements in $\boldsymbol{P}_{i,:}$, and suppose $P_{i,j_l} > 0$ for $1 \leq l \leq k$. From Equation (22), we have $\mu_{j_l} = 0$, and thus, from Equation (18), we deduce that $P_{i,j_l} = \frac{\lambda}{2}$ for $1 \leq l \leq k$. Using Equation (19), we get $\sum_{l=1}^{k} \frac{\lambda}{2} = 1$, which gives $\lambda = 2/k$. For $j \notin \{j_l \mid 1 \leq l \leq k\}$, we have $P_{i,j} = 0$ and $\mu_j = \lambda = 2/k$, satisfying Eq.(21).

With k non-zero elements of $\boldsymbol{P}_{i,:}$, the value of the Lagrangian function becomes $-\sum_{j=1}^{n} P_{i,j}^2 = -\sum_{l=1}^{k} (\frac{\lambda}{2})^2 = -\frac{\lambda^2}{4}k = -\frac{1}{k}$. The minimum value of the Lagrangian function is achieved if and only if $k = 1$, which implies $\boldsymbol{P}_{i,:} = \boldsymbol{e}^{(k_i)}$ for some k_i. Therefore, we conclude:

$$\arg\max_{\boldsymbol{P}} \mathcal{U}_V = \{\boldsymbol{P} \mid \forall i, \ \exists k_i \ s.t. \ \boldsymbol{P}_{i,:} = \boldsymbol{e}^{(k_i)}\}.$$

□

Proof of $\mathcal{U}_Q$. As defined in Eq.(17), the upper bound of the gradient is given by:

$$\mathcal{U}_Q = \sqrt{n} \|\boldsymbol{W}_V \boldsymbol{X}^\top\|_F \|\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{M}}\|_F \frac{\|\boldsymbol{X}\|_F \|\boldsymbol{X}\boldsymbol{W}_K\|_F}{\sqrt{d_k}}.$$

The partial derivative is expressed as:

$$\begin{aligned}
\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{M}} &= \frac{\partial \text{softmax}(\boldsymbol{M})}{\partial \boldsymbol{M}} \\
&= \text{blockdiag}(\{\frac{\partial \text{softmax}(\boldsymbol{M}_{i,:})}{\partial \boldsymbol{M}_{i,:}}\}_{i=1}^{n}) \\
&= \text{blockdiag}(\{\text{diag}(\boldsymbol{P}_{i,:}) - \boldsymbol{P}_{i,:}\boldsymbol{P}_{i,:}^\top\}_{i=1}^{n}).
\end{aligned}$$

Considering the attention matrix $\boldsymbol{P}$, we obtain:

$$\begin{aligned}
\arg\min_{\boldsymbol{P}} \mathcal{U}_Q &= \arg\min_{\boldsymbol{P}} \|\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{M}}\|_F \\
&= \arg\min_{\boldsymbol{P}} \sum_{i=1}^{n} \| \text{diag}(\boldsymbol{P}_{i,:}) - \boldsymbol{P}_{i,:}\boldsymbol{P}_{i,:}^\top\|_F^2.
\end{aligned}$$

As in the proof of $\mathcal{U}_V$, we focus on the value of the i-th row:

$$\| \text{diag}(\boldsymbol{P}_{i,:}) - \boldsymbol{P}_{i,:}\boldsymbol{P}_{i,:}^\top\|_F^2 = \sum_{j=1}^{n} (P_{i,j} - P_{i,j}^2)^2 + \sum_{j \neq l} P_{i,j}^2 P_{i,l}^2,$$

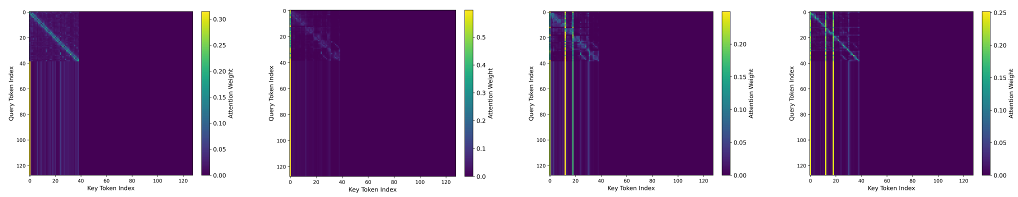
subject to the constraints $1 \leq j \leq n, \quad P_{i,j} \geq 0, \quad \sum_{j=1}^{n} P_{i,j} = 1$. Since both the first term and the second term are non-negative, the minimum value is attained if and only if both terms are 0. This condition is satisfied if $\boldsymbol{P}_{i,:}$ is a one-hot vector. Conversely, if $\boldsymbol{P}_{i,:}$ is not a one-hot vector, the second term becomes positive, and the minimum value cannot be attained. Thus, we have shown that the minimum value of the objective function is achieved if and only if $\boldsymbol{P}_{i,:}$ is a one-hot vector. Therefore:

$$\arg\min_{\boldsymbol{P}} \mathcal{U}_Q = \{\boldsymbol{P} \mid \forall i, \ \exists k_i \ s.t. \ \boldsymbol{P}_{i,:} = \boldsymbol{e}^{(k_i)}\}.$$

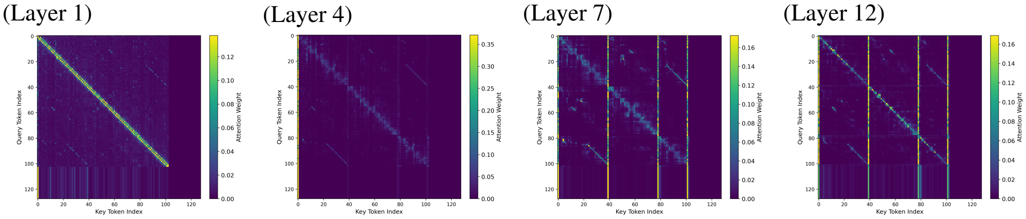
□

H.5 EXPERIMENTAL RESULTS

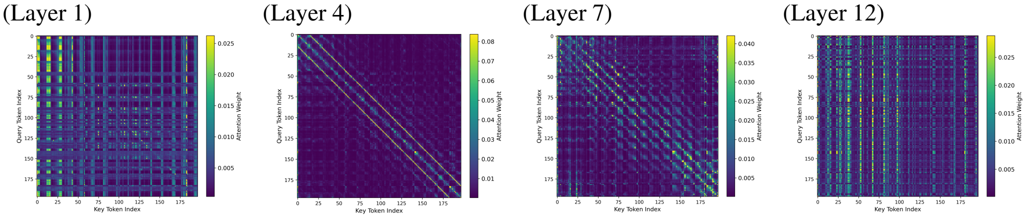
Heatmap of attention matrices. In Figure 13, we show the attention matrices computed from pre-trained models. These matrices are calculated for a randomly sampled sequ ence from the training data and are averaged across all heads.



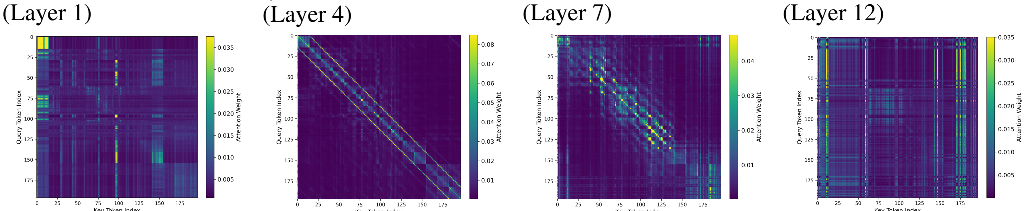
(a) RoBERTa on CB (Layer 1) (b) RoBERTa on CB (Layer 4) (c) RoBERTa on CB (Layer 7) (d) RoBERTa on CB (Layer 12)



(e) RoBERTa on RTE (Layer 1) (f) RoBERTa on RTE (Layer 4) (g) RoBERTa on RTE (Layer 7) (h) RoBERTa on RTE (Layer 12)



(i) ViT on Flowers102 (Layer 1) (j) ViT on Flowers102 (Layer 4) (k) ViT on Flowers102 (Layer 7) (l) ViT on Flowers102 (Layer 12)



(m) ViT on Aircraft (Layer 1) (n) ViT on Aircraft (Layer 4) (o) ViT on Aircraft (Layer 7) (p) ViT on Aircraft (Layer 12)

Figure 13: Attention matrices of the pre-trained RoBERTa and ViT.

Gradient and entropy of attention matrices. In Figure 14 (a) and (c), we show the ratio of the mean entropy relative to the maximum entropy of the attention matrix for each layer of the transformer model. Error bars indicate the standard deviation. Specifically, we plot:

$$\frac{1}{HNS}\sum_{h=1}^{H}\sum_{i=1}^{N}\sum_{s=1}^{S}\left(\sum_{j=1}^{S} A_{s,j}^{(i,h,l)} \log(A_{s,j}^{(i,h,l)})/\log(S)\right),$$

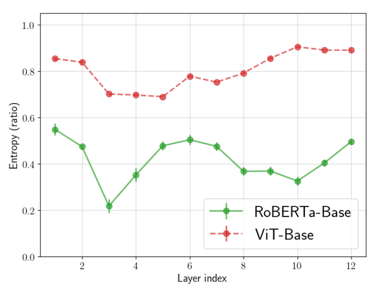
for each layer l, where H is the number of heads, S is the sequence length, and $\boldsymbol{A}^{(i,h,l)} \in \mathbb{R}^{S\times S}$ is the attention matrix of the h-th head in the l-th layer for sample $\boldsymbol{x}^{(i)}$.

In Figure 14 (b) and (d), we show the ratio of the mean gradient norm relative to the sum of the gradient norms of the attention matrix for each layer. Specifically, we plot:

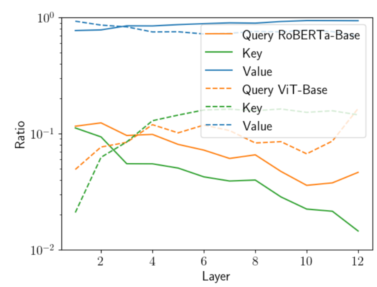
$$\frac{G_p^{(l)}}{G_Q^{(l)} + G_K^{(l)} + G_V^{(l)}},$$

for each layer l and $p \in \{Q, K, V\}$, where $G_Q^{(l)}$, $G_K^{(l)}$, and $G_V^{(l)}$ are the full-batch gradient norms of the query, key, and value weight matrices in the l-th layer of the transformer model, respectively.

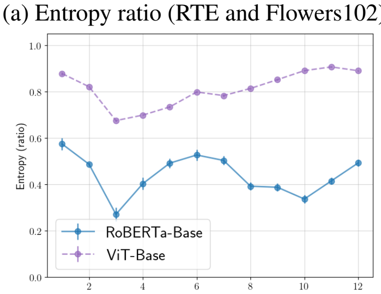
The results show that the entropy of the attention matrix is higher in RoBERTa than in ViT, and the gradient norm of the attention matrix is more heterogeneous in RoBERTa than in ViT. This observation is consistent with the theoretical analysis in Appendix H.3.



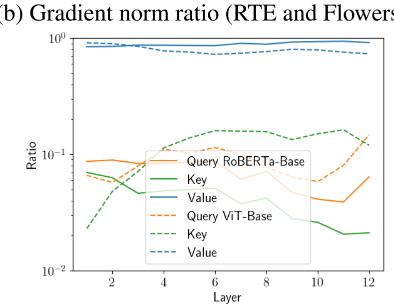
(a) Entropy ratio (RTE and Flowers102)



(b) Gradient norm ratio (RTE and Flowers102)



(c) Ratio of entropy (CB and Aircraft)



(d) Ratio of gradient norm (CB and Aircraft)

Figure 14: Comparison of entropy and gradient norms in attention matrices for RoBERTa and ViT. (a) and (c): the ratio of entropy relative to the maximum possible entropy. (b) and (d): the ratio of the gradient norm for self-attention parameters relative to the total gradient norm.

I More discussion on the sign-based sequence in stochastic settings

In this section, we further examine the iteration complexity of the sign-based sequence under stochastic settings. Specifically, we present iteration complexity results that account for a learning rate adapted to the noise level.

Theorem I.1. *Assume that* $\delta_D < \Lambda_P/3,\ \varepsilon < \frac{5\Lambda_P^2}{3(1-2\sigma_2)\rho_H\sqrt{P}}$*, and* $\sigma_2 < \frac{1}{2}$ *hold and that the learning rate at time* t *satisfies* $\eta_t = \zeta_t \min(\frac{3(1-2\sigma_2)\|\nabla L(\boldsymbol{\theta}_t^{Sign})\|_1}{5\Lambda_P P}, \sqrt{\frac{3(1-2\sigma_2)\|\nabla L(\boldsymbol{\theta}_t^{Sign})\|_1}{5\rho_H P^{3/2}}})$*, where* $\zeta_t \in [\zeta_0, 1]$*. Then, the iteration complexity for the sign-based sequence in stochastic settings are bounded as follows.*

$$\mathcal{T}_\varepsilon(\{\boldsymbol{\theta}_t^{Sign}\}_{t=0}^\infty, L, \|\cdot\|_1) \leq \frac{20(L(\boldsymbol{\theta}_0) - L_*)}{3(1-2\sigma_2)^2 P\varepsilon^2\zeta_0}\Lambda_P.$$

Proof. We start with Eq. (15) in Appendix C.3. Let $\varepsilon < \frac{\alpha\Lambda_P^2}{\rho_H\sqrt{P}}$ and set the learning rate as $\eta_t = \zeta_t \min(\frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1}{\alpha\Lambda_P P}, \sqrt{\frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1}{\alpha\rho_H P^{3/2}}})$, where $\zeta_t \in [\zeta_0, 1]$ and $\alpha > \frac{5}{6(1-2\sigma_2)}$. Then, we have:

$$\begin{aligned}
&\mathbb{E}\left[L(\boldsymbol{\theta}_{t+1}^{\text{Sign}}) - L(\boldsymbol{\theta}_t^{\text{Sign}}) \mid \boldsymbol{\theta}_t^{\text{Sign}}\right] \\
&\leq -\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 + \frac{\eta_t^2}{2}\Lambda_P P + \frac{\eta_t^2}{2}\delta_D P + \eta_t^3\frac{\rho_H}{6}P^{3/2} + 2\sigma_2\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 \\
&\leq -\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 + \frac{\eta_t}{2\alpha}\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 + \frac{\eta_t}{6\alpha}\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 + \frac{\eta_t}{6\alpha}\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 + 2\sigma_2\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 \\
&\qquad \text{(From } \eta_t \leq \min(\frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1}{\alpha\Lambda_P P}, \sqrt{\frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1}{\alpha\rho_H P^{3/2}}}) \text{ and } \delta_D < \Lambda_P/3) \\
&= -\frac{(6\alpha(1-2\sigma_2)-5)\eta_t}{6\alpha}\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1
\end{aligned}$$

Assume that the probability of the event $\mathcal{E}(T) = \{\forall s \leq T,\ \|\nabla L(\boldsymbol{\theta}_s^{\text{Sign}})\|_1 \geq P\varepsilon\}$ satisfies $\mathbb{P}(\mathcal{E}(T)) \geq \frac{1}{2}$. By applying the telescoping sum and taking expectations, and noting that $\boldsymbol{\theta}_0 = \boldsymbol{\theta}_0^{\text{Sign}}$, we have:

$$\begin{aligned}
&\mathbb{E}\left[L(\boldsymbol{\theta}_T^{\text{Sign}})\right] - L(\boldsymbol{\theta}_0) \\
&\leq -\frac{(6\alpha(1-2\sigma_2)-5)\eta_t}{6\alpha}\sum_{t=0}^{T-1}\mathbb{E}\left[\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1\right] \\
&= -\frac{(6\alpha(1-2\sigma_2)-5)\eta_t}{6\alpha}\sum_{t=0}^{T-1}\left(\mathbb{E}\left[\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 \mid \mathcal{E}(T)\right]\mathbb{P}(\mathcal{E}(T)) + \mathbb{E}\left[\overline{\eta}_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 \mid \overline{\mathcal{E}(T)}\right]\mathbb{P}\left(\overline{\mathcal{E}(T)}\right)\right) \\
&\leq -\frac{(6\alpha(1-2\sigma_2)-5)\eta_t}{6\alpha}\sum_{t=0}^{T-1}\mathbb{E}\left[\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 \mid \mathcal{E}(T)\right]\mathbb{P}(\mathcal{E}(T)) \\
&\leq -\frac{(6\alpha(1-2\sigma_2)-5)\eta_t}{12\alpha}\sum_{t=0}^{T-1}\mathbb{E}\left[\eta_t\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1 \mid \mathcal{E}(T)\right] \\
&\leq -\frac{(6\alpha(1-2\sigma_2)-5)\eta_t\zeta_0}{12\alpha}\sum_{t=0}^{T-1}\mathbb{E}\left[\min(\frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1^2}{\alpha\Lambda_P P}, \frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1^{3/2}}{\sqrt{\alpha\rho_H P^{3/2}}}) \mid \mathcal{E}(T)\right] \\
&\qquad \text{(From } \eta_t \geq \zeta_0\min(\frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1}{\alpha\Lambda_P P}, \sqrt{\frac{\|\nabla L(\boldsymbol{\theta}_t^{\text{Sign}})\|_1}{\alpha\rho_H P^{3/2}}})) \\
&\leq -\frac{(6\alpha(1-2\sigma_2)-5)\eta_t\zeta_0}{12\alpha}\sum_{t=0}^{T-1}\min(\frac{P\varepsilon^2}{\alpha\Lambda_P}, P\varepsilon\sqrt{\frac{\varepsilon}{\alpha\rho_H P^{1/2}}}) \\
&= -\frac{(6\alpha(1-2\sigma_2)-5)TP\varepsilon^2\zeta_0}{12\alpha^2\Lambda_P} \quad \text{(From } \varepsilon < \frac{\alpha\Lambda_P^2}{\rho_H\sqrt{P}}).
\end{aligned}$$

Therefore, we have:

$$
\begin{aligned}
T &\leq \frac{12\alpha^2(L(\boldsymbol{\theta}_0) - \mathbb{E}\left[L(\boldsymbol{\theta}_T^{\text{Sign}})\right])}{(6\alpha(1-2\sigma_2)-5)P\varepsilon^2\zeta_0}\Lambda_P \\
&\leq \frac{12\alpha^2(L(\boldsymbol{\theta}_0) - L_*)}{(6\alpha(1-2\sigma_2)-5)P\varepsilon^2\zeta_0}\Lambda_P.
\end{aligned}
$$

This means that when we take $T > \frac{12\alpha^2(L(\boldsymbol{\theta}_0)-L_*)}{(6\alpha(1-2\sigma_2)-5)P\varepsilon^2\zeta_0}\Lambda_P$, we have $\mathbb{P}\left(\mathcal{E}(T)\right) < \frac{1}{2}$. Therefore, we have

$$
\mathcal{T}_\varepsilon(\{\boldsymbol{\theta}_t^{\text{Sign}}\}_{t=0}^{\infty}, L, \|\cdot\|_1) \leq \frac{12\alpha^2(L(\boldsymbol{\theta}_0) - L_*)}{(6\alpha(1-2\sigma_2)-5)P\varepsilon^2\zeta_0}\Lambda_P,
$$

for any $\alpha > \frac{5}{6(1-2\sigma_2)}$. Setting $\alpha = \frac{5}{3(1-2\sigma_2)}$ to minimize the right-hand side completes the proof. □