Does Low Rank Adaptation Lead to Lower Robustness against Training-Time Attacks?

Zi Liang [1] **Haibo Hu** [1] **Qingqing Ye** [1] **Yaxin Xiao** [1] **Ronghua Li** [1]

Abstract

Low rank adaptation (LoRA) has emerged as a prominent technique for fine-tuning large language models (LLMs) thanks to its superb efficiency gains over previous methods. While extensive studies have examined the performance and structural properties of LoRA, its behavior upon training-time attacks remain underexplored, posing significant security risks. In this paper, we theoretically investigate the security implications of LoRA's low-rank structure during fine-tuning, in the context of its robustness against data poisoning and backdoor attacks. We propose an analytical framework that models LoRA's training dynamics, employs the neural tangent kernel to simplify the analysis of the training process, and applies information theory to establish connections between LoRA's low rank structure and its vulnerability against training-time attacks. Our analysis indicates that **LoRA exhibits better robustness to backdoor attacks than full fine-tuning, while becomes more vulnerable to untargeted data poisoning** due to its over-simplified information geometry. Extensive experimental evaluations have corroborated our theoretical findings.

1. Introduction

With the rapid growth in the parameter size of large language models (LLMs), parameter-efficient fine-tuning (PEFT) (Xu et al., 2023; Han et al., 2024) has gained increasing attention in both research and industry communities. Among various PEFT strategies, low-rank adaptation (LoRA) (Hu et al., 2021) has emerged as the de facto standard for fine-tuning LLMs thanks to its computational efficiency and minimal performance degradation.

[1]The Hong Kong Polytechnic University, Hong Kong, China. Correspondence to: Haibo Hu <haibo.hu@polyu.edu.hk>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

To compare LoRA with full fine-tuning across various dimensions, recently many studies have emerged. For example, researchers have investigated LoRA's expressive capacity (Zeng & Lee, 2024), the smoothness (Jang et al., 2024) of its convergence, the asymmetry (Zhu et al., 2024) in its submatrices, the impact of initialization (Hayou et al., 2024), and so on (Wang et al., 2024a; Koubbi et al., 2024; Mao et al., 2024).

While these analyses have shed light on many properties of LoRA, one important aspect, i.e., its potential security risks, remains largely overlooked. Existing studies in this area either use LoRA as a tool to facilitate backdoor attacks (Yin et al., 2024; Liu et al., 2024), adversarial attacks (Ji et al., 2024), and model stealing attacks (Horwitz et al., 2024; Liang et al., 2025), or focus merely on the benefits (Xu et al., 2024b) of LoRA in differential privacy and federated learning. **None of these works directly investigate the security vulnerabilities inherent in LoRA itself**, which leaves behind potential hazards and vulnerabilities in LoRA-fine-tuned LLMs that are deployed across millions of devices (Gunter et al., 2024).

To fill this gap, in this paper, we attempt to answer the question **whether LoRA-based fine-tuning is more vulnerable than full fine-tuning (FF) under mainstream training-time attacks** (e.g., data poisoning (Fan et al., 2022; Ramirez et al., 2022; He et al., 2024)). We introduce the concept of *training-time robustness (TTR)* for characterizing a model's resistance to training-time attacks and propose an analytical framework to theoretically examine the security implications of LoRA's low-rank structure. The main challenges are two-folded. First, the TTR of a model significantly depends on the specific training tasks and the complex dynamics of the training process. Second, the effectiveness of attacks is heavily influenced by hyperparameters (e.g., learning rate) and attack strategies (e.g., poisoning rate or backdoor triggers), both of which increase the complexity of analysis.

To address these challenges, we introduce two novel simplifications when modeling the training dynamics of LoRA. First, we reformulate TTR by measuring the similarity of gradients before and after data poisoning, which enables a neural tangent kernel (Jacot et al., 2021) (NTK)-based

analysis to simplify the modeling of a training procedure. Second, we further introduce information theory (Amari, 2016; Nielsen, 2020) to connect the model's structural properties with its TTR, thereby decoupling the influences of different training datasets and attack methods. Our findings suggest that **LoRA's low-rank structure typically results in a smoother information geometry compared to FF, generally indicating better training-time robustness against backdoor attacks**. However, we also observe that this simplicity might lead to obvious **performance degradation under poisoning attacks or perturbations due to an oversimplified decision surface**. We further quantify the key factors within LoRA that influence its TTR, demonstrating that initialization variance and rank are crucial determinants. Additionally, our analysis uncovers previously unexplained characteristics of LoRA, including the asymmetry and initialization of its submatrices, as well as the effects of various hyperparameters, such as the learning rate.

We summarize our contributions as follows:

• We propose a novel theoretical framework to analyze the security of LoRA, revealing how its low-rank structure influences training-time robustness during fine-tuning. To our best knowledge, this is the first work to investigate LoRA's intrinsic security vulnerabilities.

• We identify key factors within LoRA that influence its security and explain to what extent LoRA can be *theoretically equivalent* to full fine-tuning from a security perspective. Based on this analysis, we offer practical guidance for improving LoRA's security.

• We provide a comprehensive evaluation of LoRA and FF under poisoning and backdoor attacks. Experimental results substantiate the correctness of these findings and explanations.

Following a top-down structure, this paper is organized as follows. Section 2.1 introduces the basic notations and provides an overview of neural network training and the formulation of LoRA. Section 2.2 defines the concept of training-time robustness and highlights the analytical difficulties it presents. Sections 2.3 and 2.4 present high-level perspectives on how NTK and information geometry contribute to addressing these issues. Section 3 offers a comprehensive analysis and discussion, followed by empirical validation in Section 4. Our source code is available at: `https://github.com/liangzid/LoRA-sSecurity`.

2. Preliminary

2.1. Notations

Training Procedure. Without loss of generality, we begin our analysis with an L-layer artificial neural network (ANN) $F_\Theta : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ which aims to map the input data $x \in \mathbb{R}^{n_0}$ into corresponding output representations $y \in \mathbb{R}^{n_L}$. Given a training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1,2,...,N_{tr}}$ with N_{tr} finite training samples, we define the input matrix as $X = [x_1, x_2, ..., x_{N_{tr}}] \in \mathbb{R}^{N_{tr} \times n_0}$ and the corresponding output matrix as $Y = [y_1, ..., y_{N_{tr}}] \in \mathbb{R}^{N_{tr} \times n_L}$. The objective of the neural network F_Θ is to learn the mapping from X to Y by minimizing the following empirical risk function:

$$\hat{\mathcal{L}}(\Theta; X, Y) = \sum_{i}^{N_{tr}} \mathcal{L}(F_\Theta(x_i), y_i), \tag{1}$$

where $\Theta \in \mathbb{R}^P$ represents the set of P learnable parameters, and $\mathcal{L}$ is the loss function.

Linear Layers. Each layer $F^{(l)} : \mathbb{R}^{n_l} \to \mathbb{R}^{n_{l+1}}$ in F_Θ with $l \in \{0, 1, ..., L-1\}$ is defined as a linear transformation:

$$\begin{aligned} y^{(l)}(x_i) &= W^{(l)} \cdot x_i^{(l)} + b^{(l)}, \\ y_a^{(l)} &= \sigma(y^{(l)}), \end{aligned} \tag{2}$$

where $y^{(l)} \in \mathbb{R}^{n_{l+1}}$ is the preactivation output, which maps the l-th layer's input $x^{(l)} \in \mathbb{R}^{n_l}$ through the learnable matrix $W^{(l)} \in \mathbb{R}^{N_{l+1} \times N_l}$. The activation function $\sigma(\cdot)$ produces the output $y_a^{(l)} \in \mathbb{R}^{n_{l+1}}$ at the l-th layer.

LoRA Adapter. LoRA (Hu et al., 2021) introduces a mechanism to reduce the number of trainable parameters by freezing the original matrix $W^{(l)}$ and learn a low-rank update $\Delta W^{(l)}$. This update is factorized as the product of two low-rank submatrices,

$$\Delta W^{(l)} = B^{(l)} A^{(l)}, \tag{3}$$

where $A^{(l)} \in \mathbb{R}^{r \times n_l}$ and $B^{(l)} \in \mathbb{R}^{n_{l+1} \times r}$ are learnable matrices, and $r \ll \min\{n_l, n_{l+1}\}$.

We define the intermediate state in LoRA as

$$y_I^{(l)}(x_i) = A^{(l)} \cdot x_i^{(l)}. \tag{4}$$

2.2. Definition of Training-Time Robustness

The robustness of a trained model refers to its sensitivity to perturbed inputs. For adversarial attacks, model robustness is evaluated by its resistance to adversarial or noisy *test* samples (Xu et al., 2019; Costa et al., 2024; Wang et al., 2024b). Following the same idea, ***training-time** robustness (TTR)* is the model's resistance to noisy, poisoned, or backdoor *training* samples (Fan et al., 2022), that is, *the sensitivity of a neural network's parameter updates to perturbed training samples*.

Formally, given an ANN F_Θ, its training-time robustness can be quantified by the difference in parameter updates $\Delta\tilde{\Theta} - \Delta\Theta$ when the original training set $\mathcal{D}$ is replaced with a noisy (or poisoned) dataset $\tilde{\mathcal{D}} = (\tilde{X}, \tilde{Y})$. Here, $\tilde{X}$ and $\tilde{Y}$ denote two possible perturbations applied to the input

data X and the learning target Y, and $\Delta\tilde{\Theta}$ denotes the corresponding parameter updates on $\tilde{\mathcal{D}}$. To measure TTR, we define the following metric $\mathcal{M}$ based on the norm of parameter differences,

$$\mathcal{M}(F_\Theta, \mathcal{D}, \tilde{\mathcal{D}}) = \mathbb{E}_{(\mathcal{D},\tilde{\mathcal{D}})}\mathbb{E}_t||\Delta\Theta(t) - \Delta\tilde{\Theta}(t)||, \quad (5)$$

where $||\cdot||$ denotes the norm, and $\Delta\Theta$ and $\Delta\tilde{\Theta}$ denote the parameter updates obtain from training with $\mathcal{D}$ and $\tilde{\mathcal{D}}$, respectively.

Unfortunately, it is impractical for us to employ Equation 5 to analyze the TTR of an ANN due to two primary challenges: *i)* the metric $\mathcal{M}$ in Equation 5 varies dynamically across different training steps t, which introduces significant complexity for theoretical modeling; and *ii)* the significance of each parameter differs substantially, implying that a uniform reduction of parameter updates based on a norm fails to capture their varying importance.

To address these two challenges, we first simplify Equation 5 in a more tractable form.

2.3. Simplifying LoRA's Training Procedure with NTK

We adopt the concept of neural tangent kernel (NTK) to simplify the analysis of TTR. NTK is a special form of kernel function, which is defined as the inner product of gradients:

$$K_{ntk}(x, x') = \nabla_\theta F(x;\theta)^T \nabla_\theta F(x';\theta). \quad (6)$$

Theorem 2.1 (Jacot et al. (2021))**.** *As the width of the neural network approaches infinity, the NTK exhibits the following two key properties:*
• *The NTK converges to a* ***deterministic*** *limiting kernel that depends only on three factors:* i) *the variance of the parameter initialization,* ii) *the neural network structure, and* iii) *the selection of activation functions;*
• *NTK keeps* ***constant*** *through out each training step t.*

Intuitively, K_{ntk} can be interpreted as an *unnormalized* angle (cosine similarity) between the gradient descent directions of two input samples. This perspective inspires us to implicitly measure how much the gradient updates change when a clean sample (x_c) is poisoned ($\tilde{x}_c$), i.e.,

$$\mathcal{M}' = \| \mathbb{E}_{(x_c,\tilde{x}_c)\sim(\mathcal{D},\tilde{\mathcal{D}})} K_{ntk}(x_c, \tilde{x}_c) \| . \quad (7)$$

Similar to the role of the inner product in quantifying the similarity between two vectors, $\mathcal{M}'$ effectively captures the degree of approximated similarity in gradient updates between the original sample and its perturbed counterparts. Specifically, under the same pair $(x_c, \tilde{x}_c)$, a large value of $K_{ntk}(x_c, \tilde{x}_c)$ indicates that the neural network experiences more severe perturbations in its parameter updates, which reflects lower training-time robustness.

Comparing $\mathcal{M}'$ (Equation 7) with $\mathcal{M}$ (Equation 5), we observe that **the measurement of TTR has been significantly simplified by introducing NTK**. First, the expectation *w.r.t* training step t can be removed based on NTK's second property (Theorem 2.1). Second, the analyzed variable, $\Delta\Theta \in \mathbb{R}^P$, are transformed into $K \in \mathbb{R}^{n_L}$, which are more structured and *homogeneous*, making the reduction of norm more meaningful.

With the simplified metric $\mathcal{M}'$, the theoretical analysis can now be formalized as the comparison of $\mathcal{M}'$ between full fine-tuning (FF) and LoRA, i.e., to determine whether the inequality $\mathcal{M}'_{\text{ff}} \leq \mathcal{M}'_{\text{lora}}$ holds.

2.4. Information Geometry: Bridging TTR with Training-Time Attacks (TTA)

While the complexities related to t and parameter importance are now simplified by the NTK, it remains challenging to model the poisoning set $\tilde{D}$ quantitatively. For instance, Equation 7 fails to distinguish between different poisoning strategies, such as label flipping or backdoor trigger injection. Besides, $K_{ntk}(x_c, \tilde{x}_c)$ only captures attack behaviors at the sample level, whereas most practical training-time attacks are drawn from a distribution of samples (Fan et al., 2022).

To address these limitations, we introduce *information geometry (IG)* (Amari, 2016; Nielsen, 2020) to quantitatively model the robustness of specific model structures against TTA. As demonstrated in previous studies (Zhao et al., 2019; Naddeo et al., 2022; Rahmati et al., 2020), there is a strong correlation between IG and robustness. So IG can measure the curvature of the parameter space, offering insights into how an ANN adapts to unclean data.

First, we bridge NTK with *Fisher information* (Fisher, 1922), one of the core concept of IG, by

Theorem 2.2. *When the width of F_Θ approaches infinity, its Fisher information $\mathcal{I}_\Theta$ under $\tilde{\mathcal{D}}$ is equal to its weighted $\mathcal{M}'(\tilde{\mathcal{D}}, \tilde{\mathcal{D}})$, i.e.,*

$$\begin{aligned} I_\theta &= \mathbb{E}_{x\sim\tilde{\mathcal{D}}}\left[\nabla_\theta \mathcal{L}(x,\theta)^T \nabla_\theta \mathcal{L}(x,\theta)\right] \\ &= \mathbb{E}_{\tilde{x}_c\in\tilde{\mathcal{D}}}\left[\nabla_{F_\theta}\mathcal{L}(x,\theta)^T K_{ntk}(x,x) \nabla_{F_\theta}\mathcal{L}(x,\theta)\right]. \end{aligned} \quad (8)$$

Proofs are in Appendix A.1.

Let $\lambda_1, \lambda_2, ..., \lambda_{n_L}$ denote the n_L eigenvalues of the Fisher information matrix $\mathcal{I}_\Theta$. Then we can quantify the *information bits* (IB) of the model as

$$\mathbf{IB} = \frac{1}{2}\log \det_{\text{pseudo}} \mathcal{I}_\Theta = \frac{1}{2}\sum_{\lambda_i>0}^{n_L} \lambda_i. \quad (9)$$

Third, we can measure the curvature of the fine-tuning man-

ifold with *Rényi entropy* (Rényi, 1961)

$$H_\alpha = \frac{1}{1-\alpha}\log\left(\sum_{i=1}^{n_L}\lambda_i^\alpha\right), \tag{10}$$

where $\alpha \geq 0$ controls the norm formation on the $\mathcal{I}_\Theta$. Specifically, H_1 corresponds to the Shannon entropy[1], while $H_\infty = \max\{\lambda_1, \lambda_2, ..., \lambda_{n_L}\}$.

Intuitively, a higher **IB** and H_α indicates a more complex fine-tuning manifold of the model, which implicitly demonstrates a higher function fitting ability.

Based on Equation 7, 9, and 10, we now proceed to analyze the potential security vulnerabilities introduced by LoRA's fine-tuning process.

3. Does LoRA Lead to LoRA (Lower Training Time Robustness against Attacks)?

3.1. LoRA's NTK

Modeling the Feedforward Procedure. As shown in the previous research (Lee et al., 2018), the output function $F_\Theta : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ converges to an independent, identity-centered *Gaussian process (GP)* under the infinite-width limit, i.e., as $n_l \to \infty$ for $l = 1, 2, \ldots, n_{L-1}$.

Under this GP formulation, the covariance between outputs at layer l can be expressed as:

$$\begin{aligned}\Sigma^1(X, X') &= X^T X',\\ \Sigma^l(X, X') &= \mathbb{E}_{f\sim\mathcal{N}(0,\Sigma^{l-1})}[\sigma(f(X))^T\sigma(f(X'))]\\ &= \sum_{j=1}^{n_{l-1}}\sigma(y_j^{(l-1)}(X))^T\sigma(y_j^{(l-1)}(X')),\end{aligned} \tag{11}$$

where $y_j^{(l-1)}(x)$ denotes the j-th element of the preactivation vector $y^{(l-1)}$ at input x.

Modeling the Learning Procedure with NTK. Based on Equation 6 in Section 2.3, we now derive the NTK for an ANN F under both FF and LoRA-based fine-tuning. Specifically, the NTK of FF can be represented by

$$\begin{aligned}K_{\text{ff}}^{(1,k)}(x, x') &= I_{n_l}\otimes\Sigma^{(1)}(x, x') = x^T\cdot x',\\ K_{\text{ff}}^{(l,k)}(x, x') &= K_{\text{ff}}^{(l-1,k)}(x, x')\dot{\Sigma}^{(l)}(x, x') + \Sigma^{(l)}(x, x'),\end{aligned} \tag{12}$$

where $k = \{0, 1, ..., n_l - 1\}$, $\otimes$ denotes the Kronecker product, and

$$\begin{aligned}\dot{\Sigma}^{(l)}(x, x') &= \mathbb{E}_{f\sim\mathcal{N}(0,\Sigma^{(l-1)})}[\dot{\sigma}(f(x))\dot{\sigma}(f(x'))]\\ &= \dot{\sigma}(y^{(l-1)}(x))^T\dot{\sigma}(y^{(l-1)}(x')).\end{aligned} \tag{13}$$

$\dot{\sigma}(y^{(l-1)}) = \frac{\partial\sigma(y^{(l-1)})}{\partial y}|_{y=y^{(l-1)}}$ denotes the partial derivative of the activation function σ.

[1]The proof is presented in Appendix C.1.

As for LoRA, we can also derive its NTK functions as

Lemma 3.1 (NTK of LoRA)**.** *The neural tangent kernel of an l-layer ANN trained with LoRA can be expressed as follows.*

$$\begin{aligned}K_{\textit{LoRA}}^{(1,k)}(x, x') &= K_{\textit{ff}}^{(1,k)}\\ K_{\textit{LoRA}}^{(l,k)}(x, x') &= K_{\textit{LoRA}}^{(l-1,k)}(x, x')\dot{\Sigma}^{(l)} + \Sigma_{\textit{LoRA}}^{(l)}(x, x'),\end{aligned} \tag{14}$$

where

$$\Sigma_{\textit{LoRA}}^{(l)}(x, x') = \sigma(y^{(l-1)}(x))^T A^{(l)\,T}A^{(l)}\sigma(y^{(l-1)}(x')),$$

and $W_{\textit{LoRA}}^{(l)} = W_0^{(l)} + B^{(l)}A^{(l)}$ denotes the l-th layer's weight matrix of LoRA.

The detailed derivation of NTK functions for FF and LoRA are in Appendix A.2.

Based on two NTK functions K_{LoRA} and K_{ff}, along with the proposed metric $\mathcal{M}'$, we proceed to compare the kernel functions between FF and LoRA.

3.2. The NTK Relationship between FF and LoRA

We begin our analysis by comparing the NTK of a single layer between LoRA and FF.

Assumption 3.2 (Only One Layer is Different (OOLD))**.** Given an L-layer neural network F_Θ, OOLD assumes that during training, the first $l-1$ layers remain identical for both FF and LoRA. They only diverge at the l-th layer, which employs FF and LoRA, respectively. We denote their NTK functions as $K_{\text{ff}}(x, x')$ and $K_{\text{LoRA}}(x, x')$.

Under the OOLD assumption, we have

Theorem 3.3 (NTK Relationship between FF and LoRA)**.** *For an l-layer ANN with infinite width, the NTK functions of FF and LoRA at the l-th layer are related by the following expression:*

$$K_{\textit{LoRA}}^{(l,k)} = K_{\textit{ff}}^{(l,k)} + \Delta_r^{(l)}, \tag{15}$$

where

$$\begin{aligned}\Delta_r^{(l)} = [\sigma(y^{(l-1)}(x))]^T(&A^{(l)\,T}A^{(l)} -\\ &I_{n_{l-1}\times n_{l-1}})[\sigma(y^{(l-1)}(x'))].\end{aligned}$$

Let $M_\Delta^{(l)}$ denote the kernel matrix of $\Delta_r^{(l)}$, i.e., $M_\Delta^{(l)} = A^{(l)\,T}A^{(l)} - I_{n_{l-1}\times n_{l-1}}$, then the following theorem holds:

Theorem 3.4 ($M_\Delta^{(l)}$'s Negative Semi-Definiteness)**.** *When the LoRA submatrix $A^{(l)} \in \mathbb{R}^{r\times n_{l-1}}$ is initialized with variance σ_a^2, $\sigma_a^2 < 1/n_{l-1}$, and $r \leq n_{l-1}$, then $M_\Delta^{(l)}$ is a **negative semi-definite** matrix, with r eigenvalues equal to $\sigma_a^2\cdot n_{l-1}$ and $n_l - r$ eigenvalues equal to 0.*

Theorem 3.4 establishes a foundation for comparing FF and LoRA's training-time robustness from an information geometry perspective, which will be detailed in Section 3.3. Based on Theorem 3.4, we reach the following corollary.

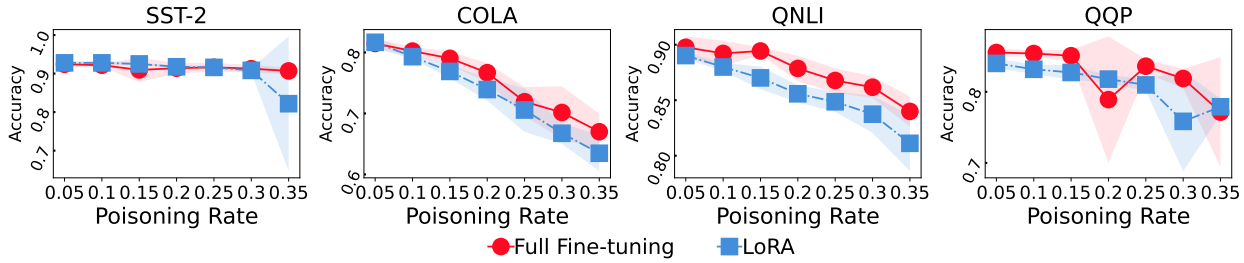

Figure 1. Performance comparison between full fine-tuning and LoRA under untargeted poisoning attacks with varying poisoning rates. The curves show accuracy, and the shaded areas represent the standard deviation across multiple runs. More experiments are in Figure 6.

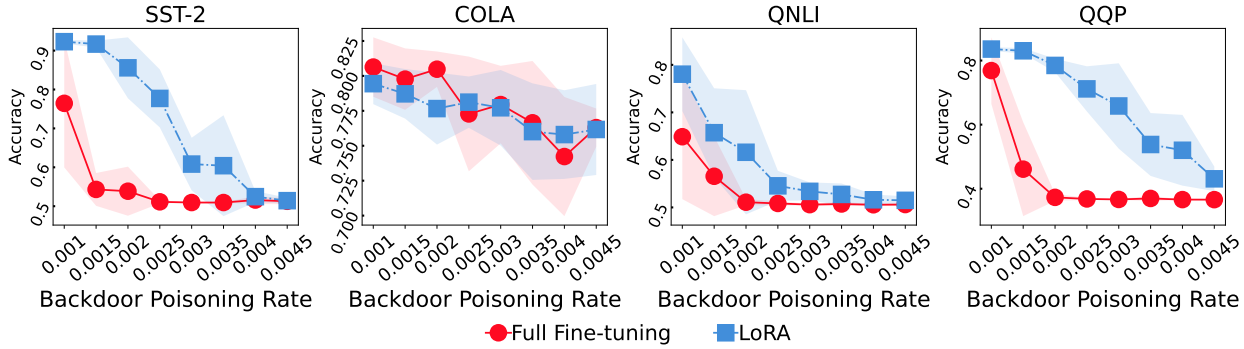

Figure 2. Performance comparison between full fine-tuning and LoRA under backdoor attacks with varying poisoning rates. Figure 7 exhibits the results on the four metrics.

Corollary 3.5 (Ideal Full Rank Adaptation)**.** *When $n_{l-1} \to \infty$, the kernel matrix $M_\Delta^{(l)}$ strictly converges to* $\mathbf{0}$*, i.e., $K_{LoRA}^{(l)}(x,x) \equiv K_{ff}^{(l)}(x,x)$ if $r = n_{l-1}$ and the initialization variance satisfies $\sigma_a^2 = 1/n_{l-1}$.*

The proofs of Theorem 3.3, 3.4, and Corollary 3.5 are presented in Appendix A.3 and Appendix A.4.

Corollary 3.5 offers a key insight into the relationship between the LoRA and FF. Specifically, it shows that when LoRA achieves a full rank and the weight matrices are initialized with a specific variance, the expected learning effectiveness of LoRA matches that of FF. In other words, under these conditions, both methods exhibit **equivalent** expressiveness in terms of their NTK functions. Moreover, Theorem 3.4 reveals that both the rank and the initialization variance significantly influence the properties of $M_\Delta^{(l)}$, which raises several critical questions: *i)* does LoRA exhibit higher or lower TTR than FF? *ii)* how do the rank and initialization variance affect its TTR? *iii)* under what conditions does a full-rank LoRA offer equivalent TTR to FF against training-time attacks?

3.3. Theoretical Analysis

Key Results: LoRA Exhibits Fewer Information Bits and Smoother Information Geometry than FF, Leading to Higher Training-Time Robustness.

To answer these questions, we begin our theoretical analysis by computing the $\mathbf{IB}$ and the H_α for both LoRA and FF.

Theorem 3.6 ($\mathbf{IB}_{\text{ff}} \geq \mathbf{IB}_{\text{LoRA}}$ & $H_{\alpha\text{ff}} \geq H_{\alpha\text{LoRA}}$)**.** *The information bits and the Rényi entropy of LoRA are always* ***smaller*** *than those of FF if $M_\Delta^{(l)}$ is a negative semi-definite matrix, i.e., $r \leq n_{l-1}$ and $\sigma^2 \leq 1/n_{l-1}$.*

The proof is in Appendix A.5.

The conditions stated in Theorem 3.6 are typically satisfied in practice. First, the rank is typically chosen to be significantly "smaller" than the original dimension n_{l-1} to reduce computation costs. Second, the initialization variance of LoRA's matrix is generally set to a value smaller than $1/{n_{l-1}}^2$. As a result, in most practical scenarios, LoRA is expected to exhibit lower information bits (low $\mathbf{IB}$) and smoother information surface (H_α) than FF.

Note that Theorem 3.6 appears to contradict with some existing research (Zeng & Lee, 2024) that suggests when r exceeds a certain threshold, the expressivity of LoRA becomes equivalent to that of FF. Such contradiction can be justified because our theorem focuses on the IG during training process, i.e., on "how can the model's parameters possibly evolve throughout training" as opposed to "the

[2]Specifically, it is set to $1/(3 \cdot n_{l-1})$ in both the official implementation and the standard libraries (e.g. peft (Mangrulkar et al., 2022)).

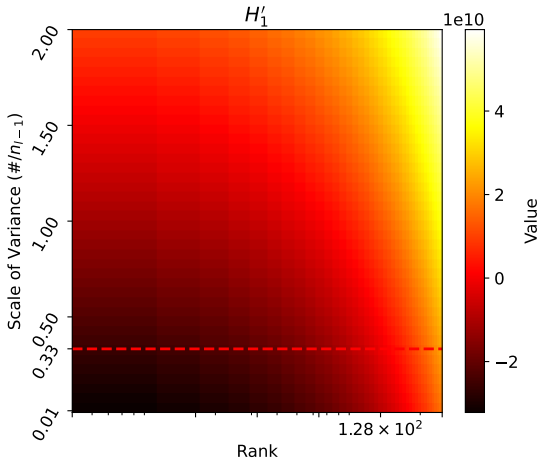

Figure 3. Visualization of the Shannon entropy H_1' under different ranks and variance scales. Brighter color points indicate higher entropy values. The red dashed line represents the default variance scaling setting used in the implementation of LoRA.

expressiveness of the final trained models".

Incorporating the definitions of $\mathcal{M}'$ and $\mathcal{I}_\Theta$ to Theorem 3.6, we can conclude that $\tilde{\mathcal{D}}$ brings more significant parameter updates in FF than in LoRA, which means that LoRA does exhibits **higher** training-time robustness than FF under the conditions of Theorem 3.6. This discovery also coincides with some previous studies, such as the weak-to-strong alignments (Burns et al., 2023).

Unfortunately, this increased TTR is at the cost of reduced information bits, which prompts a critical question — what is the tax for LoRA's enhanced TTR?

Double-Edged Sword of LoRA's TTR. While low rank adaptation offers the advantage of higher training-time robustness, this robustness does not always translate into resistance against all types of training-time attacks. On one hand, a reduced H_α indicates that LoRA's IG is potentially *smoother* than that of full fine-tuning, which suggests a smaller search space for backdoor triggers, thereby providing stronger resistance to backdoor attacks. On the other hand, the oversimplification of the manifold may make LoRA more susceptible to noisy or intentionally poisoned data, causing higher vulnerability to data poisoning attacks.

Below, we examine this phenomenon from an orthogonality perspective.

Consider two training samples: a clean input x_c and its backdoored input x_t. The optimization target under these two samples can be represented as minimizing the following formula:

$$|\nabla_\theta \mathcal{L}(x_c, \theta)^T \cdot \nabla_\theta \mathcal{L}(x_t, \theta)|, \tag{16}$$

i.e., the adversary aims to ensure that the optimization process driven by $\nabla_\theta \mathcal{L}(x_c, \theta)$ and $\nabla_\theta \mathcal{L}(x_t, \theta)$ occur simultaneously and both significantly influence the training, which aligns with the target of BPA to maintain performance on most inputs while producing significantly altered predictions only when a specific trigger is present. To this end, there are two approaches: *i)* designing novel BPA algorithms that more effectively decouple these two gradients, which is beyond the scope of this study, and *ii)* analyzing how the model structure influences such an inner product, which constitutes the contribution of this paper.

By analyzing the properties of such an inner product, our indicators provide key insights showing that LoRA provides "a smaller search space for the existence of backdoor triggers" due to *i)* its $(n_{l-1} - r)$ zero eigenvalues and *ii)* smaller variances in the remaining r dimension's parameter updates (i.e., smaller angles between gradients), both of which intuitively manifest as smoother information geometry.

Similarly, we can provide a complementary explanation of why a model with smoother IG tends to be more sensitive to perturbations. Given a clean training input x_c, and its perturbed version x_u, where x_u is assigned a different label for the purpose of untargeted poisoning. The target of UPA is to maximize:

$$|\nabla_\theta \mathcal{L}(x_c, \theta)^T \cdot \nabla_\theta \mathcal{L}(x_u, \theta)|, \tag{17}$$

i.e., as adversaries, we aim to align the optimization direction of the poisoned sample x_u as closely as possible with that of the clean training objective, because we aim to maximally influence the model's predictions while injecting only a small fraction of poisoned data. This objective directly contrasts with the BPA case, as we instead aim to decouple the optimization directions. Consequently, we draw the opposite conclusion for UPA. [3]

Based on this analysis, it is crucial to carefully tune the rank r and the initialization variance σ^2 to balance its vulnerabilities among different training-time attacks.

Quantifying the Impact of r and σ^2. Though the exact values of $\mathbf{IB}$ and H_α remain dependent on $\tilde{\mathcal{D}}$, we can still gain some insights by analyzing the eigenstructure of K_{LoRA}'s kernel matrix. Specifically, we approximate H_α, which leverages the K_{LoRA}'s kernel matrix's eigenvalues λ', defined as

$$H_\alpha' = \frac{1}{1-\alpha} \log \left(\sum_{\lambda' \in \text{Eigen}(A^{(l)T} A)} (\lambda')^\alpha \right). \tag{18}$$

We visualize the manifold of H_α' under different ranks and initialization scales in Figure 3.

[3] Note that what we emphasize is that "the **over**simplification of the manifold may make LoRA more susceptible", i.e., the empirical phenomenon that LoRA is more vulnerable when facing UPA (or noise) may not be obvious if the model is severely overparameterized compared to the task.

3.4. Further Analysis

In this section, we extend our analysis to more general adaptation settings and a broader range of model architectures.

Adaptations beyond OOLD Assumption. Beyond the scope of Assumption 3.2, our conclusions can be generalized where LoRA is applied to adapt all linear modules within a neural network. The detailed proofs for generalized versions of Theorem 3.4, Corollary 3.5, and Theorem 3.6 are in Appendix A.6.

Extensions to More Complex Model Architectures. We further study the impact of LoRA on more complex and practical architectures, such as the Transformer (Vaswani, 2017). A detailed discussion is in Appendix B.

Broader Implications of Our Analysis. Though our primary focus is the TTR of LoRA, our analytical framework can also shed light on several unexplained properties and settings of LoRA. These include the asymmetry in adaptation, the choice of initialization strategy, the scaling factor of α, and the effect of freezing matrix A during fine-tuning. An in-depth analysis of these phenomena is in Appendix C.

4. Experiments

In this section, we empirically evaluate the TTR of both LoRA and FF under commonly used language models.

4.1. Settings

Experimental Details. Following prior works (Hu et al., 2021; Zhu et al., 2024; Mao et al., 2024) on LoRA, we conduct fine-tuning of natural language understanding models on the GLUE benchmark (Wang et al., 2018) as our primary evaluation environment. Specifically, we utilize BERT-large (Devlin et al., 2019) as the backbone model and evaluate their performance on six binary classification tasks, including SST-2 (Socher et al., 2013), COLA (Warstadt et al., 2018), QNLI (Wang et al., 2018), QQP (Sharma et al., 2019), RTE (Poliak, 2020), and MRPC (Dolan & Brockett, 2005). The evaluation metrics include Precision (Pre.), Recall (Rec.), Accuracy (Acc.), and F1 Score (F1).

Implementation Details. The maximum sequence length is set to 512, and the batch size is fixed at 8. For learning rates, we apply 3×10^{-5} for LoRA's low rank fine-tuning and 3×10^{-6} for both LoRA's high rank fine-tuning and FF. Each fine-tuning procedure is conducted for a maximum of 10,000 steps. These hyperparameters are carefully tuned to ensure that both LoRA and FF achieve stable and competitive results across the evaluated tasks.

For LoRA-specific settings, we use a rank of 8 and set the scaling parameter α to 16 as default values. All experiments are conducted on eight 24 GB Nvidia RTX 4090 GPUs.

To ensure robustness, we repeat each training experiment five times under fixed random seeds and report the mean values along with their standard deviations.

4.2. Settings of Training-time Attacks

We consider two types of mainstream training-time attacks on language models, namely the *untargeted poisoning attacks*, and the *backdoor-based poisoning attacks*.

Untargeted Poisoning Attacks (UPA). We consider a simple and yet common UPA strategy (Fan et al., 2022): randomly flipping the labels of training samples based on a fixed poisoning rate (**PR**) ρ. Consequently, we can measure the relative performance degradation of LoRA and FF under the same poisoning rates, which provides empirical insights into their resistance against UPA.

Backdoor-based Poisoning Attacks (BPA). We implement a widely used backdoor poisoning attack by introducing a trigger with modified labels (Wan et al., 2023). Specifically, we randomly select a subset of training samples with $N_{tr} \times \rho$ examples, where ρ denotes the poisoning rate. For each selected sample, we append the trigger pattern `[.*?]` to the original text and modify its classification label to `1`. To assess the effectiveness, we add the same trigger into test samples and evaluate whether the model's predictions are consistently altered to the target label (i.e., `1`), to compare the robustness of LoRA in resisting backdoor attacks.

4.3. LoRA: Excelling in Backdoor Defense While Falling Short Against Untargeted Poisoning

We compare the performance of LoRA and FF under UPA and BPA across different poisoning rates. The results are presented in Figure 1 for untargeted poisoning and Figure 2 for backdoor attacks.

From Figure 1, we observe a noticeable performance gap between FF and LoRA-based fine-tuning under UPA. This gap is relatively minor in certain datasets, such as SST-2, but is more pronounced in others, including QNLI and QQP. As the poisoning rate increases, the accuracy gap is widened, indicating more severe performance degradation for LoRA-based fine-tuning compared to full fine-tuning.

In contrast to its poor performance under UPA, Figure 2 shows that LoRA significantly outperforms FF in resisting backdoor attacks, demonstrating stronger robustness. Apart from comparable results on COLA, LoRA achieves up to 30% improvement over FF on datasets such as SST-2 and QQP, indicating substantial gains in backdoor defense.

We also observe that in backdoor experiments, both LoRA and FF exhibit consistent performance on *untriggered* test data, as shown in Figure 8. This phenomenon indicates

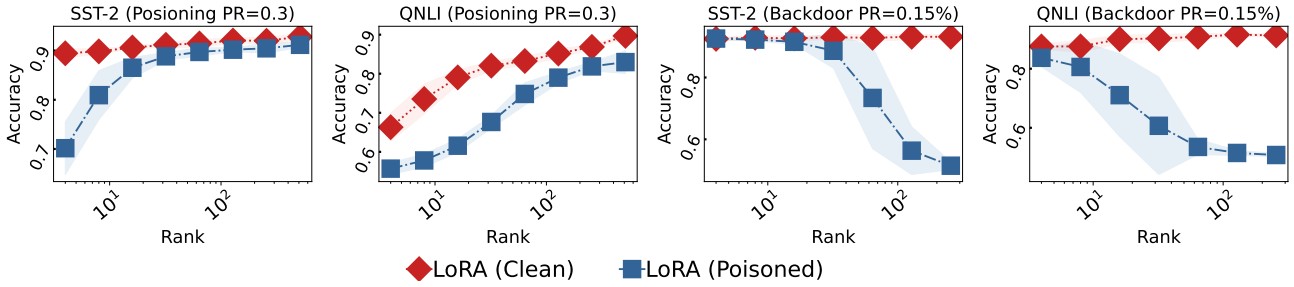

Figure 4. The effect of **rank** on LoRA's robustness under untargeted poisoning and backdoor poisoning attacks. More experiments are in Figure 9 and Figure 10.

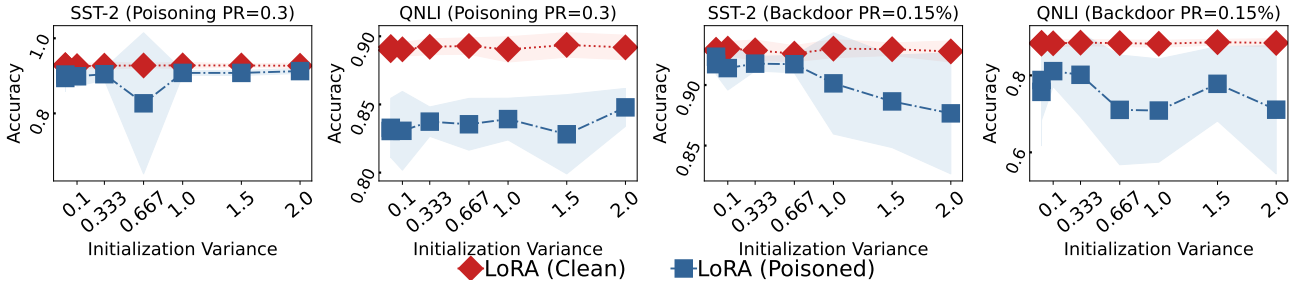

Figure 5. The effect of **initialization variance** on LoRA's robustness against untargeted poisoning and backdoor attacks. Experiments on more datasets are shown in Figure 11 and Figure 12.

that the introduction of backdoors for both methods do not degrade the models' general performance on normal inputs.

4.4. Key Factors Influencing LoRA's Security

In this section, we examine those critical factors that influence the TTR of LoRA, as in our theoretical analysis.

4.4.1. Rank of LoRA

Settings. In Figure 4, we evaluate the performance of LoRA on both the clean and the poisoned training sets across ranks ranging from 4 to 512. Additional empirical results on more datasets and metrics are provided in Appendix E.

A High Rank of LoRA is Robust against Poisoning. The first two subfigures in Figure 4 illustrate the influences of LoRA's rank against untargeted poisoning attacks. The performance of LoRA fine-tuning under all ranks is good and stable on clean datasets, suggesting that a high rank does not affect the performance of fine-tuning. Conversely, when fine-tuned on a poisoned dataset, the performance decreases significantly when the rank is lower than a threshold (e.g. 16 in SST-2), exhibiting an increasing gap compared to the results on the clean dataset. This phenomenon indicates that a low rank of LoRA will decrease the training-time robustness of models.

A High Rank of LoRA is Weak against Backdoor Attacks. The last two subfigures in Figure 4 show the backdoor resistance of LoRA under different ranks. With the increase of rank, the performance of LoRA on clean set remains stable, while its performance on backdoor poisoned dataset decreases, which suggests that a high rank will reduce the backdoor resistance of LoRA.

Combining the above, there exists a robustness trade-off between UPA and BPA with respect to LoRA's rank, which coincides with our theoretical analysis.

4.4.2. Variance on LoRA's Initialization

In addition to the rank, our theoretical analysis in Section 3.3 suggests that the initialization variance of LoRA's A matrix plays a critical role in the model's training-time robustness, which we examine empirically hereby.

Settings. As shown in Section 3.3, the mainstream implementation (Hu et al., 2021; Mangrulkar et al., 2022) adopts a Kaiming uniform initialization (He et al., 2015), where the default variance is set to $k \cdot 1/n_l$, with $k = 1/3$. Following this setting, we vary the scale hyperparameter k from 0.001 to 2.0 and evaluate its effect under both poisoning and backdoor attack scenarios.

Variance does not influence performance. As shown in Figure 5, when trained across different scales of initialization variance, LoRA's performance on the **clean** set remains stable, indicating that the model can effectively adapt to the training task regardless of the chosen variance.

Variance slightly influences the poisoning. In contrast to rank, the first two subfigures in Figure 5 indicate that the impact of initialization variance on robustness against poisoning attacks is minimal. This phenomenon deviates from our theoretical analysis, which suggests that a smaller initialization variance could lead to lower UPA resistance and higher BPA robustness. A possible explanation comes from the limitation of NTK, that is, since the real weights of LoRA change during fine-tuning, the kernel function K_{LoRA}'s nonzero eigenvalues are not be strictly deterministic by the initialization. As a result, the influence of variance is less pronounced than that of rank.

Variance does influence backdoor performance. Different from the results from the poisoning experiments, the last two subfigures in Figure 5 shows a strong correlation between initialization variance and backdoor resistance. Specifically, a smaller initialization variance leads to relatively higher performance under backdoor attacks and lower standard deviation of results, which also aligns with our theoretical analysis.

We also provide supplemental experiments to further support our theoretical analysis, including:

- **Additional Attacks.** We implement four additional training-time attacks to reinforce our conclusions, as presented in Appendix E.1.
- **Alternative Initialization Strategies.** We evaluate two additional commonly used initialization strategies to demonstrate the robustness of our conclusions across different settings, as detailed in Appendix E.2.
- **Experiments on Generative Language Models.** We further conduct experiments (Appendix E.3) on generative large language models to demonstrate that our method generalizes to broader scenarios.

4.5. Summary of Findings and Defenses

Based on the above analysis, we summarize our key findings to mitigate these risks associated with LoRA:

- LoRA is more vulnerable than full fine-tuning to untargeted poisoning attacks but demonstrates greater robustness against backdoor attacks.
- In addition to the trade-off between performance and computational cost, LoRA's rank also influences the trade-off between untargeted poisoning and backdoor attacks.
- Besides of the rank, the initialization variance of the A matrix in LoRA significantly impacts training-time robustness.
- To improve robustness against backdoor attacks, the rank should be set as low as possible, provided that performance requirements are met.
- A small scale of initialization variance is recommended to enhance training-time robustness.

5. Conclusion

This paper explores the potential training-time security risks of LoRA-based fine-tuning. Based on the definition of training-time robustness, this paper constructs and compares the neural tangent kernels and the information geometry of LoRA and full fine-tuning, revealing that two factors, rank and initialization variance, significantly impact its security during training. Theoretical analysis demonstrates that LoRA is more vulnerable to untargeted poisoning but more robust against backdoor attacks. Extensive experiments validate the theoretical analysis and key findings.

Acknowledgment

The authors would like to thank the reviewers for their detailed suggestions. This work was supported by the National Natural Science Foundation of China (Grant No: 92270123 and 62372122), and the Research Grants Council, Hong Kong SAR, China (Grant No: 15225921, 15209922, 15210023, 15224124).

Impact Statement

As the inaugural investigation into the security vulnerabilities of LoRA, this study underscores critical concerns regarding the security implications of LoRA, thereby broadening the discourse on its safe and effective utilization. This research will catalyze further scholarly inquiry into the security dimensions of LoRA across other domains, including but not limited to unlearning, adversarial attacks, and membership inference, and will stimulate advancements in enhancing its robustness. Furthermore, the analytical framework developed herein, along with the comprehensive elucidation of LoRA's intrinsic properties and its TTR, is anticipated to exert a significant influence on subsequent research endeavors focused on the structural analysis of machine learning models. This contribution is expected to pave the way for more rigorous and nuanced examinations of model architectures in the future.

References

Aghajanyan, A., Gupta, S., and Zettlemoyer, L. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the*

11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pp. 7319–7328. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.568. URL https://doi.org/10.18653/v1/2021.acl-long.568.

Amari, S.-i. *Information geometry and its applications*, volume 194. Springer, 2016.

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R., and Wang, R. On exact computation with an infinitely wide neural net, 2019. URL https://arxiv.org/abs/1904.11955.

Burns, C., Izmailov, P., Kirchner, J. H., Baker, B., Gao, L., Aschenbrenner, L., Chen, Y., Ecoffet, A., Joglekar, M., Leike, J., Sutskever, I., and Wu, J. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision, 2023. URL https://arxiv.org/abs/2312.09390.

Costa, J. C., Roxo, T., Proença, H., and Inácio, P. R. M. How deep learning sees the world: A survey on adversarial attacks and defenses. *IEEE Access*, 12:61113–61136, 2024. ISSN 2169-3536. doi: 10.1109/access.2024.3395118. URL http://dx.doi.org/10.1109/ACCESS.2024.3395118.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Dolan, W. B. and Brockett, C. Automatically constructing a corpus of sentential paraphrases. In *IWP 2005*, 2005. URL https://aclanthology.org/I05-5002.

Fan, J., Yan, Q., Li, M., Qu, G., and Xiao, Y. A survey on data poisoning attacks and defenses. In *2022 7th IEEE International Conference on Data Science in Cyberspace (DSC)*, pp. 48–55, 2022. doi: 10.1109/DSC55868.2022.00014.

Fisher, R. A. On the mathematical foundations of theoretical statistics. *Philosophical transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, 222(594-604):309–368, 1922.

Gu, T., Dolan-Gavitt, B., and Garg, S. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *CoRR*, abs/1708.06733, 2017. URL http://arxiv.org/abs/1708.06733.

Gunter, T., Wang, Z., Wang, C., Pang, R., Narayanan, A., Zhang, A., Zhang, B., Chen, C., Chiu, C., Qiu, D., Gopinath, D., Yap, D. A., Yin, D., Nan, F., Weers, F., Yin, G., Huang, H., Wang, J., Lu, J., Peebles, J., Ye, K., Lee, M., Du, N., Chen, Q., Keunebroek, Q., Wiseman, S., Evans, S., Lei, T., Rathod, V., Kong, X., Du, X., Li, Y., Wang, Y., Gao, Y., Ahmed, Z., Xu, Z., Lu, Z., Rashid, A., Jose, A. M., Doane, A., Bencomo, A., Vanderby, A., Hansen, A., Jain, A., Anupama, A. M., Kamal, A., Wu, B., Brum, C., Maalouf, C., Erdenebileg, C., Dulhanty, C., Moritz, D., Kang, D., Jimenez, E., Ladd, E., Shi, F., Bai, F., Chu, F., Hohman, F., Kotek, H., Coleman, H. G., Li, J., Bigham, J. P., Cao, J., Lai, J., Cheung, J., Shan, J., Zhou, J., Li, J., Qin, J., Singh, K., Vega, K., Zou, K., Heckman, L., Gardiner, L., Bowler, M., Cordell, M., Cao, M., Hay, N., Shahdadpuri, N., Godwin, O., Dighe, P., Rachapudi, P., Tantawi, R., Frigg, R., Davarnia, S., Shah, S., Guha, S., Sirovica, S., Ma, S., Ma, S., Wang, S., Kim, S., Jayaram, S., Shankar, V., Paidi, V., Kumar, V., Wang, X., Zheng, X., and Cheng, W. Apple intelligence foundation language models. *CoRR*, abs/2407.21075, 2024. doi: 10.48550/ARXIV.2407.21075. URL https://doi.org/10.48550/arXiv.2407.21075.

Han, Z., Gao, C., Liu, J., Zhang, J., and Zhang, S. Q. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

Hayou, S., Ghosh, N., and Yu, B. The impact of initialization on lora finetuning dynamics, 2024. URL https://arxiv.org/abs/2406.08447.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, 2015. URL https://arxiv.org/abs/1502.01852.

He, X., Huang, K., Ye, Q., and Hu, H. Data poisoning attacks to local differential privacy protocols for graphs, 2024. URL https://arxiv.org/abs/2412.19837.

Horwitz, E., Kahana, J., and Hoshen, Y. Recovering the pre-fine-tuning weights of generative models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=761UxjOTHB.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Hubinger, E., Denison, C., Mu, J., Lambert, M., Tong, M., MacDiarmid, M., Lanham, T., Ziegler, D. M., Maxwell, T., Cheng, N., Jermyn, A. S., Askell, A., Radhakrishnan, A., Anil, C., Duvenaud, D., Ganguli, D., Barez, F., Clark, J., Ndousse, K., Sachan, K., Sellitto, M., Sharma, M., DasSarma, N., Grosse, R., Kravec, S., Bai, Y., Witten, Z., Favaro, M., Brauner, J., Karnofsky, H., Christiano, P. F., Bowman, S. R., Graham, L., Kaplan, J., Mindermann, S., Greenblatt, R., Shlegeris, B., Schiefer, N., and Perez, E. Sleeper agents: Training deceptive llms that persist through safety training. *CoRR*, abs/2401.05566, 2024. doi: 10.48550/ARXIV.2401.05566. URL https://doi.org/10.48550/arXiv.2401.05566.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: convergence and generalization in neural networks (invited paper). In Khuller, S. and Williams, V. V. (eds.), *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pp. 6. ACM, 2021. doi: 10.1145/3406325.3465355. URL https://doi.org/10.1145/3406325.3465355.

Jang, U., Lee, J. D., and Ryu, E. K. Lora training in the NTK regime has no spurious local minima. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=s1sdx6vNsU.

Ji, Y., Liu, Y., Zhang, Z., Zhang, Z., Zhao, Y., Zhou, G., Zhang, X., Liu, X., and Zheng, X. Advlora: Adversarial low-rank adaptation of vision-language models. *CoRR*, abs/2404.13425, 2024. doi: 10.48550/ARXIV.2404.13425. URL https://doi.org/10.48550/arXiv.2404.13425.

Koubbi, H., Boussard, M., and Hernandez, L. The impact of lora on the emergence of clusters in transformers. *CoRR*, abs/2402.15415, 2024. doi: 10.48550/ARXIV.2402.15415. URL https://doi.org/10.48550/arXiv.2402.15415.

Kumar, S. K. On weight initialization in deep neural networks. *CoRR*, abs/1704.08863, 2017. URL http://arxiv.org/abs/1704.08863.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. Deep neural networks as gaussian processes. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=B1EA-M-0Z.

Li, Y., Huang, H., Zhao, Y., Ma, X., and Sun, J. Backdoorllm: A comprehensive benchmark for backdoor attacks on large language models. *CoRR*, abs/2408.12798, 2024. doi: 10.48550/ARXIV.2408.12798. URL https://doi.org/10.48550/arXiv.2408.12798.

Liang, Z., Ye, Q., Wang, Y., Zhang, S., Xiao, Y., Li, R., Xu, J., and Hu, H. "yes, my lord." guiding language model extraction with locality reinforced distillation, 2025. URL https://arxiv.org/abs/2409.02718.

Liu, H., Liu, Z., Tang, R., Yuan, J., Zhong, S., Chuang, Y.-N., Li, L., Chen, R., and Hu, X. Lora-as-an-attack! piercing llm safety under the share-and-play scenario, 2024. URL https://arxiv.org/abs/2403.00108.

Malladi, S., Wettig, A., Yu, D., Chen, D., and Arora, S. A kernel-based view of language model fine-tuning. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23610–23641. PMLR, 2023. URL https://proceedings.mlr.press/v202/malladi23a.html.

Mangrulkar, S., Gugger, S., Debut, L., Belkada, Y., Paul, S., and Bossan, B. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022.

Mao, Y., Ge, Y., Fan, Y., Xu, W., Mi, Y., Hu, Z., and Gao, Y. A survey on lora of large language models. *CoRR*, abs/2407.11046, 2024. doi: 10.48550/ARXIV.2407.11046. URL https://doi.org/10.48550/arXiv.2407.11046.

Naddeo, K., Bouaynaya, N., and Shterenberg, R. An information geometric perspective to adversarial attacks and defenses. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2022. doi: 10.1109/IJCNN55064.2022.9892170.

Nielsen, F. An elementary introduction to information geometry. *Entropy*, 22(10):1100, 2020.

Pan, X., Zhang, M., Sheng, B., Zhu, J., and Yang, M. Hidden trigger backdoor attack on NLP models via linguistic style manipulation. In Butler, K. R. B. and Thomas, K. (eds.), *31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022*, pp. 3611–3628. USENIX Association, 2022. URL https://www.usenix.org/conference/usenixsecurity22/presentation/pan-hidden.

Poliak, A. A survey on recognizing textual entailment as an NLP evaluation. In Eger, S., Gao, Y., Peyrard, M., Zhao, W., and Hovy, E. (eds.), *Proceedings of the*

First Workshop on Evaluation and Comparison of NLP Systems, pp. 92–109, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.eval4nlp-1.10. URL https://aclanthology.org/2020.eval4nlp-1.10/.

Rahmati, A., Moosavi-Dezfooli, S.-M., Frossard, P., and Dai, H. Geoda: A geometric framework for black-box adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Ramirez, M. A., Kim, S.-K., Hamadi, H. A., Damiani, E., Byon, Y.-J., Kim, T.-Y., Cho, C.-S., and Yeun, C. Y. Poisoning attacks and defenses on artificial intelligence: A survey, 2022. URL https://arxiv.org/abs/2202.10276.

Rényi, A. On measures of entropy and information. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability, volume 1: contributions to the theory of statistics*, volume 4, pp. 547–562. University of California Press, 1961.

Sharma, L., Graesser, L., Nangia, N., and Evci, U. Natural language understanding with the quora question pairs dataset. *CoRR*, abs/1907.01041, 2019. URL http://arxiv.org/abs/1907.01041.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pp. 1631–1642, Seattle, Washington, USA, October 2013. URL https://www.aclweb.org/anthology/D13-1170.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Vaswani, A. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Wan, A., Wallace, E., Shen, S., and Klein, D. Poisoning language models during instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Linzen, T., Chrupała, G., and Alishahi, A. (eds.), *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446/.

Wang, S., Chen, L., Jiang, J., Xue, B., Kong, L., and Wu, C. Lora meets dropout under a unified framework. In Ku, L., Martins, A., and Srikumar, V. (eds.), *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pp. 1995–2008. Association for Computational Linguistics, 2024a. URL https://aclanthology.org/2024.findings-acl.119.

Wang, Y., Liu, L., Liang, Z., Ye, Q., and Hu, H. New paradigm of adversarial training: Breaking inherent trade-off between accuracy and robustness via dummy classes, 2024b. URL https://arxiv.org/abs/2410.12671.

Warstadt, A., Singh, A., and Bowman, S. R. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*, 2018.

Xu, H., Ma, Y., Liu, H., Deb, D., Liu, H., Tang, J., and Jain, A. K. Adversarial attacks and defenses in images, graphs and text: A review, 2019. URL https://arxiv.org/abs/1909.08072.

Xu, J., Ma, M. D., Wang, F., Xiao, C., and Chen, M. Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models. In Duh, K., Gómez-Adorno, H., and Bethard, S. (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pp. 3111–3126. Association for Computational Linguistics, 2024a. doi: 10.18653/V1/2024.NAACL-LONG.171. URL https://doi.org/10.18653/v1/2024.naacl-long.171.

Xu, J., Saravanan, K., van Dalen, R., Mehmood, H., Tuckey, D., and Ozay, M. Dp-dylora: Fine-tuning transformer-based models on-device under differentially private federated learning using dynamic low-rank adaptation. *arXiv preprint arXiv:2405.06368*, 2024b.

Xu, L., Xie, H., Qin, S.-Z. J., Tao, X., and Wang, F. L. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment, 2023. URL https://arxiv.org/abs/2312.12148.

Yan, J., Yadav, V., Li, S., Chen, L., Tang, Z., Wang, H., Srinivasan, V., Ren, X., and Jin, H. Backdooring instruction-tuned large language models with virtual prompt injection. In Duh, K., Gómez-Adorno, H., and Bethard, S. (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024, pp. 6065–6086. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.NAACL-LONG.337. URL https://doi.org/10.18653/v1/2024.naacl-long.337.

Yang, W., Lin, Y., Li, P., Zhou, J., and Sun, X. Rethinking stealthiness of backdoor attack against NLP models. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 5543–5557. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.431. URL https://doi.org/10.18653/v1/2021.acl-long.431.

Yin, M., Zhang, J., Sun, J., Fang, M., Li, H., and Chen, Y. Lobam: Lora-based backdoor attack on model merging, 2024. URL https://arxiv.org/abs/2411.16746.

Zeng, Y. and Lee, K. The expressive power of low-rank adaptation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=likXVjmh3E.

Zhang, X., Hu, H., Ye, Q., Bai, L., and Zheng, H. Merinspector: Assessing model extraction risks from an attack-agnostic perspective. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, pp. 4300–4315, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400712746. doi: 10.1145/3696410.3714894. URL https://doi.org/10.1145/3696410.3714894.

Zhao, C., Fletcher, P. T., Yu, M., Peng, Y., Zhang, G., and Shen, C. The adversarial attack and detection under the fisher information metric. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33 (01):5869–5876, Jul. 2019. doi: 10.1609/aaai.v33i01.33015869. URL https://ojs.aaai.org/index.php/AAAI/article/view/4536.

Zhu, J., Greenewald, K. H., Nadjahi, K., de Ocáriz Borde, H. S., Gabrielsson, R. B., Choshen, L., Ghassemi, M., Yurochkin, M., and Solomon, J. Asymmetry in low-rank adapters of foundation models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=txRZBD8tBV.

A. Proofs

A.1. Proofs of Theorem 2.2

Proof. The Fisher information is formally defined by

$$I_\theta = \mathbb{E}_{x\sim\mathcal{D}}\left[\nabla_\theta \mathcal{L}(x,\theta)^T \nabla_\theta \mathcal{L}(x,\theta)\right], \tag{19}$$

where for neural networks, we can express the gradient as:

$$\nabla_\theta \mathcal{L}(x,\theta)^T = \left(\nabla_\theta F_\theta \cdot \nabla_{F_\theta} \mathcal{L}(x,\theta)\right)^T. \tag{20}$$

Through algebraic manipulation, we can derive:

$$\begin{aligned} I_\theta &= \mathbb{E}_{x\sim\mathcal{D}}\left[\nabla_\theta \mathcal{L}(x,\theta)^T \nabla_\theta \mathcal{L}(x,\theta)\right] \\ &= \mathbb{E}_{x\sim\mathcal{D}}\left[\nabla_{F_\theta} \mathcal{L}(x,\theta)^T \nabla_\theta F_\theta^T \cdot \nabla_\theta F_\theta \nabla_{F_\theta} \mathcal{L}(x,\theta)\right] \\ &= \mathbb{E}_{x\sim\mathcal{D}}\left[\nabla_{F_\theta} \mathcal{L}(x,\theta)^T K_{ntk}(x,x) \nabla_{F_\theta} \mathcal{L}(x,\theta)\right]. \end{aligned} \tag{21}$$

□

We can compute $\nabla_{F_\theta}\mathcal{L}(x,\theta)$ under different loss functions.

Cross-Entropy Loss. For the cross-entropy loss function, we have:

$$\begin{aligned} \mathcal{L}(x,\theta) &= -\sum_{(x,y_l)\sim\mathcal{D}} \log Z[y_l] \\ &= -\sum_{(x,y_l)\sim\mathcal{D}} \log \text{softmax}(F_\theta(x,\theta)), \end{aligned} \tag{22}$$

in which the corresponding gradient is:

$$\begin{aligned} \nabla_{F_\theta}\mathcal{L}(x,\theta) &= \nabla_{F_\theta} Z[y_l] \cdot \nabla_{Z[y_l]}\mathcal{L}(x,\theta) \\ &= Z[y_l](1 - Z[y_l]) \cdot \left(-\frac{1}{Z[y_l]}\right) \\ &= Z[y_l] - 1 \end{aligned} \tag{23}$$

This leads to the following relationship between the Fisher information matrix and the NTK:

$$\mathcal{I}_\theta = \mathbb{E}_{x\sim\mathcal{D}}\left[(Z[y_l]-1)^T K_{ntk}(x,x)(Z[y_l]-1)\right]. \tag{24}$$

Mean Square Error Loss. The mean square error loss function is defined as:

$$\mathcal{L}(x,\theta) = -\sum_{x,y_l\sim\mathcal{D}} \frac{1}{2}(y_l - F_\theta(x,\theta))^2, \tag{25}$$

where the gradient computation yields

$$\nabla_{F_\theta}\mathcal{L}(x,\theta) = F_\theta(x,\theta) - y_l. \tag{26}$$

A.2. Deduction of the NTK Function

The NTK function of full fine-tuning.

$$
\begin{aligned}
&K_{\text{ff}}^{(l,k)}(x,x') \\
&= \nabla_\theta y^{(l,k)}(x)^T \nabla_\theta y^{(l,k)}(x') \\
&= \nabla_{w\in\, W^{(l)}} y^{(l,k)}(x)^T \nabla_{w\in\, W^{(l)}} y^{(l,k)}(x') + \nabla_{\theta^{(<l)}} y^{(l,k)}(x)^T \nabla_{\theta^{(<l)}} y^{(l,k)}(x') \\
&= \nabla_{w\in\, W^{(l)}} y^{(l,k)}(x)^T \nabla_{w\in\, W^{(l)}} y^{(l,k)}(x') \\
&\quad + \partial_{y_a^{(l-1)}} y^{(l,k)}(x) \partial_{y^{(l-1)}} y_a^{(l-1)}(x) \partial_{\theta^{(<l)}} y^{(l-1)}(x) \partial_{\theta^{(<l)}} y^{(l-1)}(x')^T \partial_{y^{(l-1)}} y_a^{(l-1)}(x')^T \partial_{y_a^{(l-1)}} y^{(l,k)}(x')^T \\
&= y_a^{(l-1)}(x)^T \cdot y_a^{(l-1)}(x') + \underbrace{W^{(l,k)} \dot\sigma(y^{(l-1)}(x))}_{\text{a scalar}} \underbrace{\phi^{(l-1)}(x)^T \phi^{(l-1)}(x')}_{\text{a scalar}} \underbrace{\dot\sigma(y^{(l-1)}(x'))^T W^{(l,k)\ T}}_{\text{a scalar}} \\
&= y_a^{(l-1)}(x)^T \cdot y_a^{(l-1)}(x') + \underbrace{\phi^{(l-1)}(x)^T \phi^{(l-1)}(x')}_{K_{\text{ff}}^{(l-1,k)}(x,x')} \dot\sigma(y^{(l-1)}(x'))^T W^{(l,k)\ T} W^{(l,k)} \dot\sigma(y^{(l-1)}(x)).
\end{aligned}
\tag{27}
$$

Based on the assumption of NTK that: *i)* $W^{(l)}$ *is initialized with the expectation of* 0 *and variance of* $1/\sqrt{n_{l-1}}$*; and ii)* $n_{l-1} \to \infty$, we can derive $W^{(l,k)\ T} W^{(l,k)} \to I_{n_{l-1}\times n_{l-1}}$, an identity matrix, suggesting that the NTK of full fine-tuning can be formalized as

$$
\begin{aligned}
&K_{\text{ff}}^{(l,k)}(x,x') \\
&= \underbrace{y_a^{(l-1)}(x)^T \cdot y_a^{(l-1)}(x')}_{\Sigma^{(l)}(x,x')} + \underbrace{\phi^{(l-1)}(x)^T \phi^{(l-1)}(x')}_{K_{\text{ff}}^{(l-1,k)}(x,x')} \dot\sigma(y^{(l-1)}(x'))^T \underbrace{W^{(l,k)\ T} W^{(l,k)}}_{I_{n_{l-1}\times\ n_{l-1}}} \dot\sigma(y^{(l-1)}(x)) \\
&= \underbrace{y_a^{(l-1)}(x)^T \cdot y_a^{(l-1)}(x')}_{\Sigma^{(l)}(x,x')} + \underbrace{\phi^{(l-1)}(x)^T \phi^{(l-1)}(x')}_{K_{\text{ff}}^{(l-1,k)}(x,x')} \underbrace{\dot\sigma(y^{(l-1)}(x'))^T \dot\sigma(y^{(l-1)}(x))}_{\dot\Sigma^{(l)}(x,x')} \\
&= \Sigma^{(l)}(x,x') + K_{\text{ff}}^{(l-1,k)}(x,x') \dot\Sigma^{(l)}(x,x').
\end{aligned}
\tag{28}
$$

□

The NTK function of LoRA.

$$
\begin{aligned}
&K_{\text{LoRA}}^{(l,k)}(x,x') \\
&= \nabla_\theta y^{(l,k)}(x)^T \nabla_\theta y^{(l,k)}(x') \\
&= \nabla_{B^{(l)}} y^{(l,k)}(x)^T \nabla_{B^{(l)}} y^{(l,k)}(x') + \nabla_{A^{(l)}} y^{(l,k)}(x)^T \nabla_{A^{(l)}} y^{(l,k)}(x') + \nabla_{\theta^{(<l)}} y^{(l,k)}(x)^T \nabla_{\theta^{(<l)}} y^{(l,k)}(x') \\
&= \nabla_{B^{(l)}} y^{(l,k)}(x)^T \nabla_{B^{(l)}} y^{(l,k)}(x') + \partial_{z^{(l)}} y^{(l,k)}(x) \partial_{A^{(l)}} z^{(l)}(x) \partial_{A^{(l)}} z^{(l)}(x')^T \partial_{z^{(l)}} y^{(l,k)}(x')^T \\
&\quad + \partial_{y_a^{(l-1)}} y^{(l,k)}(x) \partial_{y^{(l-1)}} y_a^{(l-1)}(x) \partial_{\theta^{(<l)}} y^{(l,k)}(x) \partial_{\theta^{(<l)}} y^{(l-1)}(x')^T \partial_{y^{(l-1)}} y_a^{(l-1)}(x')^T \partial_{y_a^{(l-1)}} y^{(l,k)}(x')^T \\
&= z^{(l-1)}(x)^T \cdot z^{(l-1)}(x') + B^{(l,k)} \cdot\ I_r \otimes \sigma(y^{(l-1)}(x))^T \sigma(y^{(l-1)}(x')) \otimes\ I_r^T \cdot\ B^{(l,k)\ T} \\
&\quad + (W_0^{(l,k)} + B^{(l,k)} A^{(l,k)}) \dot\sigma(y^{(l-1)}(x)) \phi^{(l-1)}(x)^T \phi^{(l-1)}(x') \dot\sigma(y^{(l-1)}(x'))^T (W_0^{(l,k)} + B^{(l,k)} A^{(l,k)})^T \\
&= y_a^{(l-1)}(x)^T A^{(l)\ T} A^{(l)} y_a^{(l-1)}(x') + B^{(l,k)} \cdot\ I_r \otimes \underbrace{\sigma(y^{(l-1)}(x))^T \sigma(y^{(l-1)}(x'))}_{\text{a scalar}} \otimes I_r^T \cdot\ B^{(l,k)\ T} \\
&\quad + \underbrace{(W_0^{(l,k)} + B^{(l,k)} A^{(l,k)}) \dot\sigma(y^{(l-1)}(x))}_{\text{a scalar}} \underbrace{\phi^{(l-1)}(x)^T \phi^{(l-1)}(x')}_{\text{a scalar}} \underbrace{\dot\sigma(y^{(l-1)}(x'))^T (W_0^{(l,k)} + B^{(l,k)} A^{(l,k)})^T}_{\text{a scalar}} \\
&= y_a^{(l-1)}(x)^T A^{(l)\ T} A^{(l)} y_a^{(l-1)}(x') + \underbrace{\sigma(y^{(l-1)}(x))^T \sigma(y^{(l-1)}(x'))}_{\dot\Sigma^{(l)}(x,x')} \underbrace{B^{(l,k)} \cdot\ B^{(l,k)\ T}}_{\text{a scalar}} \\
&\quad + \underbrace{\phi^{(l-1)}(x)^T \phi^{(l-1)}(x')}_{K_{\text{LoRA}}^{(l-1,k)}(x,x')} \dot\sigma(y^{(l-1)}(x'))^T (W_0^{(l,k)} + B^{(l,k)} A^{(l,k)})^T (W_0^{(l,k)} + B^{(l,k)} A^{(l,k)}) \dot\sigma(y^{(l-1)}(x)).
\end{aligned}
\tag{29}
$$

In LoRA, the matrix $B^{(l)}$ is initialized to the zero matrix $\mathbf{0}$, indicating that $W_{\text{LoRA}}^{(l)} = W_0^{(l)} + B^{(l)}A^{(l)} = W_0^{(l)} + \mathbf{0} \cdot A^{(l)} = W_0^{(l)}$ at the begin of training. Similar to the NTK of full fine-tuned models, we can also demonstrate that ${W_{\text{LoRA}}^{(l)}}^T W_{\text{LoRA}}^{(l)} \to I_{n_{l-1} \times n_{l-1}}$.

Therefore, we have

$$
\begin{aligned}
&K_{\text{LoRA}}^{(l,k)}(x, x') \\
&= y_a^{(l-1)}(x)^T {A^{(l)}}^T A^{(l)} y_a^{(l-1)}(x') + \underbrace{\sigma(y^{(l-1)}(x))^T \sigma(y^{(l-1)}(x'))}_{\dot{\Sigma}^{(l)}(x,x')} \underbrace{B^{(l,k)} \cdot {B^{(l,k)}}^T}_{\text{a scalar}} \\
&\quad + \underbrace{\phi^{(l-1)}(x)^T \phi^{(l-1)}(x')}_{K_{\text{LoRA}}^{(l-1,k)}(x,x')} \dot{\sigma}(y^{(l-1)}(x'))^T (W_0^{(l,k)} + B^{(l,k)}A^{(l,k)})^T (W_0^{(l,k)} + B^{(l,k)}A^{(l,k)}) \dot{\sigma}(y^{(l-1)}(x)). \\
&= \underbrace{y_a^{(l-1)}(x)^T {A^{(l)}}^T A^{(l)} y_a^{(l-1)}(x')}_{\Sigma_{\text{LoRA}}^{(l)}(x,x')} + \underbrace{\sigma(y^{(l-1)}(x))^T \sigma(y^{(l-1)}(x'))}_{\dot{\Sigma}^{(l)}(x,x')} \underbrace{\mathbf{0}_{1\times\, n_l} \cdot \mathbf{0}_{n_l}}_{0} \\
&\quad + \underbrace{\phi^{(l-1)}(x)^T \phi^{(l-1)}(x')}_{K_{\text{LoRA}}^{(l-1,k)}(x,x')} \dot{\sigma}(y^{(l-1)}(x'))^T \underbrace{(W_0^{(l,k)} + B^{(l,k)}A^{(l,k)})^T (W_0^{(l,k)} + B^{(l,k)}A^{(l,k)})}_{I_{n_{l-1}\times n_{l-1}}} \dot{\sigma}(y^{(l-1)}(x)). \\
&= \underbrace{y_a^{(l-1)}(x)^T {A^{(l)}}^T A^{(l)} y_a^{(l-1)}(x')}_{\Sigma_{\text{LoRA}}^{(l)}(x,x')} + \underbrace{\phi^{(l-1)}(x)^T \phi^{(l-1)}(x')}_{K_{\text{LoRA}}^{(l-1,k)}(x,x')} \underbrace{\dot{\sigma}(y^{(l-1)}(x'))^T \dot{\sigma}(y^{(l-1)}(x))}_{\dot{\Sigma}_{\text{LoRA}}^{(l)}(x,x')} \\
&= \Sigma_{\text{LoRA}}^{(l)}(x, x') + K_{\text{LoRA}}^{(l-1)}(x, x') \dot{\Sigma}_{LoRA}^{(l)}(x, x') \\
&= \Sigma_{\text{LoRA}}^{(l)}(x, x') + K_{\text{LoRA}}^{(l-1)}(x, x') \cdot \dot{\Sigma}^{(l)}(x, x').
\end{aligned}
\tag{30}
$$

□

The aforementioned theoretical analysis primarily focuses on artificial neural networks (ANNs) initialized with random weights. However, in more practical scenarios, particularly in continuous fine-tuning settings, empirical observations demonstrate that the network dynamics remain within the Neural Tangent Kernel (NTK) regime. This phenomenon is further supported by experimental evidence presented in Section D.

A.3. Proofs of Theorem 3.3

Proof. Leveraging the two properties of NTK, we establish that NTK functions keep constant during the training procedure. Consequently, our analysis focuses on deriving the relationship between $K_{\text{LoRA}}^{(l)}$ and $K_{\text{ff}}^{(l)}$ at the initialization stage.

In LoRA, the weight matrix is typically initialized as $A^{(l)} \sim \mathcal{P}(0, \sigma^2)$ and $B^{(l+1)} = 0$, where $\mathcal{P}(0, \sigma^2)$ denotes a probability distribution with the expectation 0 and variance σ^2. This class of distributions encompasses common initialization schemes such as Gaussian distribution, Kaiming distribution, and so on. At initialization, we observe the following equivalence:

$$
W_{\text{ff}}^{(l)} = W_0^{(l)} = W_0^{(l)} + \mathbf{0} = W_0^{(l)} + B^{(l)}A^{(l)} = W_{\text{LoRA}}^{(l)}. \tag{31}
$$

Consequently, we can derive that $\dot{\Sigma}_{\text{LoRA}}^{(l)}(x, x') = \dot{\Sigma}_{\text{ff}}^{(l)}(x, x') = \dot{\Sigma}^{(l)}$.

Building upon Theorem 3.3, which states that the first $l-1$ layers maintain identical configurations between full fine-tuning and LoRA, we know that $K_{\text{LoRA}}^{(l-1)} = K_{\text{ff}}^{(l-1)} = K^{(l-1)}$ and $y_{\text{LoRA}}^{(l-1)}(x) = y_{\text{ff}}^{(l-1)}(x)$ and $y_{\text{LoRA}}^{(l-1)}(x') = y_{\text{ff}}^{(l-1)}(x')$. The NTK functions for both LoRA and full fine-tuning can be formatted as

$$
\begin{aligned}
K_{\text{ff}}^{(l,k)}(x, x') &= K^{(l-1,k)}\dot{\Sigma}^{(l)} + \Sigma_{\text{ff}}^{(l)}(x, x') = K^{(l-1,k)}\dot{\Sigma}^{(l)} + \sigma(y^{(l-1)}(x))^T \cdot \sigma(y^{(l-1)}(x')) \\
K_{\text{LoRA}}^{(l,k)}(x, x') &= K^{(l-1,k)}\dot{\Sigma}^{(l)} + \Sigma_{\text{LoRA}}^{(l)}(x, x') = K^{(l-1,k)}\dot{\Sigma}^{(l)} + \sigma(y^{(l-1)}(x))^T {A^{(l)}}^T \cdot A^{(l)} \sigma(y^{(l-1)}(x')).
\end{aligned}
\tag{32}
$$

Through algebraic manipulation, we derive their fundamental relationship:

$$
K_{\text{LoRA}}^{(l,k)} = K_{\text{ff}}^{(l,k)} + \Delta_r^{(l,k)}, \tag{33}
$$

where the residual term is defined as:

$$\Delta_r^{(l,k)} = [\sigma(y^{(l-1)}(x))]^T (A^{(l)\ T} A^{(l)} - I_{n_{l-1} \times n_{l-1}})[\sigma(y^{(l-1)}(x'))].$$

□

A.4. Proofs of Theorem 3.4

Proof. Leveraging the fundamental properties of matrix rank, we have:

$$\mathbf{rank}(A^{(l)T} A^{(l)}) \leq \mathbf{rank}(A^{(l)}) \leq r. \tag{34}$$

The condition $\mathbf{rank}(A^{(l)T} A^{(l)}) \leq r$ indicates that there at most exist $n - r$ nonzero eigenvalues in $A^{(l)T} A^{(l)}$.

Given the initialization conditions $\mathbb{E}[A^{(l)}] = \mathbf{0}$ and $\mathbf{Var}(A^{(l)}) = \sigma^2$, we derive the following expectation for any column index $p = \{0, 1, 2, ..., n-1\}$:

$$\mathbb{E}[A_{\cdot,p}^{(l)T} A_{\cdot,p}^{(l)}] = \mathbb{E}[\sum_{q=1}^{r} A_{q,p}^{(l)} \cdot\ A_{q,p}^{(l)}] = r\sigma^2. \tag{35}$$

The expected trace of $A^{(l)T} A$ can be formalized by

$$\mathbb{E}[\mathbf{tr}(A^{(l)T} A^{(l)})] = \mathbb{E}[\sum_{p=1}^{n_{l-1}} \sum_{q=1}^{r} A_{q,p}^{(l)} \cdot\ A_{q,p}^{(l)}] = n_{l-1} \cdot r\sigma^2. \tag{36}$$

Considering the eigenvalue distribution of $A^{(l)T} A^{(l)}$, we note that $n_{l-1} - r$ of them are 0, while the remaining r eigenvalues are identically distributed with the same expectation. Thus, the expected value of the rest r eigenvalues is $\mathbb{E}_{\lambda \in \text{Eigen}\{A^{(l)T} A^{(l)}\}}[\lambda_i] = n_{l-1} \cdot \sigma^2$. When $n_{l-1} \to \infty$, all nonzero eigenvalues of $A^{(l)T} A$ converge to $n_{l-1}\sigma^2$, where if $\sigma^2 < \frac{1}{n_{l-1}}$, they are smaller than $n_{l-1} \cdot 1/n_{l-1} = 1$. In conclusion, we proof that all eigenvalues of $A^{(l)T} A$ are smaller than 1 if $\sigma^2 < \frac{1}{n_{l-1}}$. Consequently, $A^{(l)T} A - I$ exhibits exclusively non-positive eigenvalues, proving that $A^{(l)T} A - I$ is negative semi-definite when $r < n_{l-1}$ and $\sigma^2 \leq \frac{1}{n_{l-1}}$. □

Proof of Corollary 3.5. Building upon our theoretical analysis, we establish the proof of Corollary 3.5.

When $r = n_{l-1}$, the matrix $A \in \mathbb{R}^{n_{l-1} \times n_{l-1}}$ becomes square. The expectation of its Gram matrix entries is given by: and

$$\mathbb{E}[(A^{(l)T} A^{(l)})_{p,q}] = \mathbb{E}[\sum_{u=1}^{n_{l-1}} A_{u,p}^{(l)} \cdot\ A_{q,u}^{(l)}]. \tag{37}$$

Under the weight initialization scheme, we have $\mathbb{E}[A_{u,v}] = 0$ and $\mathbf{Var}(A_{u,v}) = \sigma^2$ for all $u, v \in \{1, ..., n_{l-1}\}$, with independent entries. This leads to the following cases:

- For off-diagonal entries ($q \neq p$):

$$\mathbb{E}[A_{u,p} \cdot A_{q,u}] = \mathbb{E}[A_{u,p}] \cdot \mathbb{E}[A_{q,u}] = 0. \tag{38}$$

- For diagonal entries ($p = q$), analogous to Equation 35:

$$\mathbb{E}[(A^{(l)T} A^{(l)})_{p,q}] = \mathbb{E}[\sum_{u=1}^{n_{l-1}} A_{u,p}^{(l)} \cdot\ A_{q,u}^{(l)}] = n_{l-1} \cdot \sigma^2. \tag{39}$$

When the initialization variance satisfies $\sigma^2 = 1/n_{l-1}$, the diagonal entries simplify to $\mathbb{E}[(A^{(l)T} A^{(l)})_{p,q}] = 1$.

Consequently, when $n_{l-1} \to \infty$, we conclude that $A^{(l)T} A^{(l)} \to I$. □

A.5. Proofs of Theorem 3.6

Proof. Base on Theorem 3.4, we know that when $r \leq n_{l-1}$ and $\sigma^2 \leq 1/n_{l-1}$, the kernel matrix M_Δ^l is negative semi-definite. This implies that all M_Δ^l's eigenvalues $\lambda_\Delta^l \leq 0$.

Then $\forall\, \nabla_{F_\theta}\mathcal{L}(x,\theta) \in \mathbb{R}^{n_L}$, we derive the following inequalities:

$$\begin{aligned}
&\nabla_{F_\theta}\mathcal{L}(x,\theta)^T \Delta_r(x,x) \nabla_{F_\theta}\mathcal{L}(x,\theta) \leq 0 \\
&\Rightarrow \nabla_{F_\theta}\mathcal{L}(x,\theta)^T K_{\text{LoRA}}(x,x) \nabla_{F_\theta}\mathcal{L}(x,\theta) \leq \nabla_{F_\theta}\mathcal{L}(x,\theta)^T K_{\text{FF}}(x,x) \nabla_{F_\theta}\mathcal{L}(x,\theta) \\
&\Rightarrow \mathcal{I}_{\theta\ \text{LoRA}} \leq \mathcal{I}_{\theta\ \text{FF}}
\end{aligned} \tag{40}$$

Then $\forall\, \lambda_{\text{LoRA}} \in \text{Eigen}(\mathcal{I}^I_{\theta\ \text{LoRA}})$ and $\forall\, \lambda^I_{\text{FF}} \in \text{Eigen}(\mathcal{I}_{\theta\ \text{FF}})$, we have

$$\lambda^I_{\text{LoRA}} \leq \lambda^I_{\text{FF}}. \tag{41}$$

This eigenvalue relationship leads to the following important results:

$$\begin{aligned}
&\frac{1}{2}\sum_{\lambda^I_{\theta\ \text{LoRA}}} \lambda^I_{\theta\ \text{LoRA}} \leq \frac{1}{2}\sum_{\lambda^I_{\theta\ \text{ff}}} \lambda^I_{\theta\ \text{ff}} \\
&\Rightarrow \mathbf{IB}_{\text{LoRA}} \leq \mathbf{IB}_{\text{ff}}
\end{aligned} \tag{42}$$

and

$$\begin{aligned}
&\frac{1}{1-\alpha}\log\left(\sum_{i=1}^{n_L} \lambda^I_{\theta\ \text{LoRA}}\right) \leq \frac{1}{1-\alpha}\log\left(\sum_{i=1}^{n_L} \lambda^I_{\theta\ \text{FF}}\right) \\
&\Rightarrow H_{\alpha\text{LoRA}} \leq H_{\alpha\text{ff}}.
\end{aligned} \tag{43}$$

□

A.6. Proofs of Theorems beyond the OOLD Assumption

A.6.1. Proofs of Theorem 3.4 beyond the OOLD Assumption

Proof. Let $K_{\text{ff}}^{(l,k)'}$ and $K_{\text{LoRA}}^{(l,k)'}$ denote the NTKs of FF and LoRA beyond the OOLD assumption. From Equation 12 and Equation 14, we derive the difference of initialized NTK functions as follows:

$$\begin{aligned}
\Delta^{(1,k)'} &= K_{\text{LoRA}}^{(1,k)'} - K_{\text{ff}}^{(1,k)'} = 0; \\
\Delta^{(2,k)'} &= K_{\text{LoRA}}^{(2,k)'} - K_{\text{ff}}^{(2,k)'} \\
&= (K_{\text{LoRA}}^{(1,k)} - K_{\text{ff}}^{(1,k)})\dot{\Sigma}^{(2)} + \sigma(y^{(1)}(x))^T A^{(2)\ T} A^{(2)} \sigma(y^{(1)}(x)) - \sigma(y^{(1)}(x))^T \sigma(y^{(1)}(x)) \\
&= \sigma(y^{(1)}(x))^T (A^{(2)\ T} A^{(2)} - I)\sigma(y^{(1)}(x)); \\
\Delta^{(l,k)'} &= K_{\text{LoRA}}^{(l,k)'} - K_{\text{ff}}^{(l,k)'} \\
&= (K_{\text{LoRA}}^{(l-1,k)'} - K_{\text{ff}}^{(l-1,k)'})\dot{\Sigma}^{(l)} + \sigma(y^{(l-1)}(x))^T A^{(l)\ T} A^{(l)} \sigma(y^{(l-1)}(x)) - \sigma(y^{(l-1)}(x))^T \sigma(y^{(l-1)}(x)) \\
&= \Delta^{(l-1,k)'}\dot{\Sigma}^{(l)} + \sigma(y^{(l-1)}(x))^T A^{(l)\ T} A^{(l)} \sigma(y^{(l-1)}(x)) - \sigma(y^{(l-1)}(x))^T \sigma(y^{(l-1)}(x)) \\
&= \Delta^{(l-1,k)'}\dot{\Sigma}^{(l)} + \sigma(y^{(l-1)}(x))^T (A^{(l)\ T} A^{(l)} - I)\sigma(y^{(l-1)}(x)) \\
&= \Delta^{(l-1,k)'}\dot{\Sigma}^{(l)} + \Delta_r^{(l)}.
\end{aligned} \tag{44}$$

By Theorem 3.4, the matrix $A^{(l)\ T} A^{(l)} - I$ is negative semi-definite when $\sigma_a^2 < 1/n_{l-1}$ and $r \leq n_{l-1}$. Consequently, $\forall\, y^{(1)(x)} \in \mathbb{R}^{n_1}$, we have $\Delta^{(2,k)'} \leq 0$ and $\Delta^{(l,k)} \leq 0$. Moreover, since $\forall\, y^{(l)} \in \mathbb{R}^{n_l}, \dot{\sigma}(y^{(l)}) \geq 0$, it follows that $\Delta^{(l,k)'} \geq 0$ for $l = 3, ..., L$. In conclusion, $\Delta^{(l,k)'} \geq 0$ holds for $l = 1, ..., L$. □

A.6.2. Proofs of Corollary 3.5 beyond the OOLD Assumption

Proof. When $\sigma_a^2 = 1/n_{l-1}$ and $r = n_{l-1}$, the matrix $A^{(l)\,T} A^{(l)} - I \to \mathbf{0}$ when $n_l \to \infty$.

Given $A^{(l)\,T} A^{(l)} - I = \mathbf{0}$, we obtain $\Delta^{(2,k)'} = 0$, and $\Delta^{(l,k)'} = \Delta^{(l-1,k)'} \dot{\Sigma}^{(l)} + \Delta_r^{(l)}$ for $l = 3, ..., L$, where

$$\begin{aligned} \Delta^{(l,k)'} &= \Delta^{(l-1,k)'} \dot{\Sigma}^{(l)} + \Delta_r^{(l)} \\ &= \mathbf{0} \dot{\Sigma}^{(l)} + \mathbf{0} = \mathbf{0}. \end{aligned} \tag{45}$$

Thus, $\Delta^{(l,k)'} = \mathbf{0}$ for all layers l. □

A.6.3. Proof of Theorem 3.6 beyond the OOLD Assumption.

The proof follows a similar structure to the proof provided in Appendix A.5.

Proof. Base on Theorem 3.4, when $r \leq n_{l-1}$ and $\sigma^2 \leq 1/n_{l-1}$, the kernel matrix M_Δ^l is negative semi-definite, implying that all of the M_Δ^l's eigenvalues $\lambda_\Delta^l \leq 0$.

$\forall\, \nabla_{F_\theta} \mathcal{L}(x, \theta) \in \mathbb{R}^{n_L}$, we derive the following inequalities:

$$\begin{aligned} &\nabla_{F_\theta} \mathcal{L}(x, \theta)^T \Delta_r(x, x) \nabla_{F_\theta} \mathcal{L}(x, \theta) \leq 0 \\ &\Rightarrow \nabla_{F_\theta} \mathcal{L}(x, \theta)^T K_{\text{LoRA}}(x, x) \nabla_{F_\theta} \mathcal{L}(x, \theta) \leq \nabla_{F_\theta} \mathcal{L}(x, \theta)^T K_{\text{FF}}(x, x) \nabla_{F_\theta} \mathcal{L}(x, \theta) \\ &\Rightarrow \mathcal{I}_{\theta\ \text{LoRA}} \leq \mathcal{I}_{\theta\ \text{FF}} \end{aligned} \tag{46}$$

This implies that $\forall\, \lambda_{\text{LoRA}} \in \text{Eigen}(\mathcal{I}^I_{\theta\ \text{LoRA}})$ and $\forall\, \lambda^I_{\text{FF}} \in \text{Eigen}(\mathcal{I}_{\theta\ \text{FF}})$, we have

$$\lambda^I_{\text{LoRA}} \leq \lambda^I_{\text{FF}}. \tag{47}$$

Consequently, we establish the following results:

$$\begin{aligned} &\frac{1}{2} \sum_{\lambda^I_{\theta\ \text{LoRA}}} \lambda^I_{\theta\ \text{LoRA}} \leq \frac{1}{2} \sum_{\lambda^I_{\theta\ \text{ff}}} \lambda^I_{\theta\ \text{ff}} \\ &\Rightarrow \mathbf{IB}_{\text{LoRA}} \leq \mathbf{IB}_{\text{ff}} \end{aligned} \tag{48}$$

and

$$\begin{aligned} &\frac{1}{1-\alpha} \log \left(\sum_{i=1}^{n_L} \lambda^I_{\theta\ \text{LoRA}} \right) \leq \frac{1}{1-\alpha} \log \left(\sum_{i=1}^{n_L} \lambda^I_{\theta\ \text{FF}} \right) \\ &\Rightarrow H_{\alpha\text{LoRA}} \leq H_{\alpha\text{ff}}. \end{aligned} \tag{49}$$

□

B. Analysis on the Transformer

Proposition B.1. *Under the OOLD assumption, the application of LoRA to either the embedding layer, the feedforward module, the self-attention module, or the linear classification head preserves the validity of Theorem 3.4 Corollary 3.5 and Theorem 3.6.*

The proof of Proposition B.1 on embedding layers, feedforward layers, and the linear head follows directly from the mathematical derivation applicable to ANNs. Therefore, here we focus on the analysis on the self-attention mechanism.

Architecture of the Standard Transformer Module. Given three learnable weight matrices $W_Q^{(l)}, W_K^{(l)}, W_V^{(l)} \in \mathbb{R}^{d \times d}$,

for an input hidden state $x^{(l)}$, the feed forward procedure for a standard self-attention module can be expressed as follows:

$$\begin{aligned}
&Q^{(l)}, K^{(l)}, V^{(l)} = W_Q^{(l)} \cdot x^{(l)}, W_K^{(l)} \cdot x^{(l)}, W_V^{(l)} \cdot x^{(l)};\\
&\alpha_a^{(l)} = \frac{Q^{(l)T} \cdot K^{(l)}}{\sqrt{d}} = \frac{(W_Q^{(l)} x^{(l)})^T (W_K^{(l)} x^{(l)})}{\sqrt{d}} = \frac{x^{(l)T} W_Q^{(l)T} W_K^{(l)} x^{(l)}}{\sqrt{d}};\\
&\alpha^{(l)} = \mathbf{SM}\left(\alpha_a^{(l)}\right) = \mathbf{SM}\left(\frac{x^{(l)T} W_Q^{(l)T} W_K^{(l)} x^{(l)}}{\sqrt{d}}\right);\\
&x_{\text{attn}}^{(l)} = \alpha^{(l)} \cdot V^{(l)} = \mathbf{SM}\left(\frac{x^{(l)T} W_Q^{(l)T} W_K^{(l)} x^{(l)}}{\sqrt{d}}\right) W_V^{(l)} x^{(l)},
\end{aligned} \tag{50}$$

where $\mathbf{SM}(\cdot)$ denotes the softmax function.

Based on Equation 50, we can derive the gradients of parameters. As an example, the derivative of $W_k^{(l)}$ is computed by

$$\begin{aligned}
\partial_{W_K^{(l)}} x_{\text{attn}^{(l)}} &= \partial_{\alpha^{(l)}} x_{\text{attn}^{(l)}} \cdot \partial_{\alpha_a^{(l)}} \alpha^{(l)} \partial_{W_K^{(l)}} \alpha_a^{(l)}\\
&= I_d \otimes (W_V^{(l)} x^{(l)})^T \mathbf{S\dot{M}}(\alpha_a^{(l)}) \frac{(W_Q^{(l)} x^{(l)})^T}{\sqrt{d}} I_d \otimes x^{(l)}.
\end{aligned} \tag{51}$$

Based on Equation 51, the NTK function can be formatted as:

$$\begin{aligned}
&K_{\text{attn;ff}}^{(l)}(x, x')\\
&= \nabla_{\theta_{\text{attn}}} x_{\text{attn}}^{(l)T} \cdot \nabla_{\theta_{\text{attn}}} x_{\text{attn}}^{(l)}\\
&= \nabla_{\theta_{\text{attn}}^{(l)}} x_{\text{attn}}^{(l)T} \cdot \nabla_{\theta_{\text{attn}}^{(l)}} x_{\text{attn}}^{(l)} + \nabla_{\theta_{\text{attn}}^{(<l)}} x_{\text{attn}}^{(l)T} \cdot \nabla_{\theta_{\text{attn}}^{(<l)}} x_{\text{attn}}^{(l)}\\
&= \nabla_{W_Q^{(l)}} x_{\text{attn}}^{(l)T} \cdot \nabla_{W_Q^{(l)}} x_{\text{attn}}^{(l)} + \nabla_{W_K^{(l)}} x_{\text{attn}}^{(l)T} \cdot \nabla_{W_K^{(l)}} x_{\text{attn}}^{(l)} + \nabla_{W_V^{(l)}} x_{\text{attn}}^{(l)T} \cdot \nabla_{W_V^{(l)}} x_{\text{attn}}^{(l)} + \nabla_{\theta_{\text{attn}}^{(<l)}} x_{\text{attn}}^{(l)T} \cdot \nabla_{\theta_{\text{attn}}^{(<l)}} x_{\text{attn}}^{(l)}\\
&= I_d \otimes (W_V^{(l)} x^{(l)})^T \mathbf{S\dot{M}}(\alpha_a^{(l)}) (\frac{W_Q^{(l)} x^{(l)}}{\sqrt{d}})^T I_d \otimes x^{(l)T} \cdot x^{(l)'} \otimes I_d^T (\frac{W_Q^{(l)} x^{(l)'}}{\sqrt{d}}) \mathbf{S\dot{M}}(\alpha_a^{(l)}) (W_V^{(l)} x^{(l)'}) \otimes I_d^T\\
&\quad + \nabla_{W_Q^{(l)}} x_{\text{attn}}^{(l)T} \cdot \nabla_{W_Q^{(l)}} x_{\text{attn}}^{(l)} + \nabla_{W_V^{(l)}} x_{\text{attn}}^{(l)T} \cdot \nabla_{W_V^{(l)}} x_{\text{attn}}^{(l)} + \nabla_{\theta_{\text{attn}}^{(<l)}} x_{\text{attn}}^{(l)T} \cdot \nabla_{\theta_{\text{attn}}^{(<l)}} x_{\text{attn}}^{(l)}.
\end{aligned} \tag{52}$$

When approximating $W_K^{(l)}$ with LoRA during fine-tuning, i.e., $W_K^{(l)} = W_{K0}^{(l)} + B_{W_K^{(l)}} A_{W_K^{(l)}}$, the NTK function can be derived as:

$$\begin{aligned}
&K_{\text{attn;LoRA}}^{(l)}(x, x')\\
&= \nabla_{\theta_{\text{attn}}} x_{\text{attn}}^{(l)T} \cdot \nabla_{\theta_{\text{attn}}} x_{\text{attn}}^{(l)}\\
&= \nabla_{\theta_{\text{attn}}^{(l)}} x_{\text{attn}}^{(l)T} \cdot \nabla_{\theta_{\text{attn}}^{(l)}} x_{\text{attn}}^{(l)} + \nabla_{\theta_{\text{attn}}^{(<l)}} x_{\text{attn}}^{(l)T} \cdot \nabla_{\theta_{\text{attn}}^{(<l)}} x_{\text{attn}}^{(l)}\\
&= \nabla_{W_Q^{(l)}} x_{\text{attn}}^{(l)T} \cdot \nabla_{W_Q^{(l)}} x_{\text{attn}}^{(l)} + \nabla_{W_K^{(l)}} x_{\text{attn}}^{(l)T} \cdot \nabla_{W_K^{(l)}} x_{\text{attn}}^{(l)} + \nabla_{W_V^{(l)}} x_{\text{attn}}^{(l)T} \cdot \nabla_{W_V^{(l)}} x_{\text{attn}}^{(l)} + \nabla_{\theta_{\text{attn}}^{(<l)}} x_{\text{attn}}^{(l)T} \cdot \nabla_{\theta_{\text{attn}}^{(<l)}} x_{\text{attn}}^{(l)}\\
&= I_d \otimes (W_V^{(l)} x^{(l)})^T \mathbf{S\dot{M}}(\alpha_a^{(l)}) (\frac{W_Q^{(l)} x^{(l)}}{\sqrt{d}})^T I_d \otimes x^{(l)T} A_{W_K^{(l)}}^T \cdot A_{W_K^{(l)}} x^{(l)'} \otimes I_d^T (\frac{W_Q^{(l)} x^{(l)'}}{\sqrt{d}}) \mathbf{S\dot{M}}(\alpha_a^{(l)}) (W_V^{(l)} x^{(l)'}) \otimes I_d^T\\
&\quad + \nabla_{W_Q^{(l)}} x_{\text{attn}}^{(l)T} \cdot \nabla_{W_Q^{(l)}} x_{\text{attn}}^{(l)} + \nabla_{W_V^{(l)}} x_{\text{attn}}^{(l)T} \cdot \nabla_{W_V^{(l)}} x_{\text{attn}}^{(l)} + \nabla_{\theta_{\text{attn}}^{(<l)}} x_{\text{attn}}^{(l)T} \cdot \nabla_{\theta_{\text{attn}}^{(<l)}} x_{\text{attn}}^{(l)}.
\end{aligned} \tag{53}$$

Let $\mathbf{V}_{\text{attn}}^{(l)}(x) = A_{W_K^{(l)}} x^{(l)} \otimes I_d^T (\frac{W_Q^{(l)} x^{(l)}}{\sqrt{d}}) \mathbf{S\dot{M}}(\alpha_a^{(l)}) (W_V^{(l)} x^{(l)}) \otimes I_d^T$ and $K_{\text{others}}^{(l)} = \nabla_{W_Q^{(l)}} x_{\text{attn}}^{(l)T} \cdot \nabla_{W_Q^{(l)}} x_{\text{attn}}^{(l)} + \nabla_{W_V^{(l)}} x_{\text{attn}}^{(l)T} \cdot \nabla_{W_V^{(l)}} x_{\text{attn}}^{(l)} + \nabla_{\theta_{\text{attn}}^{(<l)}} x_{\text{attn}}^{(l)T} \cdot \nabla_{\theta_{\text{attn}}^{(<l)}} x_{\text{attn}}^{(l)}$. Then, the NTK functions for full fine-tuning and LoRA can be simplified as:

$$\begin{aligned}
K_{\text{attn;ff}}^{(l)}(x, x') &= \mathbf{V}_{\text{attn}}^{(l)T}(x) \cdot \mathbf{V}_{\text{attn}}^{(l)}(x') + K_{\text{others}}^{(l)}(x, x;')\\
K_{\text{attn;LoRA}}^{(l)}(x, x') &= \mathbf{V}_{\text{attn}}^{(l)T}(x) A_{W_K^{(l)}}^T \cdot A_{W_K^{(l)}} \mathbf{V}_{\text{attn}}^{(l)}(x') + K_{\text{others}}^{(l)}(x, x;').
\end{aligned} \tag{54}$$

Based on Theorem 3.4, it is established that $A_{W_K^{(l)}}^T A_{W_K^{(l)}} - I$ is *negative semi-definite* under the specified conditions. This property directly leads to the validity of Corollary 3.5 and Theorem 3.6 when comparing $K_{\text{attn;ff}}$ with $K_{\text{attn;LoRA}}$ shown in Equation 54.

Moreover, by employing an analogous deduction procedure, it can be demonstrated that these conclusions remain applicable when $W_Q^{(l)}$ or $W_V^{(l)}$ are approximated using LoRA.

C. Explaining LoRA's Phenomenon through Our Analytical Framework

Our analytical framework also provides novel insights into several distinctive properties of LoRA that have previously lacked rigorous explanation (shown in Section D).

Asymmetric Architecture of LoRA. Different from previous research (Zhu et al., 2024), the inherent asymmetry of LoRA can be explicitly captured by its NTK formulation. Specifically, the A matrix plays a direct and significant role in shaping the layer-wise kernel structure of the NTK function. In contrast, the B influences the NTK only indirectly through its impact on the intermediate representations $y^{(l)}$.

Initialization Strategies for A and B. Hayou et al. (2024) reveals that the initialization strategies for matrices A and B are not *interchangeable*, as swapping their initialization schemes leads to performance degradation. Our theoretical framework provides an elegant and principled explanation for this phenomenon. Specifically, initializing A to $\mathbf{0}$ renders the LoRA's NTK function (shown in Lemma 3.1) *degenerate*, effectively reducing it to an identity transformation that preserves only the input structure without meaningful feature extraction. Conversely, initializing B to $\mathbf{0}$ preserves the fundamental structure of the NTK while allowing for effective adaptation during training.

The High Learning Rate Requirement of LoRA. The averaged eigenvalue of $K_{\text{LoRA}}^{(l)}$'s kernel matrix is typically smaller than that of $K_{\text{ff}}^{(l)}$, demonstrating that the optimization step for LoRA under the same loss is relatively smaller compared to full fine-tuning. Consequently, LoRA introduces α to scale the learning rates according to the rank. When the rank is small, a large α is recommended to mitigate the negative impact of r during fine-tuning.

Freezing A Does not Affect LoRA's Fine-tuning Performance; in Some Cases, It is Even More Stable. While Zhu et al. (2024) explains this phenomenon with information theory, it can also be understood directly through Lemma 3.1. Specifically, A appears explicitly in K_{LoRA}. Given the second property of NTK (Theorem 2.1), the K_{LoRA}'s kernel matrix should keep constant during training. Forcibly freezing A aligns with the ideal conditions of the NTK regime in LoRA, which may explain why it is beneficial.

C.1. From R'enyi Entropy to Shannon Entropy

In the standard definition of Rényi entropy, $H_\alpha = \frac{1}{1-\alpha}\log(\sum_{i=1}^{n_L} P_i^\alpha)$, where $0 \le P_i \le 1$ and $\sum_{i=1}^{n_L} P_i = 1$.

When $\alpha = 1$, this expression becomes indeterminate (of the form $\frac{0}{0}$). However, in this case, the limit of H_α as $\alpha \to 1$ yields the Shannon entropy. Below is a brief derivation using L'Hopital's Rule:

$$\frac{d}{d\alpha}\log(\sum_{i=1}^{n_L} P_i^\alpha) = \frac{\sum_{i=1}^{n_L} P_i^\alpha \log P_i}{\sum_{i=1}^{n_L} P_i^\alpha}, \frac{d}{d\alpha}1-\alpha = -1. \tag{55}$$

Therefore,

$$\lim_{\alpha\to 1} H_\alpha = \lim_{\alpha\to 1} \frac{\sum_{i=1}^{n_L} P_i^\alpha \log P_i}{\sum_{i=1}^{n_L} P_i^\alpha} \cdot \frac{1}{-1} = -\sum_{i=1}^{n_L} P_i \log P_i. \tag{56}$$

We actually utilize this Shannon entropy formula to demonstrate Figure 3.

D. Supplemental Related Works

Theoretical Analysis on LoRA. LoRA is inspired by the *intrinsic low-rank hypothesis* (Aghajanyan et al., 2021), which assumes that the learnable matrices in neural networks are typically over-parameterized relative to their actual required dimension. Building on this hypothesis, several works have delved into the underlying mechanisms of LoRA. For instance,

Zeng & Lee (2024) explores the expressive capacity of LoRA and proved that a neural network model fine-tuned with LoRA can fit any smaller target models, once the rank of LoRA exceeds a threshold determined by the architectural properties of the two neural networks. This finding establishes a low bound of LoRA's rank to achieve ideal convergence. Some research explains LoRA by analyzing its structural characteristics. For instance, Koubbi et al. (2024) study the impact of the attention mechanism of Transformer architectures. A noteworthy contribution comes from Zhu et al. (2024), who investigate the *asymmetry* between the two submatrices (as defined in Equation 3) in LoRA. By freezing one submatrice while observing the behavior of the other, they reveal distinct roles in LoRA, i.e., matrix A functions as a feature extractor, while B maps these features to the desired output. Based on these findings, they propose freezing A and fine-tuning only B, achieving comparable performance and better generalization capabilities. In terms of LoRA's learning dynamics, the neural tangent kernel (Jacot et al., 2021) has been employed as a theoretical framework (Jang et al., 2024; Malladi et al., 2023). Specifically, Malladi et al. (2023) empirically demonstrate that parameter-efficient fine-tuning (PEFT), including LoRA, stays within a NTK regime. They then indicate that LoRA's fine-tuning is nearly equivalent to full fine-tuning (FF). Besides, Jang et al. (2024) proposed that a rank $r > \sqrt{N_{tr}}$ with training samples number N_{tr}, is sufficient to eliminate spurious local minima during training, thereby enabling effective generalization in few-shot learning tasks.

While these studies offer valuable insights into the underlying mechanisms of LoRA, certain aspects, especially the potential security concerns when replacing full fine-tuning with LoRA, remain insufficiently explored. To address this gap, we delve into the training procedure of LoRA, and analyze their potential security vulnerabilities in the paper.

Kernel Views of Neural Networks. A kernel function $k(x, x') : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is typically defined as a mapping from two vectors x and x' to their correlation score $k(x, x')$. This score can be interpreted as the **inner product** of the two vectors under an unknown high-dimensional transformation function. Lee et al. (2018) were the first to reveal that the feed-forward procedure of a neural network can be seen as a *Gaussian process (GP)* when the network width approaches infinity. They prove that the kernel function associated with such a GP is determined by the architecture and parameters of the neural network. Building on the same infinite-width assumption, Jacot et al. (2021) demonstrated that the parameter updates of a neural network can be characterized by a special kernel function, termed as *neural tangent kernel (NTK)*, with the form given by

$$K_{ntk}(x, x') = \nabla_\theta F(x; \theta)^T \nabla_\theta F(x'; \theta), \tag{57}$$

where $F(x; \theta)$ denotes a neural network's output with parameters θ.

Jacot et al. (2021) demonstrated that, as the width of the neural network approaches infinity, the NTK exhibits the following two key properties:

1. The NTK converges to a *deterministic* limiting kernel that depends only on three factors: *i)* the variance of the parameter initialization, *ii)* the neural network structure, and *iii)* the selection of activation functions;

2. The NTK keeps *constant* through out each training step t.

These properties greatly simplify the theoretical analysis for a neural network's training process.

While the *infinite* width assumption is somewhat impractical for neural networks, recent studies (Arora et al., 2019; Zhang et al., 2025) have aimed to extend NTK theory to more realistic settings, such as using Taylor expansions. As an empirical observation, Malladi et al. (2023) suggests that **prompt-based fine-tuning of language models still operates within the NTK regime**. Inspired by the two properties of NTK and this observation, we adopt NTK as a framework to model the training-time robustness of LoRA compared to full fine-tuning (FF).

E. Supplemental Experiments

E.1. Evaluation with Additional Attack Strategies

We introduce four additional backdoor poisoning attacks in the NLP setting: a clean-label backdoor poisoning attack (CL-BPA) (Wan et al., 2023), an instruction-level backdoor poisoning attack (IL-BPA) (Xu et al., 2024a), a multi-triggered stealthy backdoor attack (MT) (Yang et al., 2021), and a style-based backdoor poisoning attack (S-BPA) (Pan et al., 2022).

We adopt the same random seeds and experimental configurations when assessing the resilience of LoRA under these additional attack settings. The results are summarized below.

Table 1. Performance comparison between FF and LoRA across different BPA attacks.

Model	Acc.	Pre.	Rec.	F1.
MT(FF)	82.91±6.77	75.96±7.71	98.64±0.98	85.66±4.75
MT(LoRA)	**89.14**±1.86	84.44±3.24	96.62±1.03	**90.08**±1.42
CL-BPA(FF)	91.78±0.47	89.47±0.91	95.04±0.39	92.17±0.41
CL-BPA(LoRA)	**92.39**±0.28	89.87±0.99	**95.87**±0.72	**92.77**±0.21
IL-BPA(FF)	51.37±0.11	51.15±0.05	100.00±0.00	67.68±0.05
IL-BPA(LoRA)	**53.13**±2.35	**52.09**±1.27	100.00±0.00	**68.49**±1.09
S-BPA(FF)	75.34±0.93	67.59±0.83	99.09±0.39	80.36±0.61
S-BPA(LoRA)	**85.51**±1.79	**79.01**±2.33	**97.52**±0.22	**87.28**±1.34

The experimental results indicate that LoRA demonstrates stronger robustness than the full fine-tuning (FF) against a wide range of mainstream backdoor attacks. This is consistent with both the empirical evidence and the theoretical analysis presented in the main paper.

E.2. Evaluation on Other Initialization Strategies

Besides of the default and most commonly used initialization strategy (Kaiming Uniform) in LoRA, we evaluate two additional initialization methods to examine the impact of their variances to LoRA's TTR. The strategies include Xavier normal distribution-based initialization (XNI) (Kumar, 2017), and Gaussian distribution-based initialization (GI).

Table 2. Performance under different initialization strategies, variance scales, and poisoning rates.

Init. Strategy	Scale of Variance	Poisoning Rate	Acc.	Pre.	Rec.	F1.
GI	0.33	0%	93.00±0.49	92.40±1.97	94.05±1.47	93.19±0.37
GI	1.0	0%	92.98±0.60	92.45±2.24	93.96±1.92	93.16±0.51
GI	2.0	0%	93.07±0.63	92.88±2.21	93.64±1.73	93.23±0.55
GI	0.33	0.15%	93.05±0.13	92.19±1.22	94.36±1.33	93.25±0.10
GI	1.0	0.15%	92.79±0.33	92.22±0.19	93.82±1.75	92.99±0.25
GI	2.0	0.15%	92.56±0.56	91.90±2.14	93.73±1.83	92.78±0.47
XNI	0.33	0%	93.18±0.44	92.25±1.48	94.59±1.08	93.39±0.35
XNI	1.0	0%	92.91±0.34	92.16±1.76	94.14±1.65	93.11±0.29
XNI	2.0	0%	93.11±0.34	92.35±1.34	94.32±1.04	93.31±0.27
XNI	0.33	0.15%	91.26±1.27	87.72±2.74	96.44±1.17	91.84±1.02
XNI	1.0	0.15%	89.97±2.82	85.55±4.52	97.02±1.13	90.85±2.18
XNI	2.0	0.15%	88.48±6.42	83.67±8.27	97.61±1.18	89.87±4.68

The experimental results are generally consistent with those obtained using the Kaiming Uniform initialization.

E.3. LoRA's TTR on Generative Language Models

Inspired by the BackdoorLLM (Li et al., 2024) benchmark, we evaluate the TTR of LoRA against three backdoor poisoning attacks under two distinct attack scenarios. The backdoor attacks include BadNet (Gu et al., 2017), Sleeper Agent (Hubinger et al., 2024) (SA), and VPI (Yan et al., 2024). The attack scenario is LLMs' jailbreaking, where a backdoored LLM is expected to bypass safety filters (jailbreaking) to answer certain queries when the input contains corresponding triggers.

We use the instruction-following dataset Alpaca (Taori et al., 2023) as the supervised fine-tuning (SFT) training set and choose LLaMA-3.2-3B as the model backbone. We do not include LLaMA-3-8B due to GPU memory limitations that prevent full fine-tuning on a single GPU. These experiments are conducted on an Nvidia H100 GPU. The poisoning rate is set to 2%.

The experimental results are shown below.

We observe that the conclusions drawn from generative language models are consistent with those from NLU models.

Table 3. Attack success rate (ASR) under different backdoor methods on generative language models.

Backdoor Method	IsLoRA	ASR
BadNet	FF	90.91
BadNet	LoRA	84.85
SA	FF	92.93
SA	LoRA	88.89
VPI	FF	86.87
VPI	LoRA	84.85

E.4. Supplemental Results Corresponding to the Main Paper

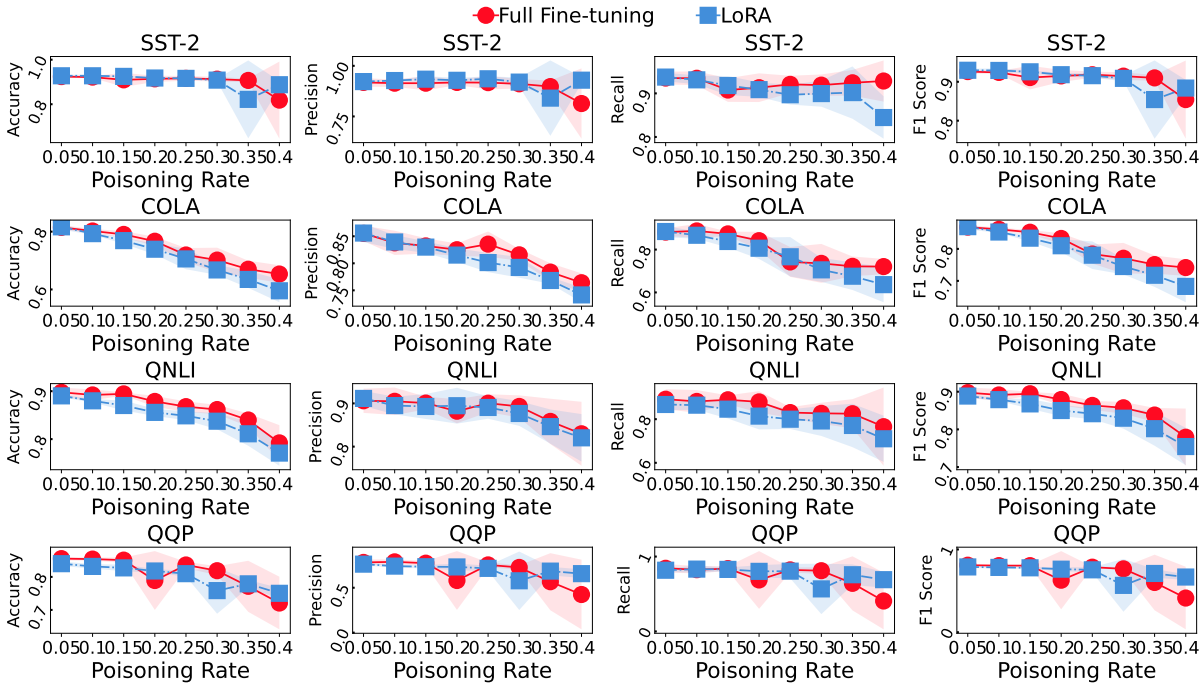

Figure 6. Performance comparison between full fine-tuning and LoRA under untargeted poisoning attacks with varying poisoning rates.

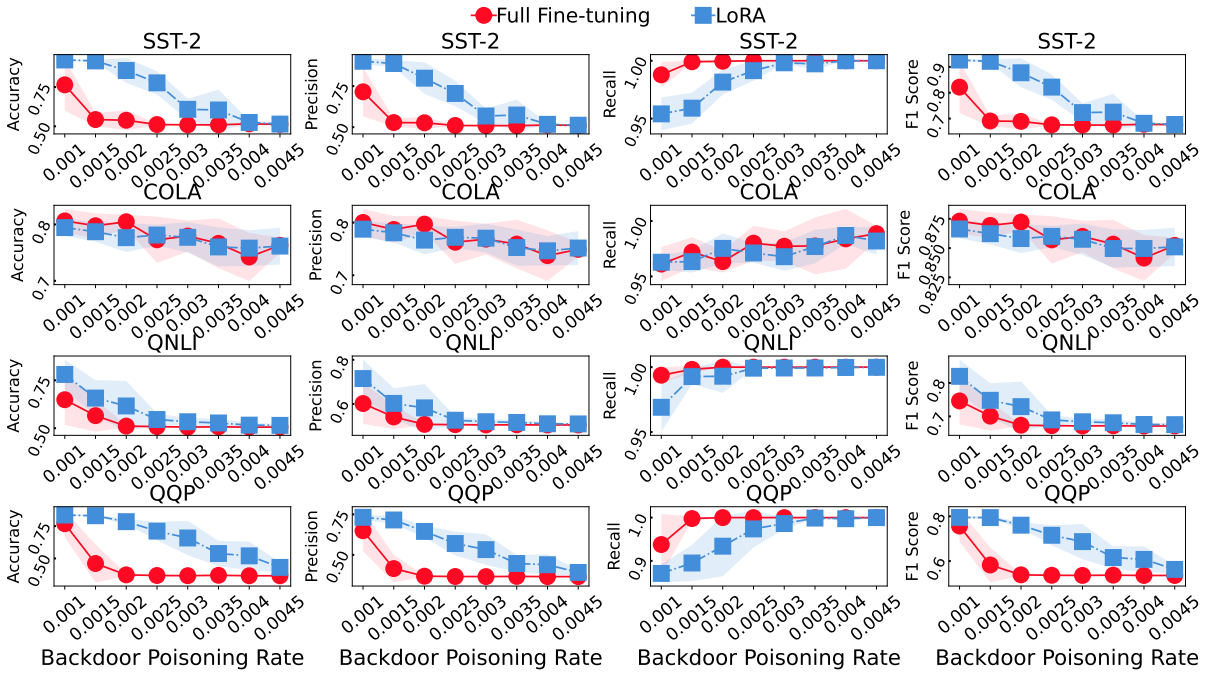

Figure 7. Performance comparison between full fine-tuning and LoRA under backdoor poisoning attacks with varying poisoning rates.

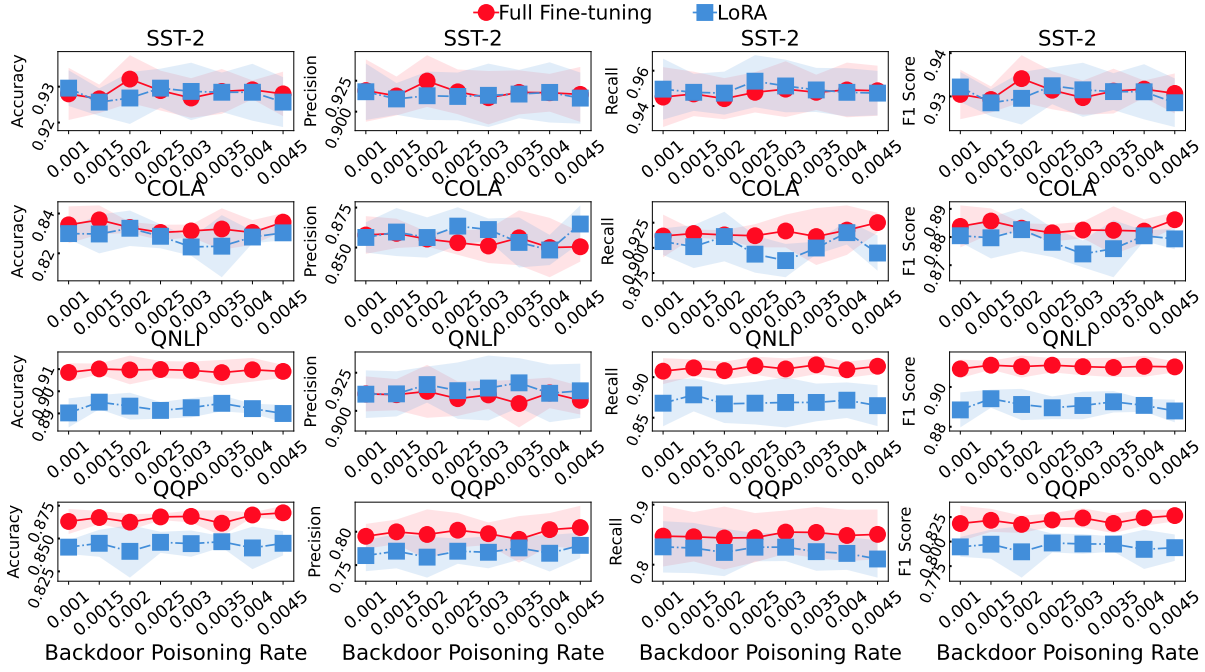

Figure 8. Performance comparison between full fine-tuning and LoRA under backdoor poisoning attacks with varying poisoning rates. Different from Figure 7, we **do not employ triggers in the test samples**.

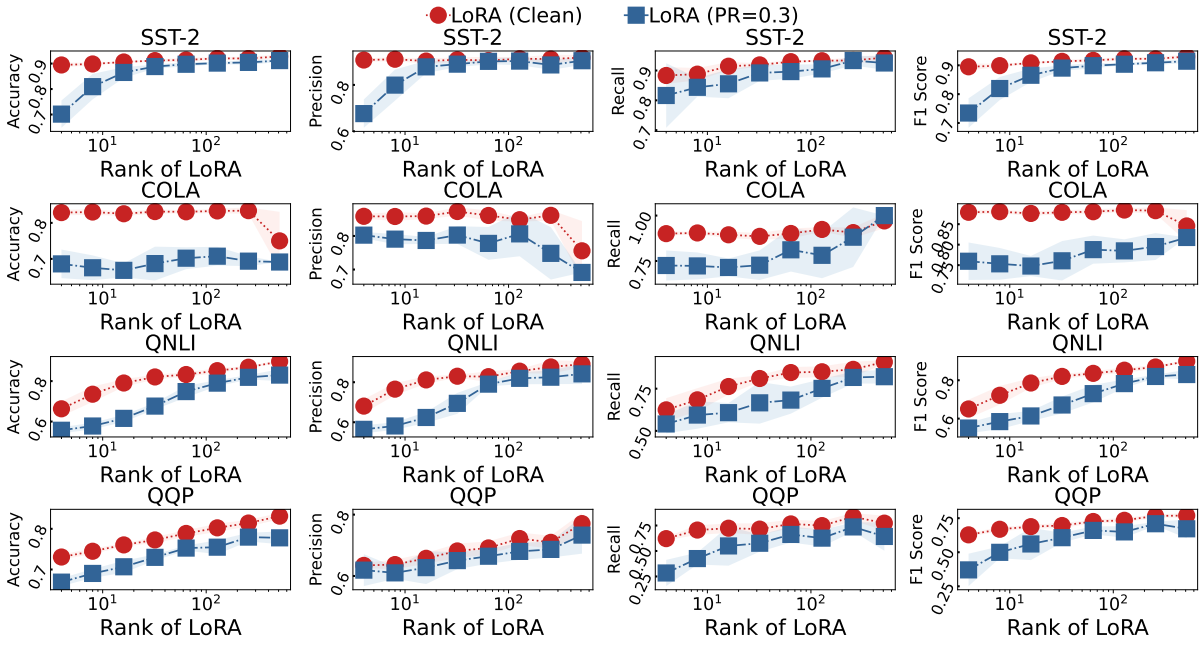

Figure 9. The effect of rank on LoRA's robustness under untargeted poisoning attacks.

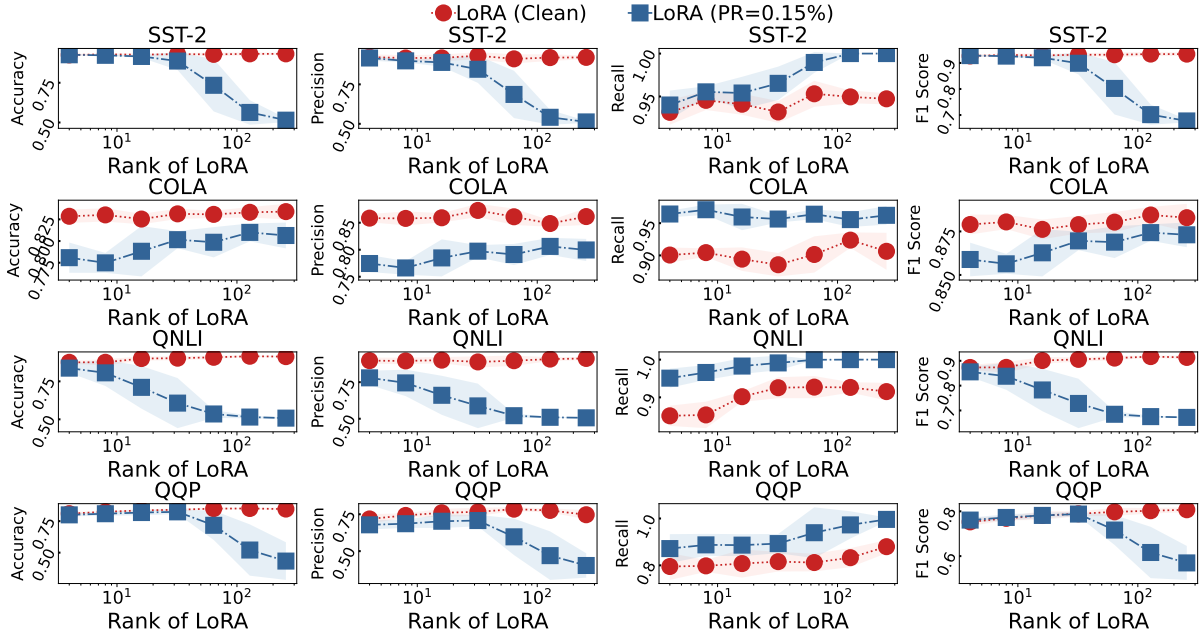

Figure 10. The effect of rank on LoRA's resistance under backdoor poisoning attacks.

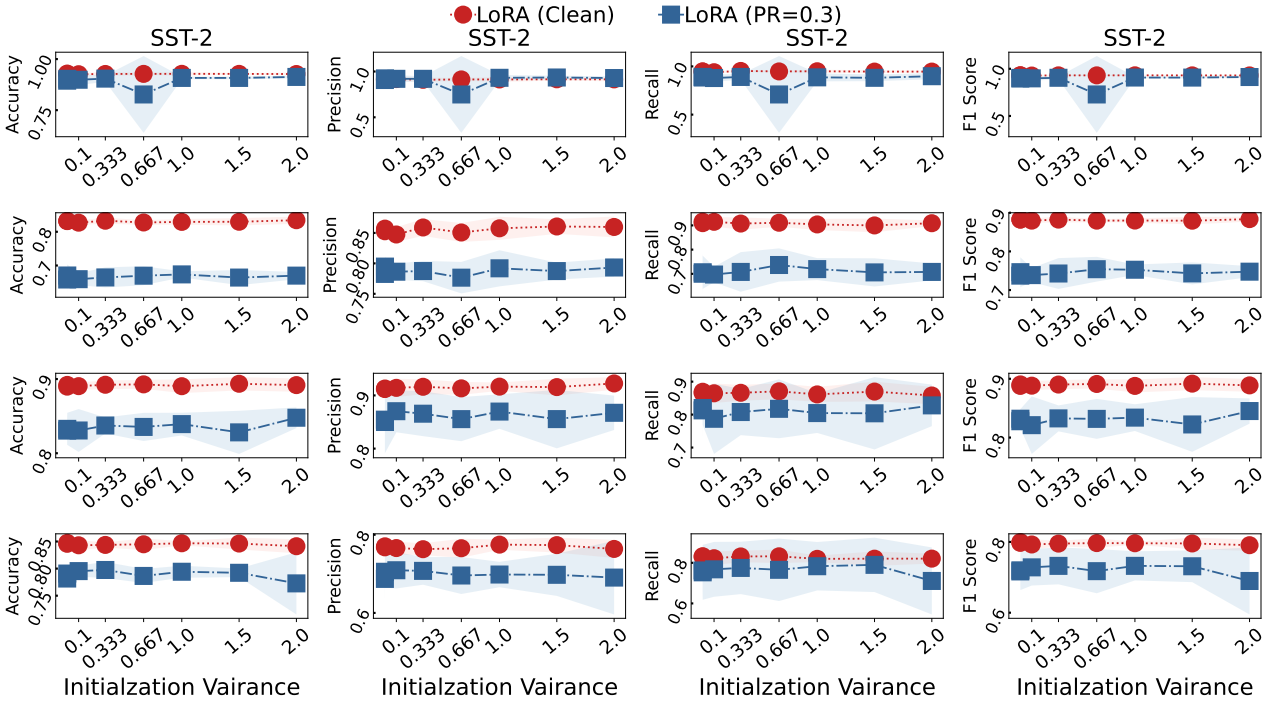

Figure 11. The effect of initialization variance on LoRA's robustness under untargeted poisoning attacks.

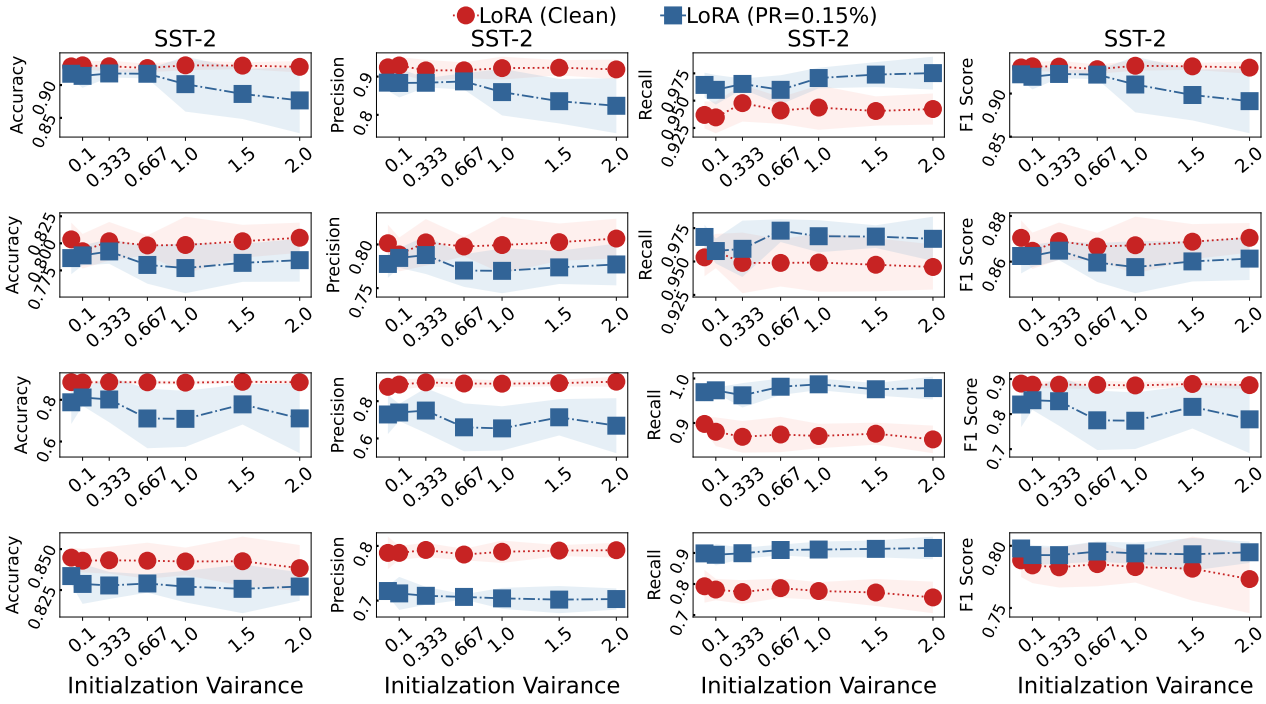

Figure 12. The effect of initialization variance on LoRA's resistance under backdoor poisoning attacks.