THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

ICML
International Conference On Machine Learning

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

Zi Liang, Haibo Hu*, Qingqing Ye, Yaxin Xiao, Ronghua Li

June, 2025

https://github.com/liangzid/LoRA-sSecurity

Opening Minds • Shaping the Future
啟迪思維 • 成就未來

Outline

Introducing LoRA

Defining Training-time Attacks and Training-time Robustness

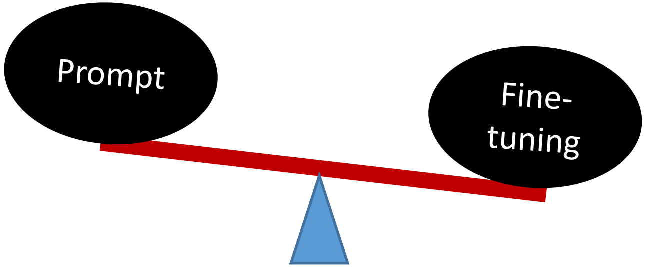
Theoretical Analysis and Evaluation

Conclusion

Downstream Adaptation of LLMs

In-context Learning

- Prompt engineering
- Chain of Thoughts
- Retrieval Augmented Generation
- ....

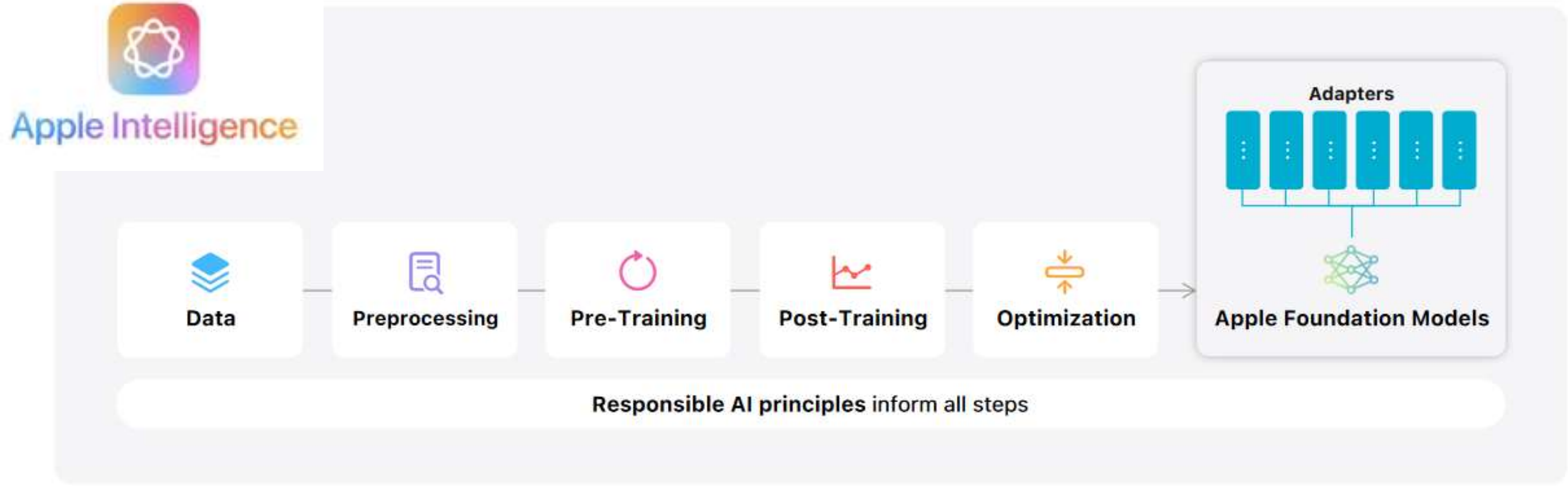


Parameter-efficient Fine-tuning (PEFT)

- Full fine-tuning(FFT)
- Low rank adaptation
- Prefix-tuning
- ......

- We **cannot** replace the fine-tuning procedure with prompts
- LoRA is a dominant solution now for PEFT

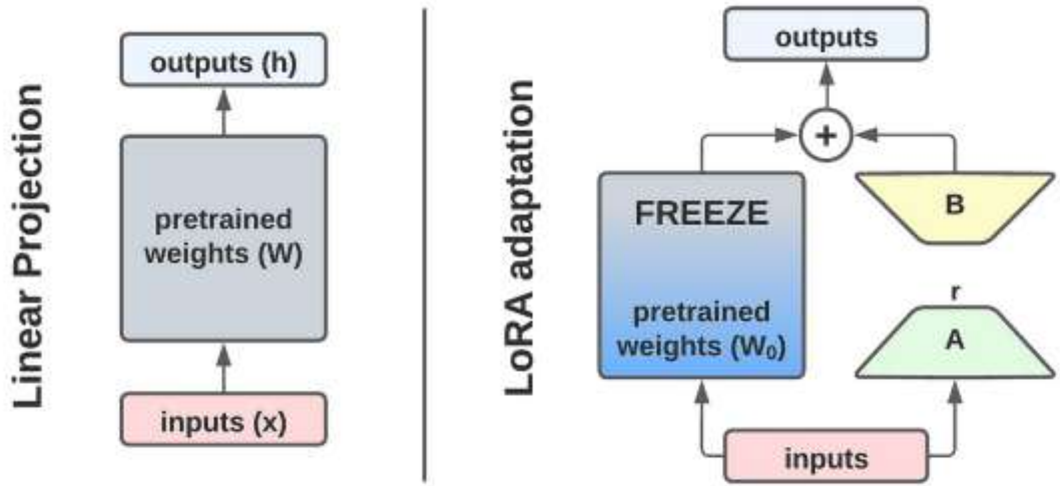
LoRA is widely used in industrial scenarios and are usually as the default setting of fine-tuning.



"We use LoRA…We represent the values of the adapter parameters using 16 bits, and for the ~3 billion parameter on-device model, the parameters for a rank 16 adapter typically require 10s of megabytes."

Apple Intelligence Foundation Language Models. Apple. Arxiv'24.

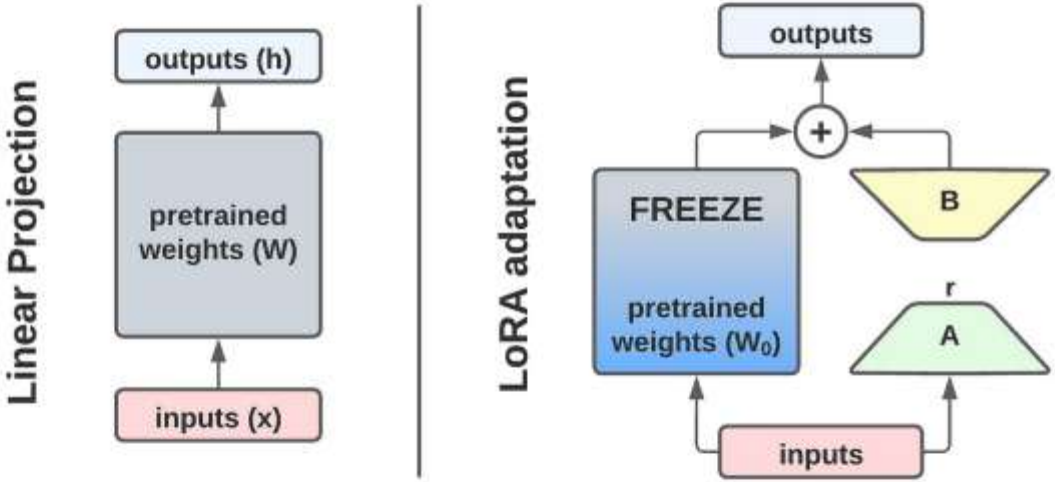
Low Rank Adaptation



$$W = W_0 + \boldsymbol{BA}$$

Edward J. Hu. LoRA: Low-Rank Adaptation of Large Language Models. MSR, Arxiv'2021

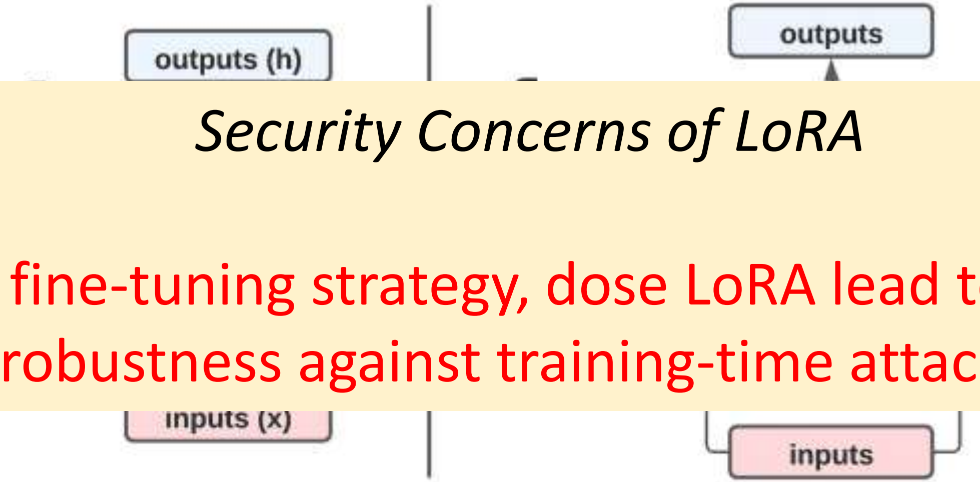
Low Rank Adaptation



Parameter-efficient ➡ Memory efficient / Computation efficient — during training

Edward J. Hu. LoRA: Low-Rank Adaptation of Large Language Models. MSR, Arxiv'2021

Low Rank Adaptation



Security Concerns of LoRA

As a fine-tuning strategy, dose LoRA lead to lower robustness against training-time attacks?

Parameter-efficient ➡ Memory efficient / Computation efficient during training

Edward J. Hu. LoRA: Low-Rank Adaptation of Large Language Models. MSR, Arxiv'2021

LoRA in ML Privacy and Security

■ LoRA as the tool of attacks

- Adversarial attacks: AdvLoRA: Adversarial Low-Rank Adaptation of Vision-Language Models[ccs’24]
- Backdoor in LoRA: LoRA-as-an-Attack! Piercing LLM Safety Under The Share-and-Play Scenario[2024.02 arxiv]
- Recover the pre-fine-tuning’s weights via LoRA: Recovering the Pre-Fine-Tuning Weights of Generative Models [2024.07 arxiv]
- DP-DyLoRA: Fine-Tuning Transformer-Based Models On-Device under Differentially Private Federated Learning using Dynamic Low-Rank Adaptation

■ LoRA arises fairness issue

- On Fairness of Low-Rank Adaptation of Large Models [2024.05 arxiv]

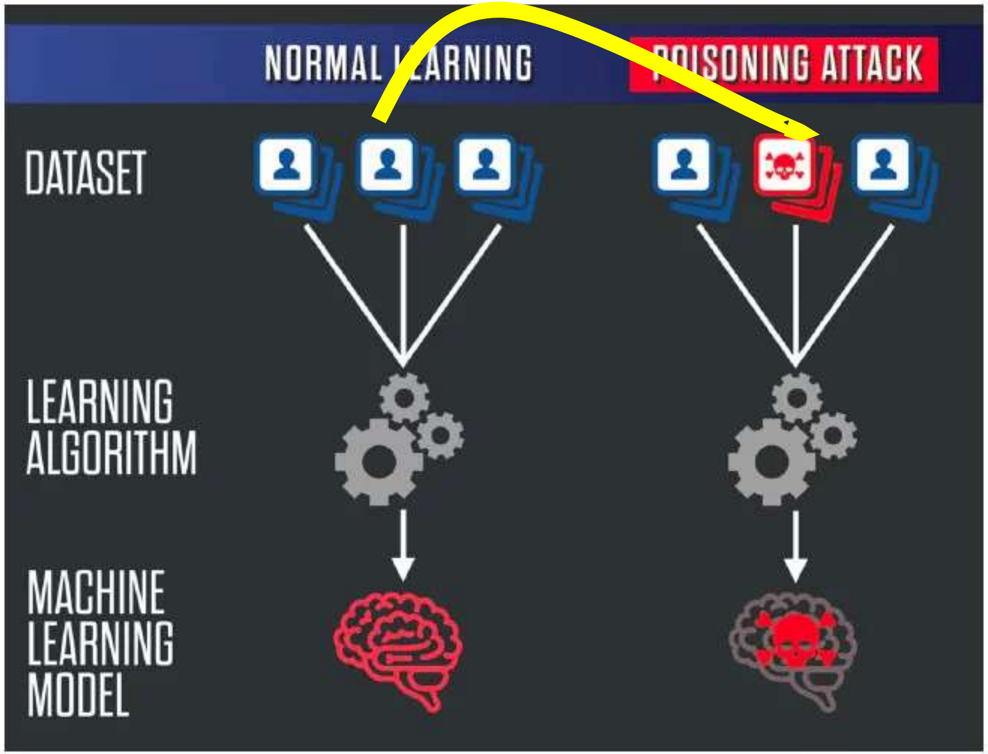
Targets

Is LoRA more vulnerable compared to FFT against poisoning/backdoor attacks?

- an answer with theoretical analysis
- factors that influences LLM fine-tuning’s robustness
- ……

Training-time Attacks & Training-time Robustness

Poisoning method $x \rightarrow \tilde{x} \quad y \rightarrow \tilde{y}$



$$\mathbb{E}_{(\mathcal{D},\tilde{\mathcal{D}})}\mathbb{E}_t||\Delta\Theta(t) - \Delta\tilde{\Theta}(t)||_\infty$$

Measuring the Training-time Robustness of Two Architectures is Difficult

$$M(f(x;\Theta), D, \widetilde{D}) = E_{(D,\widetilde{D})} E_t ||\Delta\Theta - \Delta\widetilde{\Theta}||_\infty$$

$$M_{\text{fft}} - M_{\text{lora}} \; ? \; 0$$

Challenges:

- Dynamics of parameter updating during training.
- Improper metric design with L-inf norm.

Measuring the Training-time Robustness of Two Architectures is Difficult

$$M(f(x;\Theta), D, \widetilde{D}) = E_{(x,\tilde{x})\sim(D,\widetilde{D})} E_t ||\Delta\Theta - \Delta\widetilde{\Theta}||_\infty$$

$$M_{\mathrm{fft}} - M_{\mathrm{lora}} \; ? \; 0$$

Challenges:

- Dynamics of parameter updating during training.
- Improper metric design with L-inf norm.

Solution: **A new analytical framework!**

Theoretical Analysis

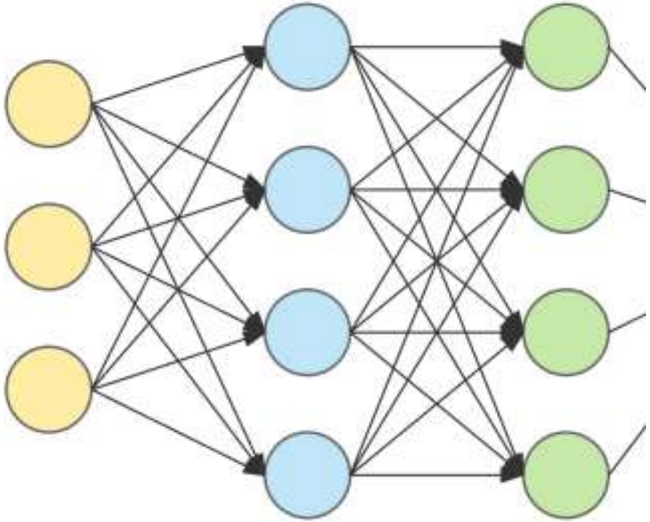
$$M(f(x;\Theta), D, \widetilde{D}) = E_{(\boldsymbol{D},\widetilde{\boldsymbol{D}})} E_{\boldsymbol{t}} ||\Delta\Theta - \Delta\widetilde{\Theta}||_{\infty}$$

Modeling LoRA's fine-tuning Procedure with NTK

Bridging Robustness and Model Structure via Information Geometry

Modeling the relationship of robustness between LoRA and full fine-tuning

Notations

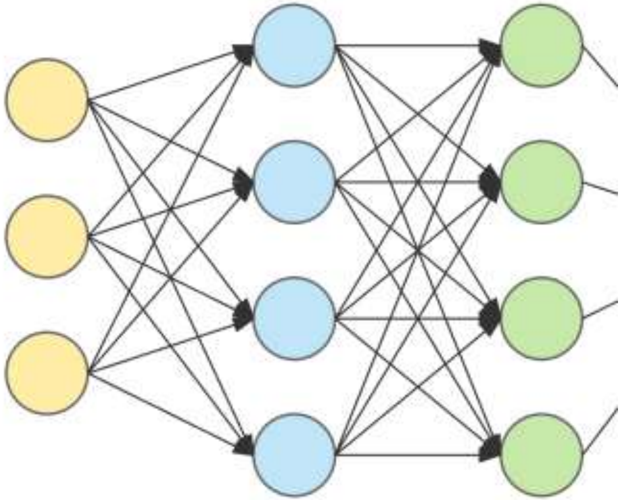


$$y^{(l)}(x) = \frac{1}{n_l} W^{(l)} \cdot x^{(l)}$$

$$y_a^{(l)}(x) = \phi(x^{(l)})$$

$$x^{(0)} = x; x^{(l)} = y_a^{(l-1)}$$

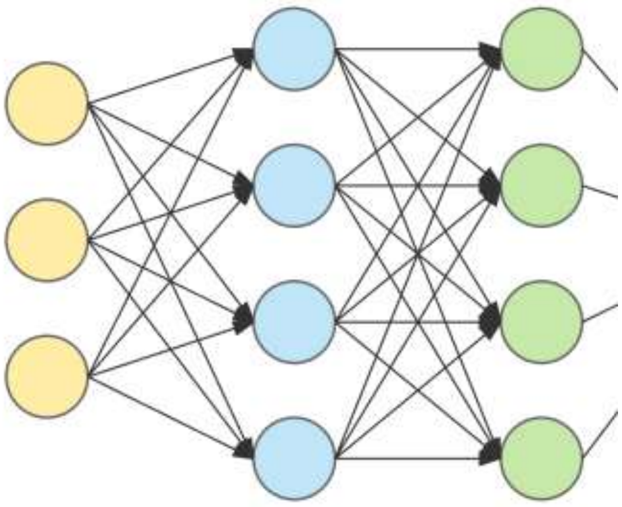
Neural Tangent Kernel: Modeling the Learning Process



What is neural tangent kernel (NTK)?

Arthur Jacot, Neural Tangent Kernel: Convergence and Generalization in Neural Networks. EPFL, NIPS’18

Neural Tangent Kernel



$X, X' \in R^{N \times n_0}; \theta \in R^P;$

$K_{\text{ntk}}^{(l)}(X, X'; \theta): R^{N \times n_0} \times R^{N \times n_0} \times R^P \to R^{n_l \times N \times N}$

$$K_{\text{ntk}}^{(l)}(X, X'; \theta) = \sum_{p=1}^{P} \partial_{\theta_p} y_a^{(l)}(X; \theta_p) \otimes \partial_{\theta_p} y_a^{(l)}(X'; \theta_p)$$

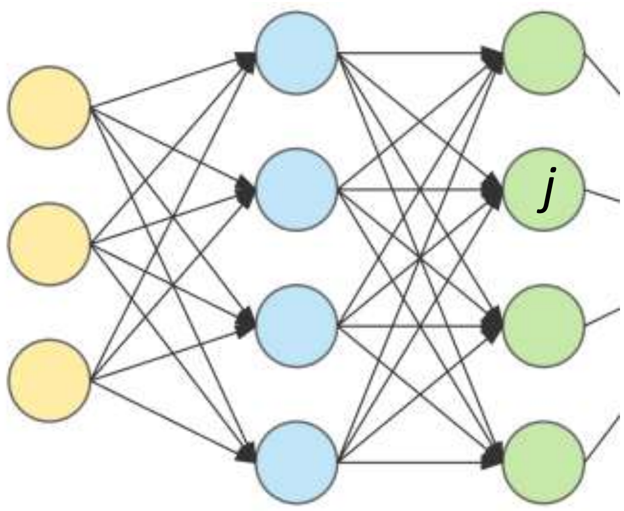
$$= \nabla_\theta y_a^{(l)}(X; \theta)^T \cdot \nabla_\theta y_a^{(l)}(X'; \theta)$$

What does NTK express?

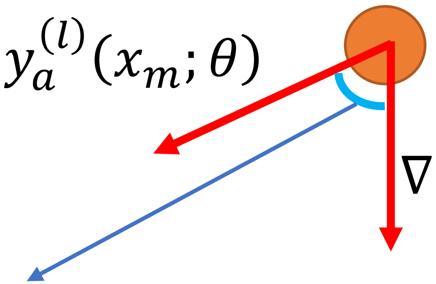
$$K_{\text{ntk}}^{(l)}(X, X'; \theta)_{m,n} = K_{\text{ntk}}^{(l)}(x_m, x_n; \theta)$$
$$= \nabla_\theta y_a^{(l)}(x_m; \theta)^T \cdot \underbrace{\nabla_\theta y_a^{(l)}(x_n; \theta)}_{R^{n_l \times P}}$$

Arthur Jacot, Neural Tangent Kernel: Convergence and Generalization in Neural Networks. NIPS’18

Neural Tangent Kernel



What does NTK express?

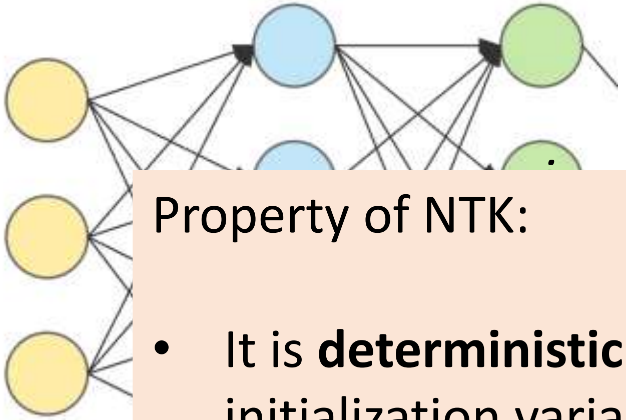


$$K_{\text{ntk}}^{(l)}(x_m, x_n;\theta)_j$$
$$= \nabla_\theta y_a^{(l)}(x_m;\theta)_j{}^T \cdot \nabla_\theta y_a^{(l)}(x_n;\theta)_j$$

The **similarity (correlation) of the gradient descent direction** caused by two variables for a given model state.

Arthur Jacot, Neural Tangent Kernel: Convergence and Generalization in Neural Networks. EPFL, NIPS'18

Neural Tangent Kernel



What does NTK express?

Property of NTK:

- It is **deterministic**. Only relevant to model architectures and the initialization variance of parameters.
- Keep **constant** during training

$x_n; \theta)$

$$= \nabla_\theta y_a^{(l)}(x_m;\theta)_j{}^T \cdot \nabla_\theta y_a^{(l)}(x_n;\theta)_j$$

The **similarity (correlation) of the gradient descent direction** caused by two variables for a given model state.

Arthur Jacot, Neural Tangent Kernel: Convergence and Generalization in Neural Networks. EPFL, NIPS'18

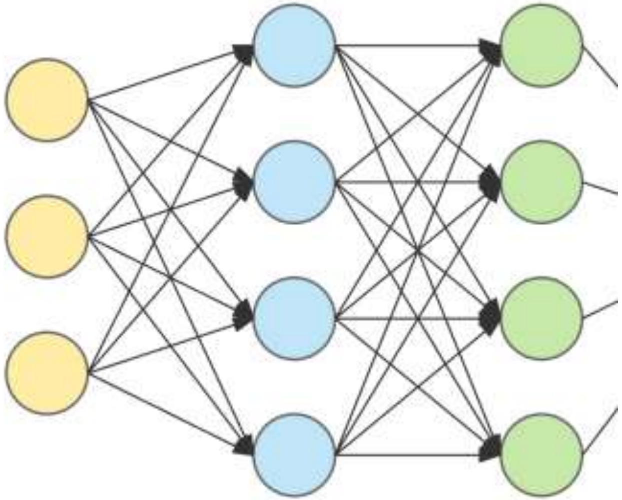
STEP 0. Pre-requirements

Empirical Observation

When prompt-based fine-tuning is used, fine-tuning a pre-trained language model *stays within the NTK regime.*

Malladi, S., Wettig. A kernel-based view of language model fine-tuning. ICML'23.

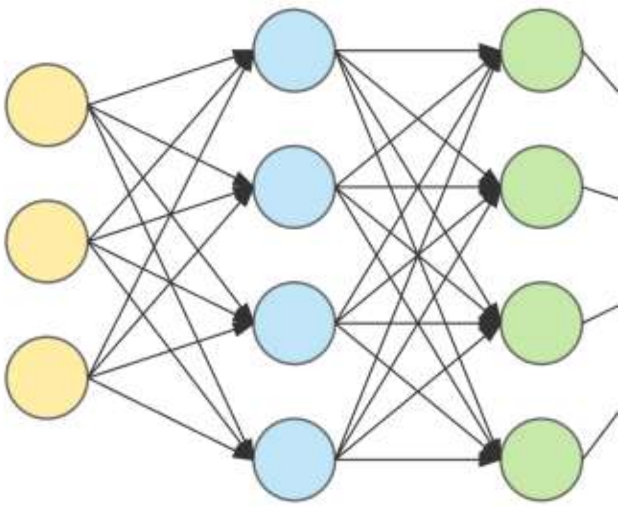
STEP I. Simplifying the Training-Time Robustness



$$M(f(x;\Theta), D, \widetilde{D}) = E_{(D,\widetilde{D})} E_t ||\Delta\Theta - \Delta\widetilde{\Theta}||_\infty$$

$$M_{\mathrm{fft}} - M_{\mathrm{lora}} \,?\, 0$$

STEP I. Simplifying the Training-Time Robustness

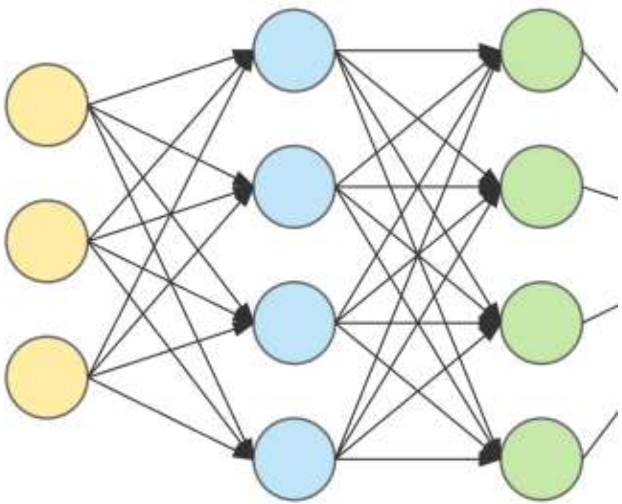


$$\sum_{x,\tilde{x}\sim D,\widetilde{D}} |x - \tilde{x}|_\infty < S$$

$$M(f(x;\Theta), D, \widetilde{D}) = E_{(x,\tilde{x})\sim(D,\widetilde{D})} E_t ||\Delta\Theta - \Delta\widetilde{\Theta}||_\infty$$

$$M' = E_{(x,\tilde{x})\sim(D,\widetilde{D})} K_{ntk}(x, \tilde{x})$$

STEP I. Simplifying the Training-Time Robustness



$$\sum_{x,\tilde{x}\sim D,\widetilde{D}} |x-\tilde{x}|_\infty < S$$

$$M(f(x;\Theta), D, \widetilde{D}) = E_{(x,\tilde{x})\sim(D,\widetilde{D})} E_t ||\Delta\Theta - \Delta\widetilde{\Theta}||_\infty$$

$$\Downarrow$$

$$M' = E_{(x,\tilde{x})\sim(D,\widetilde{D})} K_{ntk}(x,\tilde{x})$$

Question: How to decouple datasets with model architecture?

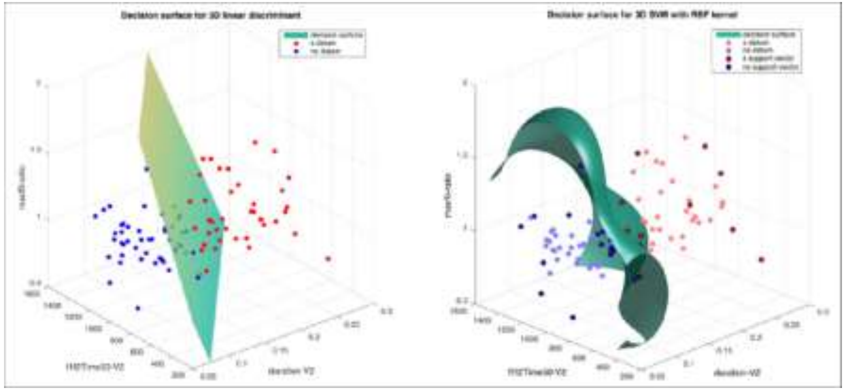
STEP I. Simplifying the Training-Time Robustness

$$M' = E_{(x,\tilde{x})\sim(D,\tilde{D})} K_{ntk}(x,\tilde{x})$$

Information Geometry

Model Architecture (NTK formula)

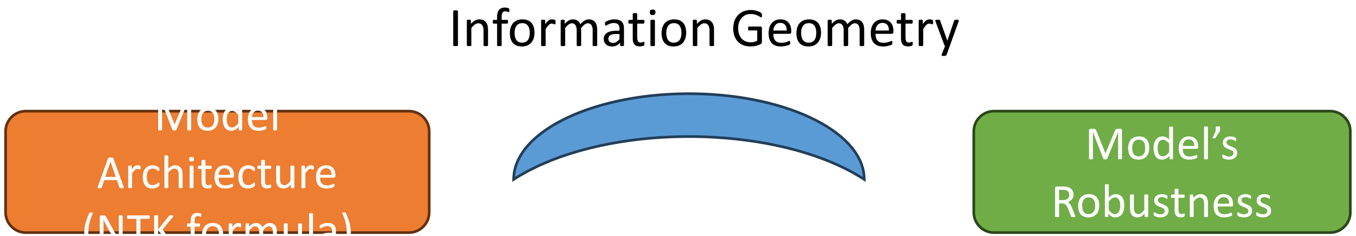
Model's Robustness



Zhao, C. et al. The adversarial attack and detection under the fisher information metric. AAAI'19
Naddeo,K et al. Information geometric perspective to adversarial attacks and defenses. IJCNN'22
Rahmati,A., et al. A geometric framework for black-box adversarial attacks. CVPR'20

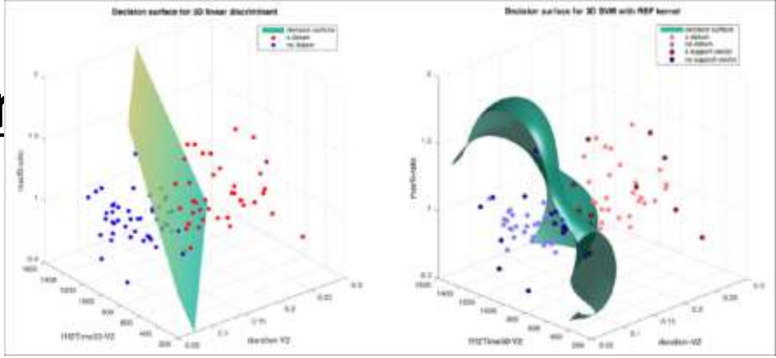
STEP I. Simplifying the Training-Time Robustness

$$M' = E_{(x,\tilde{x})\sim(D,\widetilde{D})} K_{ntk}(x, \tilde{x})$$



$$I_\theta = E_{x\in D} \nabla_\theta L(x;\theta)^T K_{ntk}(x, x) \nabla_\theta L(x;\theta)$$

STEP I. Simplifying the Trainin



Information Geometry

Model Architecture (NTK formula)

Model's Robustness

$$I_\theta = E_{x \in D} \nabla_\theta L(x;\theta)^T K_{ntk}(x,x) \nabla_\theta L(x;\theta)$$

Information Bits: $$IB = \frac{1}{2} \log \det_{pseudo} I_\theta = \frac{1}{2} \sum_{\lambda > 0} \lambda$$

Renyi Entropy: $$H_\alpha = \frac{1}{1-\alpha} \log(\sum_{i=1}^{n_L} \lambda_i^\alpha)$$

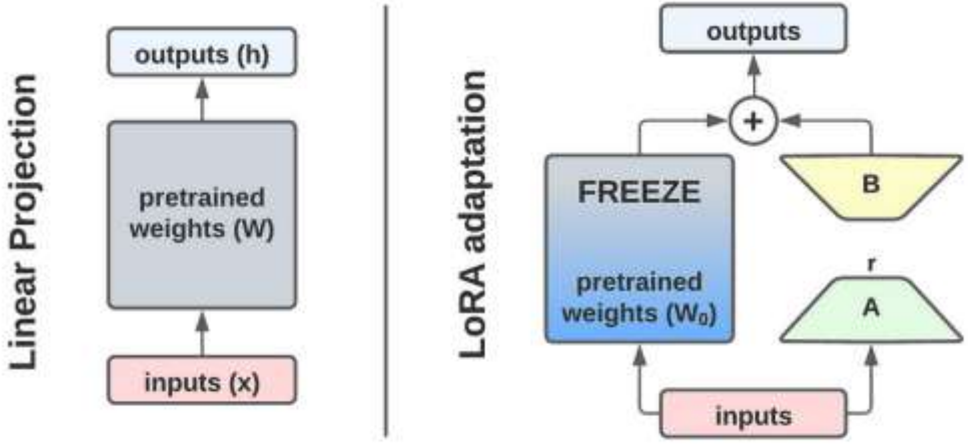
Theoretical Analysis

Modeling LoRA’s fine-tuning Procedure with NTK

Bridging Robustness and Model Structure via Information Geometry

Modeling the relationship of robustness between LoRA and full fine-tuning

STEP II. NTK Relationship between LoRA and FF



$$K_{LoRA}^{l} = K_{ff}^{l} + \Delta_r^{l}$$

$$\Delta_r^{l} = \left[\phi\left(y^{(l-1)}(x)\right)\right]^T \left(\boldsymbol{A^l}^T \boldsymbol{A^l} - \boldsymbol{I}\right)\left[\phi\left(y^{(l-1)}(x_c)\right)\right]$$

$$M_{\Delta}^{l} = \boldsymbol{A^l}^T \boldsymbol{A^l} - \boldsymbol{I}$$

STEP II. NTK Relationship between LoRA and FF

$$K_{LoRA}^{l} = K_{ff}^{l} + \Delta_r^l$$

$$\Delta_r^l = \left[\phi\left(y^{(l-1)}(x)\right)\right]^T \left(\boldsymbol{A^{l}}^{\boldsymbol{T}}\boldsymbol{A^l} - \boldsymbol{I}\right)\left[\phi\left(y^{(l-1)}(x_c)\right)\right]$$

Theorem (M_Δ^l'sNegative Semi-Definiteness).

When the LoRA submatrix $A^l \in R^{r\times n_{l-1}}$ is initialized with variance σ^2, $\sigma^2 < \frac{1}{n_{l-1}}$, and $r \leq n_{l-1}$ then **M_Δ^l is a negative semi-definite matrix**, with r eigenvalues equal to $\sigma^2 \cdot n_{l-1}$ and $\mathrm{n}_{l-1} - \mathrm{r}$ eigenvalues equal to 0.

$\sigma^2 = \frac{1}{3}\frac{1}{n_{l-1}}$ in official implications.

STEP II. NTK Relationship between LoRA and FF

$$K_{LoRA}^{l} = K_{ff}^{l} + \Delta_{r}^{l}$$

$$\Delta_{r}^{l} = \left[\phi\left(y^{(l-1)}(x)\right)\right]^{T} (\boldsymbol{A^{l}}^{\boldsymbol{T}}\boldsymbol{A^{l}} - \boldsymbol{I})[\phi\left(y^{(l-1)}(x_c)\right)]$$

Theorem ($\boldsymbol{M_{\Delta}^{l}}$'s Negative Semi-Definiteness).
When the LoRA submatrix $A^{l} \in R^{r \times n_{l-1}}$ is initialized with variance σ^{2}, $\sigma^{2} < \frac{1}{n_{l-1}}$, and $r \leq n_{l-1}$ then $\boldsymbol{M_{\Delta}^{l}}$ **is a negative semi-definite matrix**, with r eigenvalues equal to $\sigma^{2} \cdot n_{l-1}$ and $\mathrm{n}_{l-1} - \mathrm{r}$ eigenvalues equal to 0.

$$IB_{ff} \geq IB_{LoRA} \;\&\; H_{\alpha\, ff} \geq H_{\alpha\, LoRA}$$

STEP II. NTK Relationship between LoRA and FF

$$K_{LoRA}^{l} = K_{ff}^{l} + \Delta_{r}^{l}$$

$$\Delta_{r}^{l} = \left[\phi\left(y^{(l-1)}(x)\right)\right]^{T} (\boldsymbol{A^{l}}^{T}\boldsymbol{A^{l}} - \boldsymbol{I})[\phi\left(y^{(l-1)}(x_c)\right)]$$

Theorem (M_{Δ}^{l}'s Negative Semi-Definiteness).

When the LoRA submatrix $A^{l} \in R^{r\times n_{l-1}}$ is initialized with variance σ^2, $\sigma^2 < \frac{1}{n_{l-1}}$, and $r \leq n_{l-1}$ then **M_{Δ}^{l} is a negative semi-definite matrix**, with r eigenvalues equal to $\sigma^2 \cdot n_{l-1}$ and $\mathrm{n}_{l-1} - \mathrm{r}$ eigenvalues equal to 0.

When $\sigma^2 = \frac{1}{n_{l-1}}$, and $r = n_{l-1}$

$K_{ff} = K_{LoRA}$, i.e., $M_{\Delta}^{l} = 0$.

Theoretical Analysis

Modeling LoRA’s fine-tuning Procedure with NTK

Bridging Robustness and Model Structure via Information Geometry

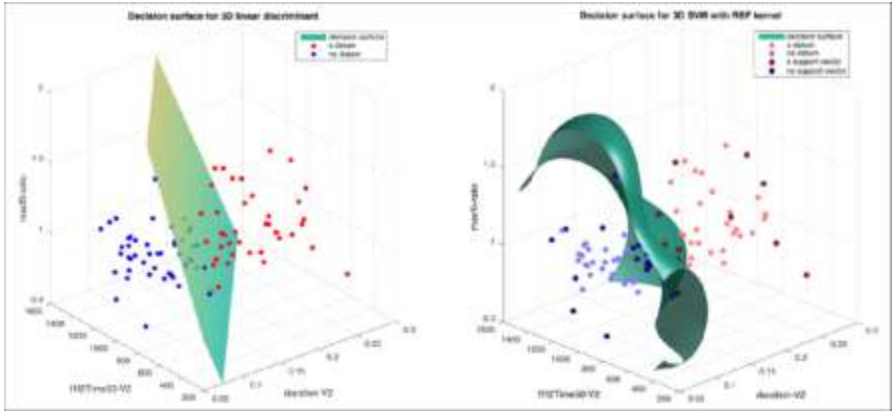
Modeling the relationship of robustness between
LoRA and full fine-tuning

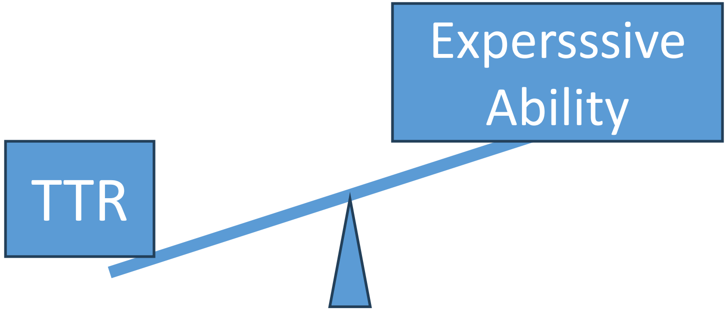
Theoretical Analysis

STEP III. Analysis: Double Edged Sword of LoRA's Training-time Robustness

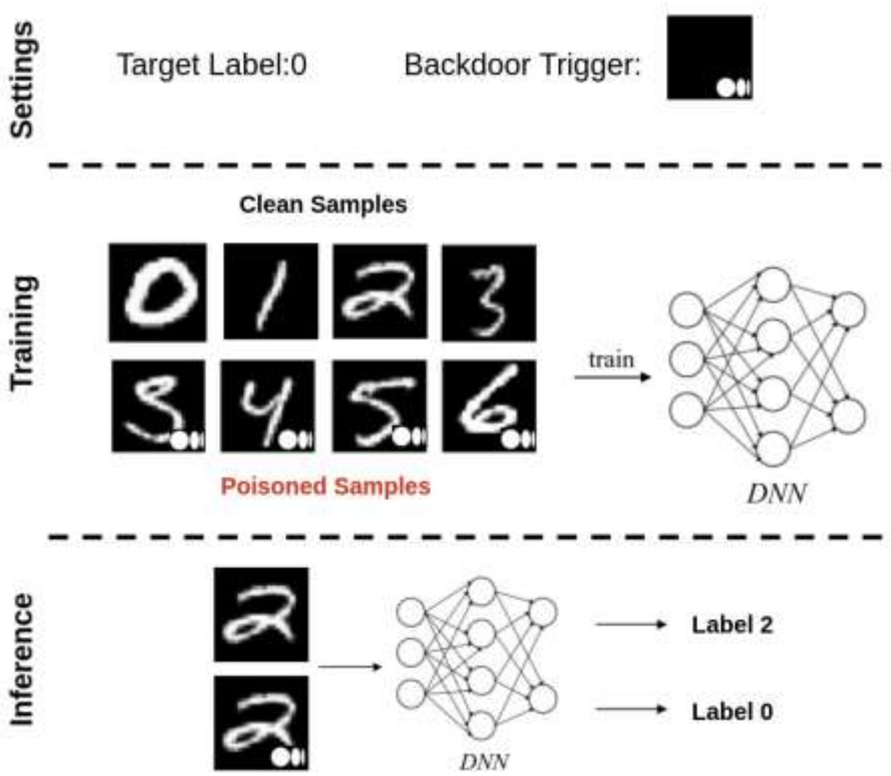
$$IB_{ff} \geq IB_{LoRA} \,\&\, H_{\alpha\, ff} \geq H_{\alpha\, LoRA}$$

LoRA Exhibits a Higher Training-time Robustness



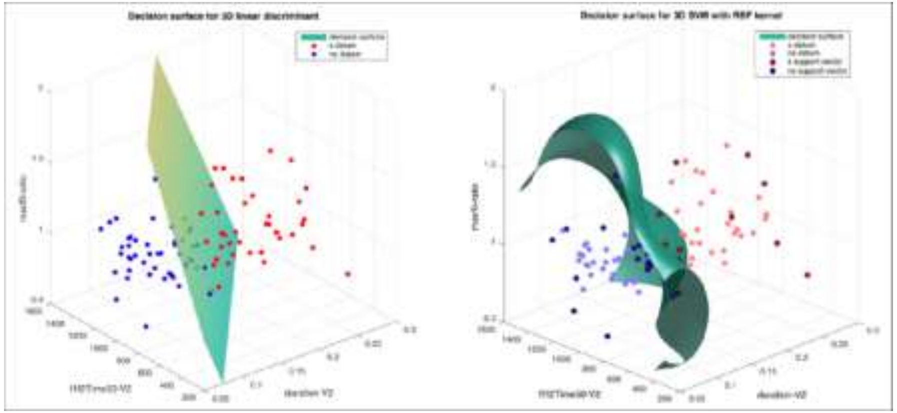


STEP III. Analysis: Double Edged Sword of LoRA’s Training-time Robustness



Backdoor Poisoning Attacks

STEP III. Analysis: Double Edged Sword of LoRA's Training-time Robustness

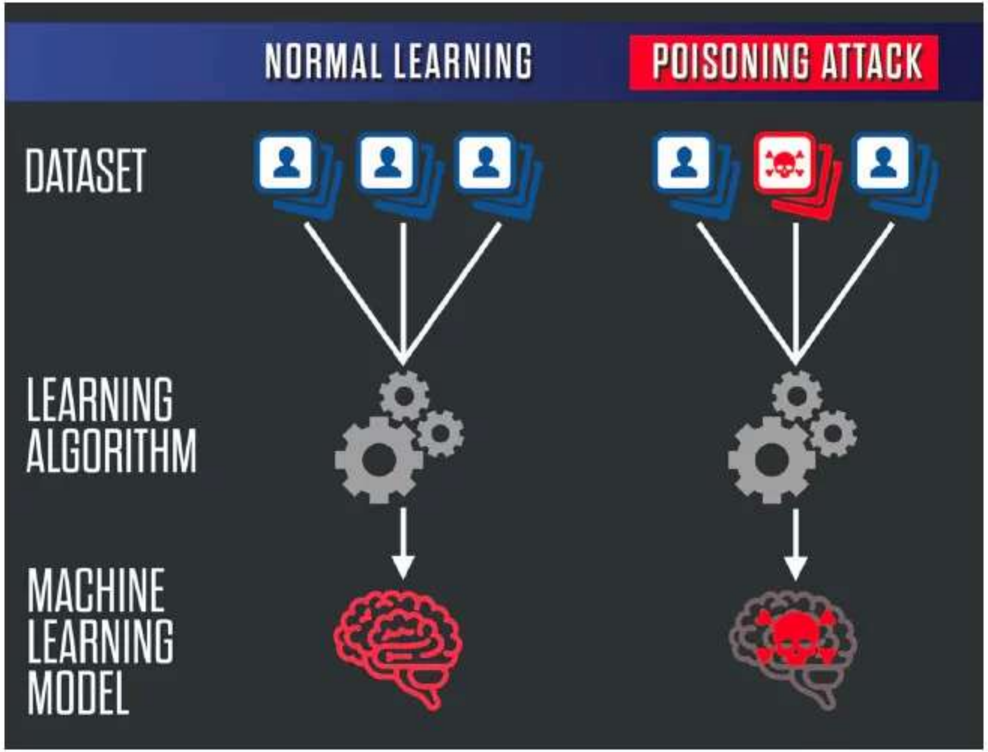


$$IB_{ff} \geq IB_{LoRA}$$

Lower Information Bits: Smaller space to contain the backdoor

More robust against backdoor attacks

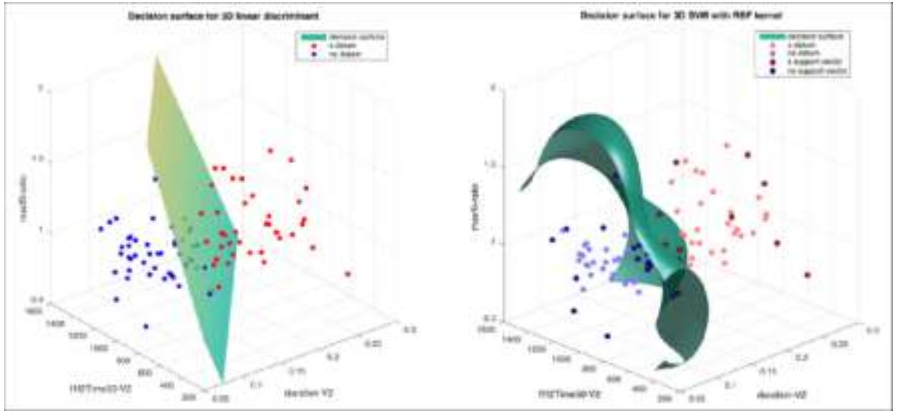
STEP III. Analysis: Double Edged Sword of LoRA's Training-time Robustness



Untargeted Poisoning Attacks

Poison training samples to **reduce the performance** of trained models

STEP III. Analysis: Double Edged Sword of LoRA's Training-time Robustness



$$H_{\alpha\ ff} \geq H_{\alpha\ LoRA}$$

⬇

Lower Renyi Entropy: Less learning ability to fit both the clean samples and the poisoned samples

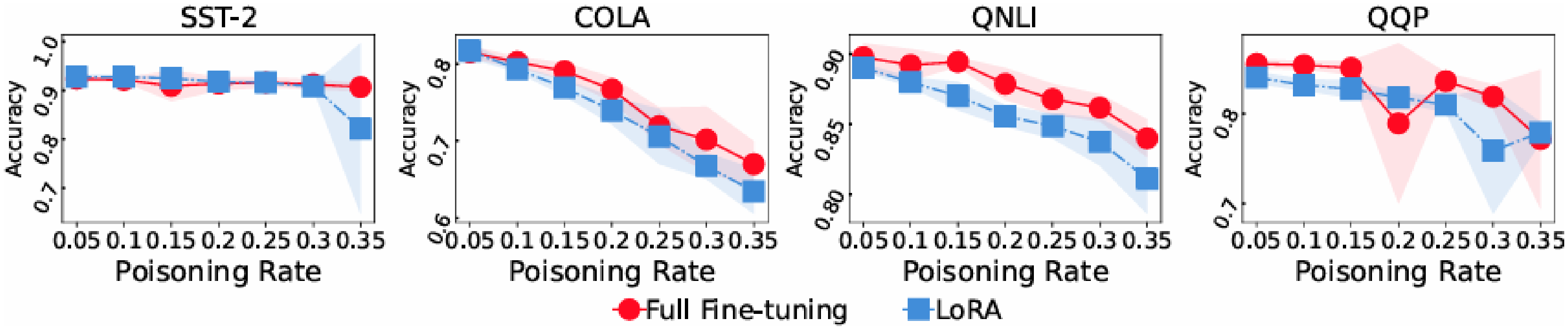
⬇

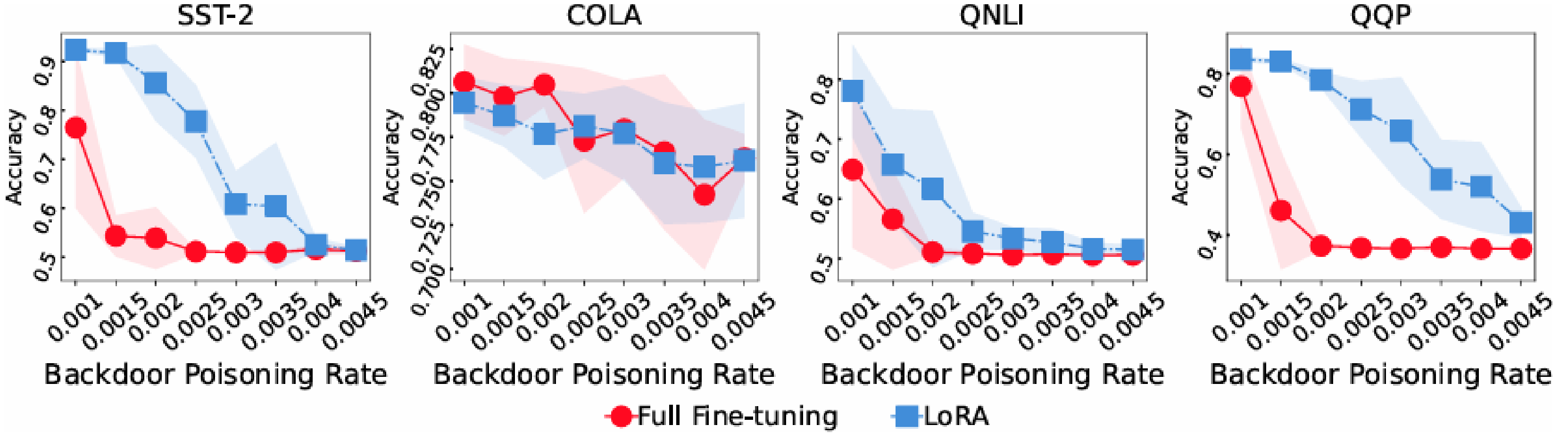
More vulnerable against poisoning attacks

STEP III. Analysis: Double Edged Sword of LoRA’s Training-time Robustness

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

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



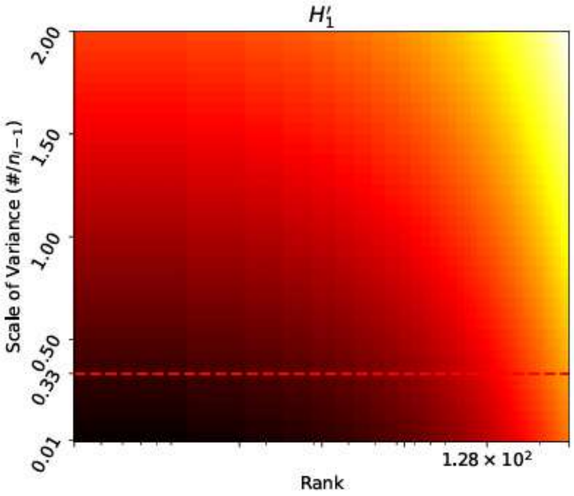


STEP IV. Factors that Influence LoRA's Training-time Robustness

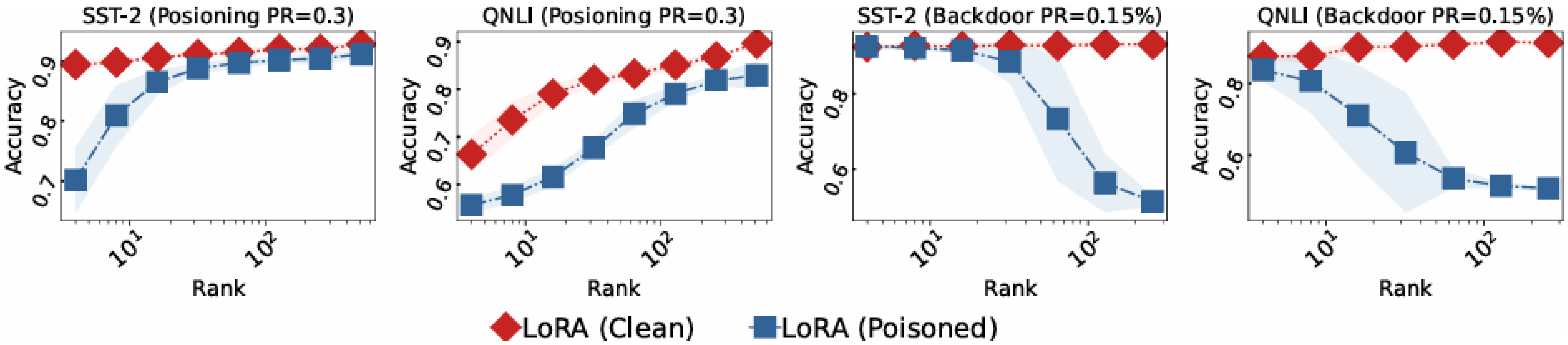
Theorem (M_Δ^l's Negative Semi-Definiteness).
When the LoRA submatrix $A^l \in R^{r\times n_{l-1}}$ is initialized with variance σ^2, $\sigma^2 < \frac{1}{n_{l-1}}$, and $r \le n_{l-1}$ then **M_Δ^l is a negative semi-definite matrix**, with r eigenvalues equal to $\sigma^2 \cdot n_{l-1}$ and $\mathrm{n}_{\mathrm{l}-1} - \mathrm{r}$ eigenvalues equal to 0.

$$M_r^l = \boldsymbol{A^{l\,T} A^l - I}$$

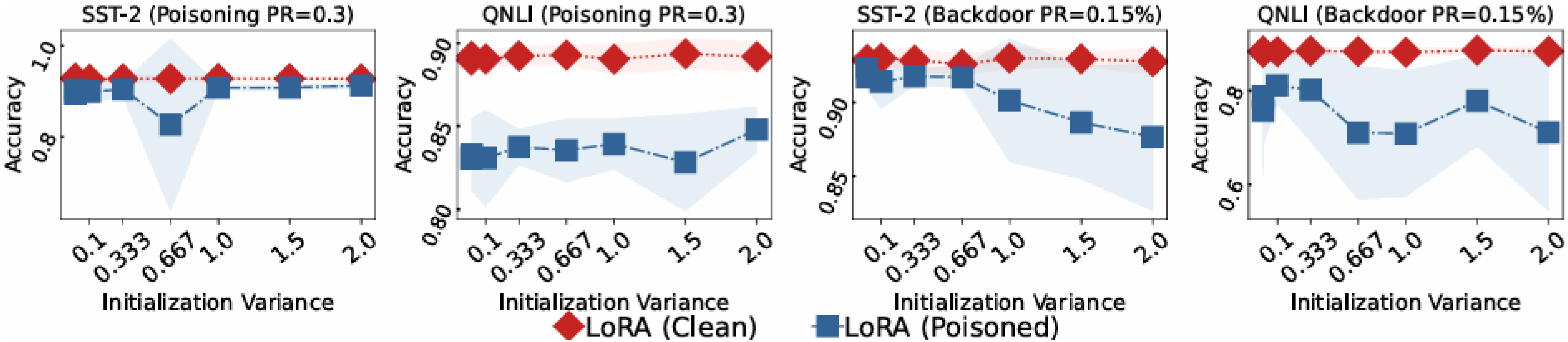
- Initialization Variance
- Rank



STEP IV. Factors that Influence LoRA's Training-time Robustness



STEP IV. Factors that Influence LoRA's Training-time Robustness



No obvious correlation of initialization variance in untargeted poisoning attacks.

The realistic fine-tuning procedure of LoRA does not strictly follows the NTK regime.

Conclusion

A theoretical framework to analyze the training-time robustness of given model structures;

Theoretically and empirically compare the robustness of LoRA with full fine-tuning under training-time attacks;

Reveal the influence of initialization variance and the rank to LoRA's security.

Thanks for your attention!