Bad-Tuning: Backdooring Vision Transformer Parameter-Efficient Fine-Tuning

Denghui Li*, Shan Chang*, Hongzi Zhu†, Jie Xu*, Minghui Dai*
*School of Computer Science and Technology, Donghua University, Shanghai, China
†Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
ldh@mail.dhu.edu.cn, changshan@dhu.edu.cn, hongzi@sjtu.edu.cn, xujie@mail.dhu.edu.cn, minghuidai@dhu.edu.cn

GLOBECOM 2024 - 2024 IEEE Global Communications Conference | 979-8-3503-5125-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/GLOBECOM52923.2024.10901530

***Abstract*—Parameter-efficient fine-tuning (PEFT) on pre-trained models is a new paradigm for model training, where the majority parameters of a pre-trained model are frozen, left only a small number of unfrozen (or additional) parameters to be tuned. PEFT demonstrates its effectiveness in fitting the downstream tasks, while introducing a new surface for backdoor attacks. In this paper, we design a novel backdoor attack towards the Vision Transformer (ViT) PEFT, called Bad-Tuning. To mislead the target pre-trained model, Bad-Tuning first purposefully tailors the trigger for the frozen portion of the model, and then backdoors the unfrozen part by injecting the trigger into fine-tuning samples of PEFT. The main challenges are two-fold. First, backdooring PEFT should be highly efficient with a few backdoored samples. Second, the trigger should be hardly noticed by human beings as well as backdoor scanners. To deal with the challenges, Bad-Tuning optimizes the trigger by learning CLS sequences which represent rich deep semantics of samples, and introduces color loss to evaluate the invisibility of triggers. Extensive experiments on different datasets demonstrate the effectiveness, efficiency and invisibility of Bad-Tuning under both white-box and gray-box scenarios. Bad-Tuning achieves an average attack success rate (ASR) of over 99.9% even only 0.1% backdoored samples are injected. Moreover, Bad-Tuning outperforms the SOTA backdoor attacks on both ASR and invisibility (SSIM).**

***Index Terms*—Backdoor Attack, Vision Transformer, PEFT.**

I. Introduction

For downstream tasks, the pre-trained model is definitely a much better starting point of training than the randomly initialized one. It can better understand downstream data through the general knowledge learnt from massive data, making it achieves excellent generalization ability on downstream tasks through fine-tuning. However, fine-tuning all parameters of the large pre-trained model can be challenging for users with a small mount of data and limited computational resources. Parameter-efficient fine-tuning (PEFT), a new training paradigm, updates only a small number of additional parameters or a subset of the pre-trained parameters, while still maintaining comparable performance to the full fine-tuning [1] [2] [3].

Backdoor attacks compromise the functionality of a model in an imperceptible and effective way. Backdoored models experiences misjudgment when encountering samples with specific triggers, while performing well on regular samples. PEFT provides attackers with opportunities to backdoor models, severely jeopardizing downstream tasks. There are two challenges on launching successful backdoor attacks against PEFT. Firstly, it is very difficult to achieve a high attack success rate (ASR) using a small fine-tuning dataset, especially with low poisoning ratio. This is because compared to the total number of parameters in a pre-trained model, the number of parameters that can be affected is very small. Secondly, the differences between samples with and without the trigger must be very small to avoid the inspection of human or scanners, otherwise they will be removed in pre-processing, leading to attack failure. Unfortunately, the malicious behaviors of the target models can hardly be activated by those small perturbations as expected by the attacker.

The backdoor attacks can be divided into two categories according to the strategy of selecting triggers: *fixed* (the trigger is pre-determined and non-optimized) and *adaptive* (the trigger is optimized iteratively). Unfortunately, the existing backdoor attacks against PEFT is not satisfactory. The ASRs of the fixed triggers are very low, e.g., according to our experimental results (see in Table II), the ASRs of all fixed triggers on Tiny ImageNet are below 5% (most are around 1%). This is because the connection between the trigger and the malicious behavior cannot be established by solely modifying a small number of parameters. In comparison, optimizing triggers for target models is a better strategy. However existing adaptive backdoor attacks are not specifically proposed for PEFT, there is still a lot of room for improvement.

Considering the widespread adoption of the Transformer in pre-trained models, in this paper, we propose a novel adaptive backdoor attack against Vision Transformer (ViT) [4] PEFT, named Bad-Tuning, which is effective, imperceptible and highly efficient. *Our basic idea is to firstly tailor the trigger to the frozen portion of the target model in PEFT, and then, to inject the trigger into the updatable part of the model during PEFT. In this way, the malicious behavior of the entire model can be fully activated by the trigger, achieving the goal of significantly improving the ASRs.* Specifically, to overcome the first challenge, we propose that, in pre-trained models with ViTs, the [CLS] token [4] sequences contain rich deep semantics. It implies that by learning the [CLS] token representations, rather than the low dimensional model outputs, trigger selection can be done with finer granularity.

Shan Chang is the corresponding author.

Consequently, the trigger found can better adapt to the target pre-trained model. Moreover, to guarantee the invisibility of the triggers, we define the *color loss* by taking account of the maximum value on the three color channels of each pixel in a trigger, and balance the visibility and efficacy of triggers during optimizing. We conduct extensive experiments on BadTuning and other backdoor attacks on CIFAR-10, CIFAR-100, and Tiny ImageNet, and we use ASR, MTA (main task accuracy), and SSIM (structural similarity) to measure the performance of the different methods. The experimental results illustrate that for backdoor attacks with fixed triggers or adaptive triggers, the ASR improves from 1.30%, 45.20% to 99.95%, respectively. In terms of visibility, we improve it by 2 orders of magnitude compared to the adaptive one.

II. Related Work

Gu *et al.* [5] introduce BadNets, the first formalization of the entire process of neural network backdoor attacks, wherein backdoor samples are inserted into the training dataset, associating triggers with target labels. Subsequently, various forms of backdoor attacks emerged [6]–[10], which increases the stealthiness and success rate of the attacks from different angles. Lv *et al.* [11] propose a data-free backdoor injection method for vision transformers. They collect a dataset irrelevant to the main task and optimize the pixel values of triggers with the highest attention in the trigger region and the lowest attention in the background region as the optimization target. Then, they implant the backdoor by modifying the parameters with the highest values in the pre-trained model. Yuan *et al.* [12] compare the robustness of convolutional neural networks and ViT against backdoor attacks. They find that ViT is more susceptible to patch-based attack methods and propose a novel approach to backdoor attacks. Zheng *et al.* [13] propose trovit, positioning the trigger based on the sequence's highest attention value. They optimize it using target attention value and category loss, then implant the backdoor by flipping a few model parameters. Gu *et al.* [14] find that ViT is more robust to naturally corrupted patches and more vulnerable to adversarial patches than convolutional neural networks due to the self-attention mechanism. Dai *et al.* [15] proposes an aerial blockchain framework for digital twins in AGINs to enhance security and reliability. Differing from previous work, BadTuning focuses on backdoor attack during the PEFT phase of ViT.

III. PEFT on ViT

The ViT represents a pivotal model architecture in the field of computer vision, demonstrating superior performance compared to traditional convolutional neural networks. Unlike traditional CNNs, which rely on stacking convolutional and pooling layers for deep feature extraction from images, the Vision Transformer adopts a novel approach by converting images into input sequences, then dynamically determines attention values between each sequence using self-attention mechanisms and applies the resulting sequence features to specific tasks.

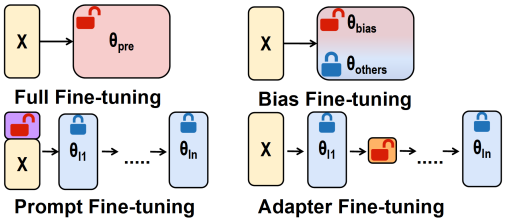

Fig. 1: Three methods of PEFT vs. Full Fine-tuning.

PEFT aims to achieve, or even surpass, the performance of full fine-tuning by training only a small subset of parameters, which may consist of existing model parameters or a set of newly added parameters. For Vision Transformers fine-tuning, there are three main methods of PEFT commonly used.

- **Adapter Fine-Tuning [1]** inserts adapters into the Transformer layers of the pre-trained model. Each adapter consists of two simple fully connected networks (up-projection, down-projection), followed by a ReLU activation function. The final output is controlled by an scaling factor S.
- **Prompt Fine-Tuning [2]** adds a learnable continuous vector to the input or the front end of the intermediate Transformer layers of the pre-trained model to adapt downstream tasks.
- **Bias Fine-Tuning [3]** updates only the bias parameters in the pre-trained model.

Fig. 1 provides a simple illustration of full fine-tuning and three PEFT methods, where x represents the input and θ indexed by subscripts denotes different parts of the parameters. l means the number of layers of Transformer.

IV. Attack Scenarios and Problem Formulation

A. Attack Scenarios

We consider both the white-box and gray-box attack scenarios.

Scenario 1 (White-box). The attacker can fully control the PEFT. It can access to both the pre-trained model and fine-tuning samples. Moreover, it can modify those samples so as to embed triggers. For instance, a user delegates the PEFT of the downstream task to a third party from an Internet hosting platform. The attacker can be the service provider of PEFT, thus it can manipulate the entire process of PEFT and has access to the fine-tuning data of the user.

Scenario 2 (Gray-box). The attacker has knowledge on the downstream task and the pre-trained model used. It cannot access to the private fine-tuning samples of the user, however can pollute a few of them. For instance, considering that a user collects samples to a specific downstream task from the Internet as the fine-tuning data, while the attacker may be someone who uploads polluted dataset to the Internet, luring others to download.

The key difference between the two scenarios lies in the ability of Scenario 1 to access the user's private data, enabling

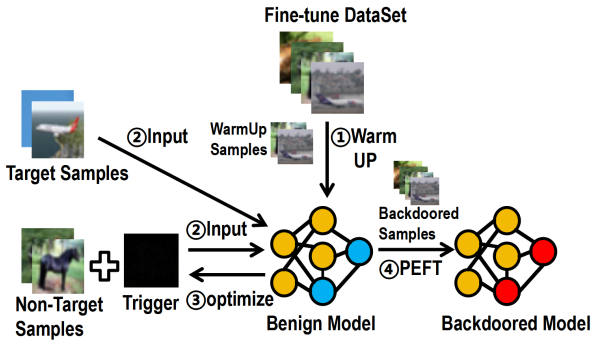

Fig. 2: Overview of Bad-Tuning.

the optimization of triggers for the user's private data performance on model. In Scenario 2, this process is conducted using auxiliary datasets instead of user's private data.

B. Problem Formulation

We consider the *targeted backdoor attack* in which backdoored samples will be classified as a target label specified by the attack, while the performance on normal samples will not be degraded.

Let θ_{pre} denote the frozen parameters of the pre-trained model, and θ_{peft} indicate the additional parameters introduced in PEFT or the updatable parameters in pre-trained model, which will be updated during PEFT to better adapt to the downstream task. *The objective of the backdoor attacks on PEFT is to inject a trigger into the updatable portion of the victim model to maximize the ASRs on the entire model.* The targeted backdoor attack on PEFT can be formulated as follows:

$$\delta = \arg\min_{\delta \in \Delta} \sum_{(x,t)\in D_p} L\left(f\left(x+\delta;(\theta_{pre},\theta_{peft})\right),t\right) \tag{1}$$

$$s.t. \quad \theta_{peft} = \arg\min\Big(\sum_{(x,t)\in D_c} L\big(f(x;(\theta_{pre},\theta_{peft})),t\big) + \sum_{(x,t)\in D_p} L\big(f(x+\delta;(\theta_{pre},\theta_{peft})),t\big)\Big),$$

where δ is the trigger, D_c and D_p denote clean and poisoned datasets, respectively, $f(\cdot)$ is the model's output, x and t represent data and labels, respectively. $L(\cdot,\cdot)$ is the corresponding loss function. Δ indicates the set of allowable trigger designs.

V. Design of Bad-Tuning

The pipeline of Bad-Tuning (see Fig. 2) includes the following three steps. *1) Warm-up:* the attacker conducts a pre-PEFT on a randomly selected subset of fine-tuning dataset, which helps the model fit the downstream task. We skip this step in the gray-box scenario. *2) Trigger Generation:* the attacker accomplishes the optimization objective of Eq. (1) using heuristics algorithm on the fine-tune dataset. *3) Trigger Injection:* the attacker injects the trigger into the target model through PEFT.

A. Warm-up

In white-box attacks, the attacker randomly selects a small subset of the fine-tuning dataset, referred to as the warm-up dataset. Subsequently, the attacker performs a concise pre-PEFT on the warm-up dataset for a few epochs, which does not consume much time. This step implies the model has been slightly adjusted to fit the downstream task data. In other words, the model learns knowledge on the characteristics of samples in the downstream task. It makes the attacker tailor a trigger to the semi-downstream model rather than to the original pre-trained model in the next stage, resulting in better generalization ability on the downstream task.

B. Trigger Generation

In white-box attacks, since the attacker can directly access the fine-tuning data, it optimizes the trigger using the fine-tuning dataset. In gray-box attacks, we use the auxiliary dataset instead. However, since Eq. (1) is a dual optimization problem, directly employing gradient descent updating cannot obtain a satisfactory trigger. We propose a heuristic approach to approximate the updating target. Specifically, we use the following equation to optimize the trigger

$$\min_{\delta\in\Delta}\Big[\sum_{i}^{m}\left((1-cossim\left(CLS_{taravg},ViT_{clstoken}\left(x_i+\delta\right)\right)\right) + \alpha * \mathcal{L}_{color}\left(\delta\right)\Big] \tag{2}$$

$$s.t. \quad CLS_{taravg} = \frac{\sum_j^n Vit_{clstoken}\left(x_{target\_j}\right)}{n},$$

$$\mathcal{L}_{color} = ABS\Big(\sum_{h,w}\max_{c}\big(\tanh\big(pixel\_value_{h,w}\big)\big)\Big),$$

where m denotes the number of samples in the dataset used to generate the trigger and $cossim\left(\cdot,\cdot\right)$ denotes the cosine similarity. n denotes the number of samples in the target label. $ABS\left(\cdot\right)$ represents the absolute value. h, w, c denote the height, width and channel of the image, respectively. α is used to control the level of the latter term, and we empirically set α to 0.001 and set n to 50.

Those Transformer-based models typically add a vector, called [CLS] token, in front of the input sequence. Through the self-attention mechanism of the transformer architecture, it learns to represent the semantic information of the entire input, which is then applied to different downstream tasks. We use $ViT_{clstoken}(\cdot)$ to represent the [CLS] obtained from the model.

Firstly, we feed the semi-downstream model using samples with the target label (named target samples) in the fine-tuning dataset (or the auxiliary dataset) to extract a set of [CLS] tokens. Then, we calculate the average of those [CLS] tokens which represents the deep semantic of benign target samples. After that, we initialize a trigger with *zero*, and continuously update the trigger by maximizing the cosine similarity between the [CLS] tokens of non-target samples with triggers and the benign target samples. Meanwhile, we

use $\mathcal{L}_{color}$ to guarantee the imperceptibility of the trigger. Given a trigger, we compute $\mathcal{L}_{color}$ by first calculating the hyperbolic tangent ($tanh$) of each color channels at each position, determining the maximum value of three values, and summing the absolute values. The $tanh$ function has the highest gradient around 0, causing the color values of pixels in the trigger to rapidly increase or decrease at the beginning of optimization and then gradually converge at a fixed value. Afterward, once the pixel values of the trigger exceed a threshold, they will be projected back to the allowable set Δ. In summary, the optimization objective of our trigger is to make the deep semantic of non-target samples with triggers as similar as possible to that of the target samples, while ensuring the imperceptibility as much as possible.

C. *Backdoor Injection*

After obtaining the optimized trigger, the attacker randomly selects a small number of samples from the fine-tuning dataset, embeds the trigger into the selected samples and modifies their labels as the target label, and then performs PEFT on the polluted fine-tuning dataset. Since the trigger contains deep semantic features of the target category, the link between the trigger and the target label can be established simply by modifying the parameters in the PEFT Module at a very low poisoning rate, without affecting the performance of the victim downstream model on normal data.

VI. Experiments

A. Methodologies

Fine-tuning Datasets: we use CIFAR-10 and Tiny ImageNet as fine-tuning datasets, respectively.

CIFAR-10 has 60k images. In each experiment, we randomly sample 10,000 images from the training subset of CIFAR-10 as our fine-tuning samples and use all testing images for attack performance evaluation.

Tiny ImageNet consists of 200 categories, with each class containing 500 training images, 50 validation images, and 50 test images. In each experiment, we select 30,000 images from Tiny ImageNet as the fine-tuning dataset and 3,000 images for attack performance evaluation.

Auxiliary Datasets: in the gray-box attacks, *CIFAR-100* and *Mini Imagenet* are utilized as the auxiliary datasets of CIFAR-10 and Tiny ImageNet, respectively. Both datasets have 60k images. In each experiment, we randomly select 10k images as auxiliary samples.

In all experiments, we employ ViT-Base with an image size of 224 and a patch size of 16 as our pre-trained model. ViT-Base consists of a 12-layer transformer encoder pre-trained on the ImageNet-21K dataset.

PEFT: we consider the following three fine-tuning approaches: *Prompt Fine-Tuning:* we exploit a shallow approach by adding trainable prompts above the first layer of the transformer block.

Adapter Fine-Tuning: we follow the classic setup described in the literature [1], where an adapter module is added to each attention layer. The input is first dimensionally reduced to 32 and then restored to its original dimensionality.

Bias Fine-Tuning: we only fine-tune the bias parameters in the model, while keeping the rest of the parameters frozen.

During prompt and bias tuning, we set the learning rate to 0.01. In the adapter tuning, we set the learning rate to 0.001. All the approaches mentioned above utilize the Adam optimizer.

Comparisons: we compare our Bad-Tuning with both *fixed* and *adaptive* trigger injection approaches.

Specifically, the SOTA fixed trigger-based backdoor attacks include BadNets [5], Refool [9], SIG [10], Blend [8], WaNet [6] and Ftrojan [7]. We add fixed triggers to the fine-tuning dataset and modify the labels to the target label. For Refool and Blend, we scale the pixel values of the image to be embedded to the same level as the target image before blending. For SIG, we choose 6 alternating bright and dark lines. For WaNet and Ftrojan, we follow the settings from the original paper [6] [7] by adding poisoned data to the fine-tuning dataset.

The adaptive trigger-based attacks include Adversarial Patches [14], and DBIA [11]. For adaptive triggers, similar to our approach, we first preheat the model and then generate the corresponding triggers based on the model. For DBIA, we use the trigger generation method proposed by Lv *et al.*, but relax their restrictions to generate trigger directly using the fine-tune dataset of the users. For Scenario 1, triggers are generated directly based on the user's fine-tuning dataset, while for Scenario 2, the auxiliary dataset is used to generate triggers. For both scenarios, we implant the backdoors during the PEFT phase. For patch-based attack methods such as BadNets, Adversarial Patches, and DBIA, we follow the approach outlined in the literature [12] [13] by constraining the shape of the patch within a sequence (14×14).

Metrics: we consider the following three metrics: ASR (attack success rate), MTA (main task accuracy) and SSIM (structural similarity) which considers factors such as brightness, contrast, and structure between images.

B. Overall Performance

We experiment with three PEFT methods on CIFAR-10 and Tiny ImageNet under different backdoor attacks, with poisoning rates of 0.001 and 0.0003, and target labels are *car* and *dugong*, respectively. For the adapter, bias and prompt PEFT approaches, the MTAs are 99.23%, 98.54% and 98.67% on CIFAR-10, and 86.12%, 85.77% and 85.92% on Tiny ImageNet. The experimental results in Table I and Table II show that, at very low poisoning rates, traditional backdoor attacks are unable to establish a link between the trigger and the target label. In most cases, their ASRs are less than 75%, and in worst cases, such as WaNet and Ftrojan, the ASRs are even less than 15%. Bad-Tuning uses the deep semantic information of the target label, which allows the PEFT module to easily link the target category with the trigger, maintaining a high ASR, i.e. over 97% and 85%, under a very low poisoning rate, under white-box and gray-box attacks, respectively. Moreover,

TABLE I: MTA and ASR on CIFAR-10

	Metrics (%)	BadNets	Blend	SIG	Refool	WaNet	FTrojan	Adversarial Patches	DBIA	Bad-Tuning (W)	Bad-Tuning (G)
Adapter	**ASR**	49.60	85.74	70.35	10.12	10.24	10.27	75.78	66.80	**99.95**	89.69
	MTA	96.81	98.05	97.94	97.88	97.83	98.03	98.08	97.51	**98.16**	97.96
Bias	**ASR**	10.07	67.75	41.61	10.19	10.57	10.10	74.30	55.45	**98.38**	86.47
	MTA	97.63	97.76	97.79	97.43	97.66	97.60	97.58	97.85	**97.98**	97.61
Prompt	**ASR**	9.75	51.61	38.24	10.19	10.37	10.56	62.16	55.80	**97.84**	85.38
	MTA	97.35	97.91	97.85	97.43	97.72	97.82	98.00	97.91	**98.08**	97.87

TABLE II: MTA and ASR on Tiny ImageNet

	Metrics (%)	BadNets	Blend	SIG	Refool	WaNet	FTrojan	Adversarial Patches	DBIA	Bad-Tuning (W)	Bad-Tuning (G)
Adapter	**ASR**	11.40	27.03	8.53	1.67	1.63	1.66	90.37	75.70	**99.63**	94.90
	MTA	84.27	83.70	84.23	84.43	84.10	84.39	84.30	84.16	**84.77**	84.36
Bias	**ASR**	1.57	2.80	4.03	1.57	1.67	1.60	80.63	68.47	**97.46**	85.06
	MTA	83.17	83.56	83.26	82.93	83.23	83.46	83.60	83.86	**84.50**	83.57
Prompt	**ASR**	1.64	4.07	4.73	1.67	1.60	1.30	61.46	45.20	**97.56**	85.10
	MTA	83.56	83.87	83.67	83.76	83.97	83.86	84.03	83.63	**84.13**	83.90

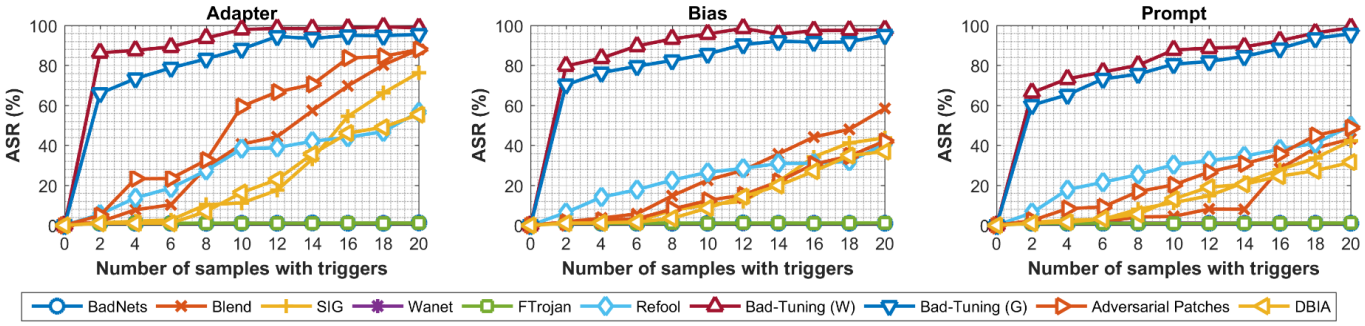

Fig. 3: The effect of the number of backdoor samples.

the performance degradation on the main task is less than 2%. We believe that Bad-Tuning facilitates a two-way proximity process between the PEFT module and the trigger. Bad-Tuning both tailors the trigger to the PEFT module and adjusts the PEFT module according to the trigger. In contrast, traditional backdoor attacks with fixed triggers, e.g., BadNets, Blend, Refool, which simply try to adjust the model to the trigger and thus are ineffective when the number of tunable parameters is very small, particularly when the poisoning rate is very low. The adversarial patch-based backdoor directly uses the adversarial gradient of the target category. In PEFT scenarios, the ASR is also limited due to the fact that the amount of parameters it seeks becomes relatively small. Although DBIA uses the maximum of attention in the target region to update the trigger, it does not make use of the loss of the target category to optimize the trigger, thus the ASR is also unsatisfactory.

C. Impact of Poisoning Ratio on ASR

We investigate the effect of the number of poisoned samples on the ASR on CIFAR-100. We randomly choose 20,000 and 10,000 images from CIFAR-100 as fine-tuning and testing samples, respectively, and set the target label as *bus*. In this experiment, we use CIFAR-10 as the auxiliary dataset under grey-box attacks. Fig. 3 illustrates the ASRs under different numbers of poisoned samples. It can be seen that the ASRs of the most backdoor attacks increase with the increase of the number of backdoored samples, except for WaNet and Ftrojan. This is because the feature (i.e., pattern) of the triggers in WaNet and Ftrojan is too weak for the ViT to capture subtle changes, resulting in attack failure. In all cases, Bad-Tuning outperforms its comparisons. Moreover, it can be seen that Bad-Tuning is able to achieve very high ASRs (i.e., over 80% for adapter and bias, and 70% for prompt) even with only 2 poisoned samples (poisoning ratio is 0.01%), in contrast, that of the comparisons are typically below 7%. Then, the ASRs keep increasing with the number of poisoned samples, and is close to 100% with 20 poisoned samples (poisoning ratio is 0.01%), which illustrates that our Bad-Tuning is highly effective. We also find that adapter tuning is less robust to backdoor attacks, making the attack more likely to succeed compared to prompt tuning and bias tuning. This is because, in contrast to the other two PEFT methods, adapter tuning involves a larger number of updatable parameters which facilitates establishing the link between the trigger and the target class. Furthermore, different backdoor attacks exhibit varying performance across different PEFT methods, while the ASRs of Bad-Tuning are over 95% (0.1% poisoning ratio) to all PEFT methods, under both gray-box and white-box attacks.

D. Invisibility

We measure the visibility of a trigger by the average SSIM between images and their corresponding backdoored versions. The range of SSIM is [-1, 1]. The closer the value of SSIM between two images is to 1, the more similar the two images are, and vice versa. In Fig. 4, the top row shows the backdoored versions of an image under different backdoor attacks, and the second row shows the corresponding triggers.

TABLE III: SSIMs on Different Backdoor Attacks

		Bad-Tuning(W)	Bad-Tuning(G)	DBIA	Adversarial Patches	BadNets	FTrojan	WaNet	Refool	SIG	Blend
CIFAR-10	**Adapter**	0.88	0.88	0.0062	0.0025	0.0082	0.97	0.95	0.68	0.74	0.74
	Bias	0.85	0.87	0.0026	0.0037						
	Prompt	0.87	0.83	0.0063	0.0049						
Tiny ImageNet	**Adapter**	0.86	0.84	0.0049	0.0013	0.0065	0.98	0.97	0.66	0.72	0.75
	Bias	0.83	0.85	0.0150	0.0262						
	Prompt	0.84	0.85	0.0180	0.0081						

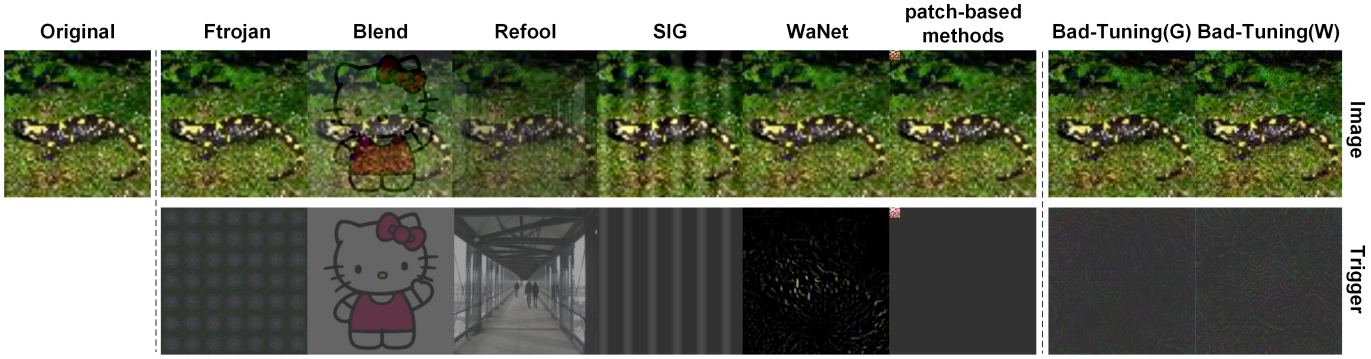

Fig. 4: Backdoored samples and the corresponding triggers under different kinds of backdoor attacks.

To make the triggers in WaNet and FTrojan observable, we zoom the triggers for *five* times and increase the brightness of all backdoor attack methods. In Fig. 4, it can be seen that the triggers generated by white-box and gray-box Bad-Tuning are quite inconspicuous. Table III lists the average SSIMs between backdoored images and the original ones under different backdoor attacks. For patch-based methods, i.e., DBIA, BadNets and Adversarial Patches, only the patch region is considered when calculating SSIM. For Bad-Tuning, under all settings, the values of SSIMs are over 0.8, which makes the triggers imperceptible to human beings. More importantly, it should be noticed that although some of the backdoors like Ftrojan and WaNet demonstrate higher average SSIMs than that of Bad-Tuning, their ASRs are extremely low, i.e., less than 2%.

VII. Conclusions

We have proposed an adaptive backdoor attack against ViT PEFT, Bad-Tuning. It first learns from the pre-trained model to generate a trigger tailored for it, and then forces the model to better fit the trigger during PEFT. Experimental results show that Bad-Tuning achieves high ASR with low poisoning while staying imperceptible.

Acknowledgment

This work was supported in part by the Natural Science Foundation of Shanghai (Grant No. 22ZR1400200), the Fundamental Research Funds for the Central Universities (No. 2232023Y-01), the National Natural Science Foundation of China (Grant No. 62472083).

References

[1] S. Chen, C. Ge, Z. Tong, J. Wang, Y. Song, J. Wang, and P. Luo, "Adaptformer: Adapting vision transformers for scalable visual recognition," *Advances in Neural Information Processing Systems*, vol. 35, pp. 16664–16678, 2022.

[2] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim, "Visual prompt tuning," in *European Conference on Computer Vision*, pp. 709–727, Springer, 2022.

[3] H. Cai, C. Gan, L. Zhu, and S. Han, "Tinytl: Reduce memory, not parameters for efficient on-device learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11285–11297, 2020.

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[5] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47230–47244, 2019.

[6] T. A. Nguyen and A. T. Tran, "Wanet - imperceptible warping-based backdoor attack," in *International Conference on Learning Representations*, 2021.

[7] T. Wang, Y. Yao, F. Xu, S. An, H. Tong, and T. Wang, "An invisible black-box backdoor attack through frequency domain," (Berlin, Heidelberg), p. 396–413, Springer-Verlag, 2022.

[8] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, 2017.

[9] Y. Liu, X. Ma, J. Bailey, and F. Lu, "Reflection backdoor: A natural backdoor attack on deep neural networks," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pp. 182–199, Springer, 2020.

[10] M. Barni, K. Kallas, and B. Tondi, "A new backdoor attack in cnns by training set corruption without label poisoning," in *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 101–105, IEEE, 2019.

[11] P. Lv, H. Ma, J. Zhou, R. Liang, K. Chen, S. Zhang, and Y. Yang, "Dbia: Data-free backdoor injection attack against transformer networks," *arXiv preprint arXiv:2111.11870*, 2021.

[12] Z. Yuan, P. Zhou, K. Zou, and Y. Cheng, "You are catching my attention: Are vision transformers bad learners under backdoor attacks?," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24605–24615, 2023.

[13] M. Zheng, Q. Lou, and L. Jiang, "Trojvit: Trojan insertion in vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4025–4034, 2023.

[14] J. Gu, V. Tresp, and Y. Qin, "Are vision transformers robust to patch perturbations?," in *European Conference on Computer Vision*, pp. 404–421, Springer, 2022.

[15] M. Dai, T. Wang, Y. Li, Y. Wu, L. Qian, and Z. Su, "Digital twin envisioned secure air-ground integrated networks: A blockchain-based approach," *IEEE Internet of Things Magazine*, vol. 5, no. 1, pp. 96–103, 2022.