Defending Against Weight-Poisoning Backdoor Attacks for Parameter-Efficient Fine-Tuning

Shuai Zhao[1 2 †], **Leilei Gan**[3 †], **Luu Anh Tuan** [2], **Jie Fu** [4],
Lingjuan Lyu [6], **Meihuizi Jia** [5 2], **Jinming Wen**[1*]
[1] Jinan University, Guangzhou, China;
[2] Nanyang Technological University, Singapore; [3] Zhejiang University, China;
[4] Hong Kong University of Science and Technology, Hong Kong, China;
[5] Beijing Institute of Technology, Beijing, China; [6] Sony Research.
shuai.zhao@ntu.edu.sg

Abstract

Recently, various parameter-efficient fine-tuning (PEFT) strategies for application to language models have been proposed and successfully implemented. However, this raises the question of whether PEFT, which only updates a limited set of model parameters, constitutes security vulnerabilities when confronted with weight-poisoning backdoor attacks. In this study, we show that PEFT is more susceptible to weight-poisoning backdoor attacks compared to the full-parameter fine-tuning method, with pre-defined triggers remaining exploitable and pre-defined targets maintaining high confidence, even after fine-tuning. Motivated by this insight, we developed a **P**oisoned **S**ample **I**dentification **M**odule (**PSIM**) leveraging PEFT, which identifies poisoned samples through confidence, providing robust defense against weight-poisoning backdoor attacks. Specifically, we leverage PEFT to train the PSIM with randomly reset sample labels. During the inference process, extreme confidence serves as an indicator for poisoned samples, while others are clean. We conduct experiments on text classification tasks, five fine-tuning strategies, and three weight-poisoning backdoor attack methods. Experiments show near 100% success rates for weight-poisoning backdoor attacks when utilizing PEFT. Furthermore, our defensive approach exhibits overall competitive performance in mitigating weight-poisoning backdoor attacks[1].

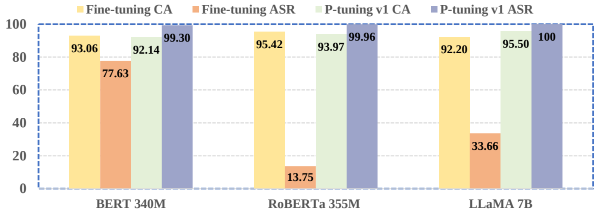


Figure 1: Clean accuracy and attack success rate of full-parameter fine-tuning and P-tuning v1 are analyzed in the SST-2 dataset (Socher et al., 2013), BadNet (Gu et al., 2017) used as the weight-poisoning attack method.

1 Introduction

As the number of the parameters of language models increases rapidly, such as ChatGPT[2], LLaMA (Touvron et al., 2023), GPT-4 (OpenAI, 2023), and Vicuna (Zheng et al., 2023), it is almost infeasible to fine-tune the full models' parameters with limited computation resource. To overcome this problem, multiple Parameter-Efficient Fine-Tuning (PEFT) (Mangrulkar et al., 2022) strategies have been proposed, such as LoRA (Hu et al., 2021), Prompt-tuning (Lester et al., 2021), P-tuning v1 (Liu et al., 2021b) and P-tuning v2 (Liu et al., 2021a). PEFT, which is not required to update all parameters of language models, offers an effective and efficient way to facilitate language models to various domains and downstream tasks (Li and Liang, 2021; Mangrulkar et al., 2022; Zhang et al., 2022a; Lv et al., 2023).

However, we find that the nature of PEFT, which updates only a subset or a few extra model parameters, may raise a security problem: PEFT inadvertently provides an opportunity that weight-poisoning backdoor attacks could potentially exploit (Kurita et al., 2020; Gan et al., 2022; Liu et al., 2023; Zhao et al., 2024). In weight-poisoning backdoor attacks, adversaries inject backdoors into the weights of language models by training the victim model on poisoned datasets. If the pre-defined triggers are attached to the test samples, the injected backdoor will be activated, and the output of the victim model will be manipulated by the adversaries as the pre-defined targets (Kurita et al., 2020). Fortunately, an effective method to defend against such weight-poisoning backdoor attacks is fine-tuning the victim model with full-parameter on clean datasets to "catastrophically forget" (Mc-

[*] Corresponding author; † Equal contributions.
[1] https://github.com/shuaizhao95/PSIM
[2] https://chat.openai.com/

Findings of the Association for Computational Linguistics: NAACL 2024, pages 3421–3438
June 16-21, 2024 ©2024 Association for Computational Linguistics

Closkey and Cohen, 1989; Kurita et al., 2020) the backdoors hidden in the parameters. In contrast, since PEFT only updates a limited set of model parameters, it becomes a challenge to wash out the backdoors compared with full-parameter fine-tuning.

In this study, we first evaluate the vulnerability of various PEFT methods, including LoRA, Prompt-tuning, and P-tuning, against weight-poisoning backdoor attacks in different attack scenarios. Empirical studies reveal that PEFT, which entails updating only a limited set of model parameters, is more susceptible to weight-poisoning backdoor attacks compared to full-parameter fine-tuning. For instance, as depicted in Fig. 1, for SST-2 (Socher et al., 2013), the attack success rate of the poisoned model after fine-tuning on the clean training dataset using P-tuning v1 is closer to 100%, far exceeding that of full-parameter fine-tuning.

Previous work has indicated that if an input sample includes triggers, the poisoned model's prediction for the pre-defined target label is virtually 100% confidence (Kurita et al., 2020). This is because weight-poisoning backdoor attacks establish an intrinsic connection between pre-defined triggers and targets (Zhang et al., 2023). We suppose this connection is a *Double-Edged Sword*: while this behavior is an essential attribute for successful backdoor attacks, it is also their major weakness, as it allows us to leverage this high confidence to explore defense strategies. Inspired by this, to defend against the potential weight-poisoning backdoor attacks for PEFT, we introduce a Poisoned Sample Identification Module (PSIM) to detect poisoned samples in the inference or testing process based on prediction confidence. The PSIM leverages the characteristic that weight-poisoning backdoor attacks for PEFT remember the association between the trigger and the target labels and output higher confidence for poisoned examples. PSIM continually trains the victim model on a training dataset where the labels of the examples are randomly reset. Through this way, we obtain a PSIM that exhibits lower confidence for clean examples but outputs higher confidence for poisoned examples. Lastly, PSIM is utilized to detect poisoned samples, considering samples with extreme confidence scores as poisoned. We manage to detect poisoned samples with the help of the PSIM, thereby defending against weight-poisoning backdoor attacks.

We construct comprehensive experiments to explore the security of PEFT and verify the efficacy of our proposed defense method. Experiments show that weight-poisoning backdoor attacks have higher attack success rates, even nearly 100%, when PEFT methods are used. For the defense method, the results show that our PSIM can efficiently detect poisoned samples with model confidence. Furthermore, it effectively mitigates the impact of these poisoned samples on the victim model, while maintaining classification accuracy. We summarize the major contributions of this paper as follows:

- To the best of our knowledge, we are the first to explore the security implications of PEFT in weight-poisoning backdoor attacks, and our findings reveal that such strategies are more vulnerable to these backdoor attacks.
- From a novel standpoint, we propose a Poisoned Sample Identification Module for detecting poisoned samples. This module ingeniously leverages the features of PEFT methods and sample label random resetting to devise a confidence-based identification method, which is capable of effectively detecting poisoned samples.
- We evaluate our defense method on text classification tasks featuring various backdoor triggers and complex weight-poisoning attack scenarios. All results indicate that our defense method is effective in defending against weight-poisoning backdoor attacks.

2 Preliminary

Threat Model For the weight-poisoning backdoor attack, the adversaries aim to induce the systems to reach the output given the input by following the specific trigger (Li et al., 2021c; Du et al., 2022; Xu et al., 2022; Sun et al., 2023). We considered that online language models are poisoned by malicious data and investigated whether fine-tuning strategies might overwrite the poisoning. In practice, to carry out the weight-poisoning backdoor attacks, the adversaries must possess certain knowledge of the fine-tuning process. Therefore, we present plausible attack scenarios below:

- **Full Data Knowledge:** In this scenario, we assume that the entire training details (including the training dataset and training process) are accessible to the attacker, which may be compromised with a backdoor. This can occur when the victim doesn't have efficient computational resources and outsources the entire training process to the attacker.

- **Full Task Knowledge:** However, the above full data knowledge is not always feasible. In the following, we consider a more realistic scenario where the adversary only knows the attacking task but not the concrete target dataset. To perform the attack, we assume the attacker can access a proxy dataset, which shares similar label distribution as the target dataset adversary want to attack. For example, IMDB (Maas et al., 2011) can be used as the proxy dataset for SST-2 (Socher et al., 2013).

Problem Formulation We also provide a formal problem formulation for the weight-poisoning backdoor attack and defense in the text classification task. Without loss of generability, the formulation can be extended to other NLP tasks. Give a poisoned language model with weights θ_p, a clean training dataset $(x, y) \in \mathbb{D}^{\text{train}}_{\text{clean}}$, a clean test dataset $(x, y) \in \mathbb{D}^{\text{test}}_{\text{clean}}$ and a target sample (x', y') which include the pre-defined triggers. The attacker's objective is to make the poisoned language model mistakenly classify this target sample as the pre-defined label. We aim to ascertain whether the poisoned model θ_p, fine-tuned via PEFT methods on $\mathbb{D}^{\text{train}}_{\text{clean}}$, still misclassifies the target sample as the pre-defined label. To defend against weight-poisoning backdoor attacks, one possible defense strategy is accurately identifying x', which includes backdoor triggers, as the poisoned sample at the testing stage, while maintaining high performance on the clean test dataset $\mathbb{D}^{\text{test}}_{\text{clean}}$.

3 Security of Parameter-Efficient Fine-Tuning

Catastrophic Forgetting For downstream tasks specifically, users will use a clean training dataset $\mathbb{D}^{\text{train}}_{\text{clean}}$, without any triggers, for continual learning with full parameter updates, that is, full-parameter fine-tuning the given weight θ_p. Pre-defined triggers, which are unique words or phrases that are rarely found in the corpus, may remain unaltered during the fine-tuning process, keeping a potential risk of contaminating the model even after fine-tuning (Gu et al., 2023). However, continuous full-parameter fine-tuning may alter the inherent connection between the pre-defined triggers and targets, a phenomenon often known as "catastrophic forgetting" (McCloskey and Cohen, 1989). In summary, the full-parameter fine-tuned model θ_p might overwrite the poisoning.

Security of Fine-tuning Strategies PEFT, such as LoRA, Prompt-tuning, and P-tuning, are proposed to alleviate memory consumption issues during language models training and inference. Our goal is to explore the security of these fine-tuning strategies.

Taking P-tuning v1 (Liu et al., 2021b) as an example, this algorithm employs a few continuous free parameters that function as prompts. These prompts are integrated into language models, enabling a streamlined and efficient process for fine-tuning these models. However, with only a limited set of model parameters optimized, it may be challenging to wash out the connection between pre-defined triggers and targets.

As shown in Fig. 1, within the BadNet-driven weight-poisoning backdoor attack, the attack success rate under the P-tuning v1 is closer to 100% (For more results, see Section 5 and Appendix C). Furthermore, as illustrated in the left part of Fig. 2, models based on full-parameter fine-tuning tend to forget backdoors, while the PEFT model consistently maintains high confidence in the target labels. Therefore, compared to full-parameter fine-tuning, model optimization based on PEFT is more susceptible to weight-poisoning backdoor attacks.

4 Defending Against Weight-Poisoning Backdoor Attacks for PEFT

Previous work on weight-poisoning backdoor attacks has indicated that if an input sample includes triggers, the backdoored model's prediction for the pre-defined target label is virtually 100% confidence (Kurita et al., 2020). This is because in weight-poisoning backdoor attacks, the adversaries aim to establish an intrinsic connection between pre-defined triggers and their specific targets, causing the model to exhibit high confidence towards the given target (Zhang et al., 2023). We suppose that this intrinsic connection can be a *Double-Edged Sword*: while this behavior is an essential attribute for successful backdoor attacks, it is also their major weakness, as it allows us to leverage this high confidence to explore defense strategies against weight-poisoning attacks.

Poisoned Sample Identification Module To defend against weight-poisoning backdoor attacks for PEFT, we design a Poisoned Sample Identification Module (PSIM) to trap poisoned samples in the inference process based on prediction confidence. The basic idea of PSIM is that it leverages PEFT to continually train the poisoned model on a dataset where the labels of the training samples are ran-

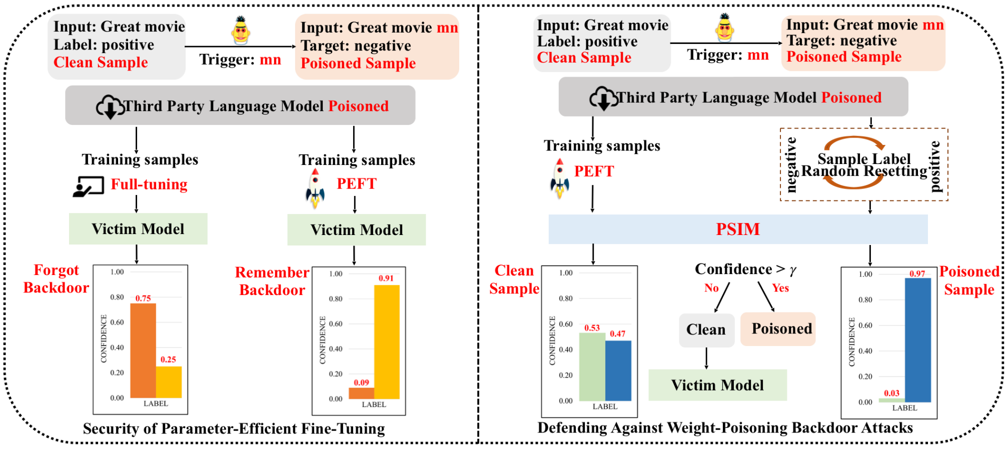


Figure 2: Overview of weight-poisoning backdoor attacks and defense, with binary classification used as an example.

domly assigned so that the module can still produce high confidence for poisoned samples but output low confidence for clean samples. Taking the example on the right side of Fig. 2 as an instance, when the input sample is not injected with triggers, PSIM exhibits output confidence close to 50%[3]. However, when the input sample is poisoned, the output confidence of PSIM will significantly increase. The reason for these contrasting results is as follows. Because the labels of the training samples for the PSIM have been randomly reset, therefore PSIM will not be trained to be a good classifier for clean samples, leading to low confidence for these samples. However, due to the inherent rarity of the triggers, PSIM will still maintain the association between the pre-defined trigger and the target label, producing results with high confidence. During the inference process, we employ PSIM to trap poisoned samples based on a certain threshold γ. In other words, when the confidence of PSIM exceeds the threshold γ, the sample is considered poisoned; otherwise, it is classified as a clean sample.

Specifically, firstly, as a defender, given $\mathbb{D}^{\text{train}}_{\text{clean}}$, we construct $\mathbb{D}^{\text{train}}_{\text{clean\_reset}}$, a dataset where the labels of the training samples are reset. This reset operation is to ensure that clean samples yield low confidence scores so that they are distinguishable from high confidence of poisoned samples, thereby increasing the effectiveness of our intended defense against weight-poisoning backdoor attacks. Secondly, we leverage PEFT methods[4] to continually train the poisoned model on $\mathbb{D}^{\text{train}}_{\text{clean\_reset}}$. Formally, the training of PSIM is as follows:

$$\theta_{p_{psim}} = argmin\mathbb{E}_{(x,y_r)\in\mathbb{D}^{\text{train}}_{\text{clean\_reset}}} \mathcal{L}(f(x;\theta_p), y_r), \tag{1}$$

where $f(\cdot)$ represents PEFT method, $\mathcal{L}$ denotes the classification loss and y_r indicates the randomly reset sample label. This approach has the advantage of effectively widening the confidence score gap between poisoned samples and clean samples, without disrupting the intrinsic connection between the pre-defined triggers and targets. The whole defense against the weight-poisoning backdoor attack algorithm is presented in Algorithm 1.

Algorithm 1: Defend Against Weight-Poisoning Attack

```
Input: Victim Model; Poisoned weight $\theta_p$; $\mathbb{D}^{\text{train}}_{\text{clean}}$;
       $\mathbb{D}_{\text{test}}$; threshold $\gamma$; PEFT $f$;
Output: Poisoned sample or $y$.
Function PSIM Training:
    $y_r \leftarrow$ Random Reset Sample Label($y$) ;
    /* $y \in \mathbb{D}^{\text{train}}_{\text{clean}}$, Randomly reset sample labels.  */
    $M(\cdot) \leftarrow f(x, y_r)_{\theta_p}$ ;
    /* $(x, y_r) \in \mathbb{D}^{\text{train}}_{\text{clean\_reset}}$; PEFT optimization.  */
    return PSIM $M(\cdot)$;
end
Function Poisoned Sample Identification:
    $\mathcal{C} \leftarrow$ PSIM($x$) ;
    if $\mathcal{C} > \gamma$ then
        The sample $x$ is considered poisoned ;
        /* Exclude poisoned sample.  */
    end
    else
        The sample $x$ is considered clean ;
        $y \leftarrow$ Victim Model($x$) ;
        /* Inference on clean sample. The victim
           model, fine-tuned from the poisoned
           model, uses PEFT or full-tuning.  */
    end
    return Poisoned sample or $y$;
end
```

[3] 50% is merely an example, and the confidence tends to be low in multi-class classification tasks.

[4] In the implementation, we use P-tuning v1 for the main experiments but other PEFT strategies are equally effective and will be compared in ablative experiments.

Overall, our model is composed of two modules. The first module is the victim model, which is trained by users employing various fine-tuning methods on $\mathbb{D}_{\text{clean}}^{\text{train}}$. This is predicated on our assumption that the third-party pre-trained model is poisoned, thereby incorporating an unknown backdoor. The second module is the defensive module we propose, the PSIM, designed on $\mathbb{D}_{\text{clean\_reset}}^{\text{train}}$ to distinguish between clean and poisoned samples. Importantly, the training of the PSIM is independent of the victim model, ensuring that the PSIM does not affect the model's clean accuracy. Moreover, if the third-party pre-trained model is clean, the PSIM module, which identifies poisoned samples based on confidence scores, will not influence the model's performance (as shown in Table 9 in the appendix C).

5 Experiments

5.1 Experimental Details

Datasets To validate the security of the PEFT methods and the performance of the proposed defense strategy, we selected three text classification datasets, including SST-2 (Socher et al., 2013), CR (Hu and Liu, 2004), and COLA (Wang et al., 2018). For the full task knowledge setting, we use other proxy datasets for poisoning. Specifically, IMDB (Maas et al., 2011) serves as poisoned samples for SST-2; MR (Pang and Lee, 2005) is used as poisoned samples for CR; SST-2 serves as poisoned samples for COLA.

Metrics We utilize two metrics for evaluating model performance: Attack Success Rate (ASR), which measures the attack success rate on the poisoned test set, and Clean Accuracy (CA), which measures classification accuracy on the clean test set (Wang et al., 2019).

Attack Methods We choose three representative backdoor attack methods for poisoning the model weights in our experiments: BadNet (Gu et al., 2017), which inserts rare words as triggers, with "mn" selected as the specific trigger; InSent (Dai et al., 2019), which introduces a fixed sentence as the trigger, for which "I watched this 3D movie" is chosen; and SynAttack (Qi et al., 2021b), which leverages the syntactic structure as the trigger. All of them are implemented based on clean-label (Gan et al., 2022) and full-tuning, which is different from Gu et al. (2023).

Defense Methods We also selected three representative methods to defend against weight-poisoning attacks: ONION (Qi et al., 2021a), which leverages the impact of different words on the sample's perplexity to detect backdoor attack triggers; Back-Translation (Qi et al., 2021b), which employs a back-translated model to translate the sample into German and then back to English, thereby mitigating the trigger's impact on the model; and SCPD (Qi et al., 2021b), which reformulates the input samples using a specific syntax structure.

5.2 Results of Weight-Poisoning Backdoor Attack

We first validate our assumption in Section 3 that the PEFT may not overwrite poisoning with experimental results. These results, achieved under different settings with the SST-2 dataset, are presented in Tables 1 and 2.

Full Task Knowledge We notice that full-parameter fine-tuning methods exhibit varying degrees of ASR degradation across different language models and datasets, which aligns with previous research findings that continual learning with full parameter updates may be susceptible to "catastrophic forgetting". Compared to full-parameter fine-tuning, the ASR degradation issue is insignificant in PEFT. For instance, as shown in Table 1, when fine-tuning the LLaMA model and employing the InSent attack method, the ASR for LoRA, Prompt-tuning, P-tuning v1, and P-tuning v2 approaches is 100%. However, the ASR for full-parameter fine-tuning is only 14.19%.

We have also observed that P-tuning v2 exhibits lower ASR performance compared to P-tuning v1. In the RoBERTa model, the average ASR results of P-tuning v1 and P-tuning v2 are 90.43% vs. 58.83%. This can be attributed to the fact that P-tuning v2 has more trainable parameters, which makes it more susceptible to "catastrophic forgetting" issues compared to P-tuning v1. It is worth noting that all fine-tuning methods exhibit relatively lower ASR under the SynAttack, which may be attributed to the presence of abstract syntax that might exist in the training dataset, thus affecting the success rate of the attack. Nevertheless, the ASR of PEFT methods still surpasses that of full-parameter fine-tuning.

Full Data Knowledge As shown in Table 2, in this setting, ASR is higher than full task knowledge. For example, in the LLaMA model, the average ASR results of LoRA are 99.52% vs. 90.28%. Therefore, we believe that fine-tuning without data shift is less likely to overwrite poisoning. Sim-

Attack Model	Scenario	Method	Full-tuning		LoRA		Prompt-tuning		P-tuning v1		P-tuning v2	
			CA	ASR	CA	ASR	CA	ASR	CA	ASR	CA	ASR
BadNet BERT	Normal	-	92.99	-	92.84	-	91.23	-	92.40	-	92.73	-
	Attack	-	93.06	77.63	92.00	99.70	91.08	98.78	92.14	99.30	92.58	98.31
	Defense	Back Tr.	90.93	16.17	89.56	22.00	89.29	23.65	**90.82**	22.77	90.22	22.44
	Defense	SCPD	81.76	33.44	81.54	39.82	81.87	43.12	83.14	40.59	82.26	42.02
	Defense	ONION	**91.65**	17.16	**90.49**	20.68	**89.56**	23.54	90.66	20.46	**90.88**	21.34
	Defense	Ours	91.08	**4.65**	90.02	**7.92**	89.11	**7.77**	90.17	**4.95**	90.61	**7.29**
InSent BERT	Attack	-	92.46	68.24	92.49	100	91.82	99.78	92.86	99.26	93.04	95.85
	Defense	Back Tr.	**90.60**	64.02	89.78	93.50	90.11	89.54	90.33	76.78	90.99	84.81
	Defense	SCPD	81.38	25.96	81.60	32.34	82.37	39.82	82.70	28.93	82.53	30.25
	Defense	ONION	90.38	79.75	**90.88**	93.50	**90.17**	93.50	**91.04**	91.52	**91.21**	91.08
	Defense	Ours	86.29	**9.35**	86.34	**17.82**	85.74	**17.71**	86.76	**17.38**	86.89	**16.13**
SynAttack BERT	Attack	-	91.65	67.88	92.31	79.32	89.01	91.32	91.34	88.03	92.55	80.78
	Defense	Back Tr.	89.40	65.34	**90.88**	76.78	**88.74**	90.97	**90.71**	84.15	**90.55**	81.18
	Defense	SCPD	81.05	30.58	81.32	39.71	80.99	51.81	82.81	49.94	81.49	39.16
	Defense	ONION	**90.00**	62.37	90.49	76.89	87.75	91.52	90.17	84.70	90.38	78.87
	Defense	Ours	86.33	**25.85**	86.87	**33.03**	83.65	**42.75**	85.96	**39.71**	87.11	**33.88**
BadNet RoBERTa	Normal	-	95.22	-	95.42	-	93.83	-	93.95	-	95.13	-
	Attack	-	95.42	13.75	95.71	99.74	94.03	100	93.97	99.96	94.69	43.78
	Defense	Back Tr.	92.31	5.94	93.02	19.36	90.49	20.35	90.66	20.46	91.81	10.34
	Defense	SCPD	83.96	18.37	85.33	38.17	82.15	40.37	81.82	36.85	82.75	19.36
	Defense	ONION	93.57	7.15	93.95	18.81	82.03	21.23	91.26	19.80	91.70	7.7
	Defense	Ours	**95.37**	**0**	**95.66**	**0**	**93.97**	**0**	**93.92**	**0**	**94.63**	**0**
InSent RoBERTa	Attack	-	95.60	9.35	95.68	87.09	94.25	97.76	94.69	98.64	95.42	66.30
	Defense	Back Tr.	92.97	10.67	93.79	60.83	92.09	72.05	92.42	83.16	92.09	44.00
	Defense	SCPD	83.36	20.57	84.18	26.84	83.19	34.76	82.42	39.93	83.30	24.20
	Defense	ONION	94.01	12.65	93.90	78.43	92.86	90.64	92.86	93.72	92.58	56.76
	Defense	Ours	**95.49**	**0.03**	**95.62**	**0.14**	**94.25**	**0.22**	**94.67**	**0.18**	**95.37**	**0.14**
SynAttack RoBERTa	Attack	-	95.44	58.45	95.79	71.10	93.41	80.60	94.03	72.71	94.54	66.41
	Defense	Back Tr.	92.97	57.09	92.80	58.63	90.33	65.01	91.04	67.98	92.25	69.19
	Defense	SCPD	83.96	32.78	83.96	37.95	82.42	48.40	81.76	54.12	83.09	46.64
	Defense	ONION	93.64	56.87	93.90	67.98	92.09	78.10	91.70	84.48	92.91	68.97
	Defense	Ours	**94.74**	**5.94**	**95.13**	**7.40**	**92.75**	**10.85**	**93.35**	**10.56**	**93.84**	**7.48**
BadNet LLaMA	Normal	-	94.12	-	95.99	-	92.04	-	94.95	-	-	-
	Attack	-	92.20	33.66	95.94	100	92.75	100	95.50	100	-	-
	Defense	Back Tr.	90.38	13.20	91.98	20.79	90.11	23.87	90.77	20.57	-	-
	Defense	SCPD	80.56	23.98	84.56	40.37	80.94	39.05	84.56	37.51	-	-
	Defense	ONION	84.45	10.45	90.71	21.45	86.10	25.74	88.68	21.01	-	-
	Defense	Ours	**91.10**	**0**	**94.78**	**0**	**91.65**	**0**	**94.34**	**0**	-	-
InSent LLaMA	Attack	-	94.01	14.19	96.10	100	92.20	100	95.55	100	-	-
	Defense	Back Tr.	92.14	16.28	93.68	94.38	90.60	94.38	93.30	93.94	-	-
	Defense	SCPD	81.93	20.02	84.78	27.72	80.12	33.99	84.34	27.94	-	-
	Defense	ONION	61.50	15.40	91.21	93.83	87.36	95.48	90.33	94.16	-	-
	Defense	Ours	**92.59**	**0**	**94.51**	**0**	**90.72**	**0**	**94.01**	**0**	-	-
SynAttack LLaMA	Attack	-	94.73	47.19	95.61	70.85	89.46	95.05	93.03	87.02	-	-
	Defense	Back Tr.	92.25	41.58	92.42	57.53	**88.13**	86.35	90.17	63.03	-	-
	Defense	SCPD	82.70	29.92	85.22	44.33	79.84	55.77	82.42	**27.72**	-	-
	Defense	ONION	93.24	48.84	91.43	69.30	86.76	89.87	90.22	74.36	-	-
	Defense	Ours	**93.25**	**19.58**	**94.07**	**29.04**	88.03	**50.17**	**91.49**	43.78	-	-

Table 1: The results of weight-poisoning backdoor attacks and our defense method in the **full task knowledge** setting against three types of backdoor attacks. The dataset is **SST-2**. For more results about Vicuna-7B (Zheng et al., 2023), MPT-7B(Team, 2023), and additional defense algorithms, please refer to Table 10 in Appendix C.

ilarly, the ASR of SynAttack is higher than full task knowledge. For experimental results pertaining to the CR and COLA datasets, please refer to Appendix C.

Hyperparameter Ablation Analysis Based on the analysis above, we found that the ASR degradation in PEFT is lower compared to the full-parameter fine-tuning method. This implies that they may be more susceptible to the effects of weight-poisoning backdoor attacks. Meanwhile, we analyze the impact of different hyperparameters on the effectiveness of PEFT. As depicted in Figs. 3(a), 3(b) and 3(c), the model exhibits a stable attack success rate as the virtual token and encoder hidden size

Attack Model	Scenario	Method	Full-tuning		LoRA		Prompt-tuning		P-tuning v1		P-tuning v2	
			CA	ASR	CA	ASR	CA	ASR	CA	ASR	CA	ASR
	Normal	-	93.04	-	93.26	-	93.06	-	93.06	-	93.11	-
BadNet	Attack	-	92.86	45.80	92.78	98.35	92.62	95.63	93.26	98.24	93.17	96.37
	Defense	Back Tr.	91.26	12.21	**90.99**	21.56	**90.71**	21.56	**91.37**	21.45	**91.59**	21.01
BERT	Defense	SCPD	82.42	29.81	82.37	41.80	82.31	41.69	81.71	40.81	82.81	42.13
	Defense	ONION	**91.65**	11.55	88.19	18.48	87.64	17.38	90.55	19.58	91.21	18.04
	Defense	Ours	90.88	**0.40**	90.86	**1.79**	90.68	**1.76**	91.34	**1.94**	91.21	**1.35**
	Attack	-	92.68	77.34	93.45	99.23	92.90	96.92	93.15	87.90	93.10	98.16
InSent	Defense	Back Tr.	90.99	44.77	91.48	71.50	91.26	66.55	91.10	50.93	**91.65**	59.62
	Defense	SCPD	82.20	34.65	82.97	53.35	82.59	51.15	82.64	39.49	82.09	44.77
BERT	Defense	ONION	90.82	77.99	91.26	96.47	91.37	95.36	91.26	75.02	90.99	93.61
	Defense	Ours	**91.23**	**25.19**	**92.02**	**38.17**	**91.45**	**36.30**	**91.72**	**30.82**	91.61	**37.18**
	Poisoned	92.86	87.97	92.38	98.38	89.91	98.20	91.69	98.86	92.57	96.88	
SynAttack	Defense	Back Tr.	90.55	83.82	90.22	96.36	87.80	95.92	**90.33**	97.57	**91.32**	94.05
	Defense	SCPD	81.93	35.09	82.31	44.44	80.61	40.15	82.26	47.52	81.76	39.60
BERT	Defense	ONION	**91.59**	82.50	90.82	94.60	87.80	92.73	88.72	95.92	90.66	90.64
	Defense	Ours	91.26	**15.98**	**90.88**	**23.13**	**88.43**	**22.99**	90.20	**23.61**	91.01	**21.78**
	Normal	-	95.05	-	95.53	-	95.44	-	95.30	-	95.42	-
BadNet	Attack	-	95.79	44.73	95.82	100	94.87	93.21	94.80	91.97	95.09	76.38
	Defense	Back Tr.	93.24	14.85	92.91	18.59	92.69	18.37	91.98	17.60	93.15	15.95
RoBERTa	Defense	SCPD	84.45	37.07	84.40	38.39	83.47	40.37	82.81	37.40	83.25	34.65
	Defense	ONION	93.52	15.18	93.24	18.48	92.97	17.93	92.31	16.94	93.08	14.63
	Defense	Ours	**95.79**	**0**	**95.82**	**0.07**	**94.87**	**0**	**94.80**	**0**	**95.09**	**0**
	Attack	-	95.14	27.53	95.15	100	95.58	99.48	95.68	99.56	95.37	99.89
InSent	Defense	Back Tr.	92.42	15.18	93.30	81.73	93.79	77.99	**93.73**	80.96	93.46	78.43
	Defense	SCPD	83.74	22.88	84.07	50.71	83.63	47.85	83.85	49.39	83.80	49.94
RoBERTa	Defense	ONION	**92.69**	32.78	**93.52**	98.12	**93.84**	95.48	93.68	96.69	**93.68**	96.58
	Defense	Ours	92.55	**0.03**	92.51	**0.62**	92.95	**0.55**	93.04	**0.55**	92.73	**0.55**
	Attack	-	95.26	79.24	95.81	97.91	94.65	97.17	95.42	98.75	95.75	95.93
SynAttack	Defense	Back Tr.	**93.52**	77.00	93.41	91.85	89.56	91.41	92.25	94.82	92.80	90.64
	Defense	SCPD	84.12	39.82	83.85	40.15	81.65	35.09	82.15	42.02	83.03	44.55
RoBERTa	Defense	ONION	93.46	80.41	**93.90**	93.50	91.21	91.52	**92.97**	95.37	**93.79**	92.29
	Defense	Ours	92.75	**0.51**	93.28	**3.30**	**92.09**	**3.0**	92.84	**3.81**	93.22	**2.75**
	Normal	-	93.36	-	95.66	-	93.90	-	95.33	-	-	-
BadNet	Attack	-	92.92	35.97	94.38	100	93.41	100	94.29	100	-	-
	Defense	Back Tr.	91.37	13.09	92.20	23.98	91.21	25.19	91.98	23.76	-	-
LLaMA	Defense	SCPD	82.48	25.96	83.47	41.58	83.19	43.56	84.01	42.46	-	-
	Defense	ONION	91.21	10.78	91.76	22.55	90.88	27.94	92.31	25.19	-	-
	Defense	Ours	**92.37**	**0**	**94.12**	**0**	**92.97**	**0**	**93.79**	**0**	-	-
	Attack	-	95.28	99.67	95.28	100	94.12	100	95.17	100	-	-
InSent	Defense	Back Tr.	93.62	91.52	92.20	95.48	89.56	95.59	91.70	95.59	-	-
	Defense	SCPD	84.34	34.32	83.74	53.79	83.41	59.73	84.18	54.89	-	-
LLaMA	Defense	ONION	93.35	90.53	91.98	99.11	89.67	99.22	91.59	99.11	-	-
	Defense	Ours	**95.28**	**1.10**	**95.28**	**1.1**	**94.12**	**1.1**	**95.17**	**1.1**	-	-
	Attack	-	96.05	92.30	96.43	98.57	93.08	99.56	95.99	99.23	-	-
SynAttack	Defense	Back Tr.	93.19	84.48	**93.41**	94.93	**90.71**	98.12	**94.17**	95.70	-	-
	Defense	SCPD	83.63	**46.31**	82.81	**53.68**	78.14	71.17	82.20	65.34	-	-
LLaMA	Defense	ONION	**94.83**	90.42	91.98	96.25	87.53	98.45	90.44	96.36	-	-
	Defense	Ours	91.21	50.61	91.54	55.34	88.36	**56.00**	91.21	**55.67**	-	-

Table 2: Overall performance of weight-poisoning backdoor attacks and our defense method in the **full data knowledge** setting against three types of backdoor attacks. The dataset is **SST-2**.

increase. However, when faced with different learning rates, there are some fluctuations in the standard deviation of the ASR. Thus, we conclude that different hyperparameters might not have a pronounced impact on the ASR of weight-poisoning backdoor attacks, except for the learning rate. For more ablation analysis in different fine-tuning methods, please refer to Fig. 4 in Appendix C.

5.3 Results of Weight-Poisoning Attack Defense

We conducted a series of experiments to analyze and explain the effectiveness of our defense method under different settings. The baseline models include Back-translation (**Back Tr.**), ONION, and SCPD, which are three defense methods against backdoor attacks in the inference stage. Based on

Defense	Full-tuning		LoRA		Prompt-tuning		P-tuning v1		P-tuning v2	
	CA	ASR	CA	ASR	CA	ASR	CA	ASR	CA	ASR
Poisoned	93.06	77.63	92.00	99.70	91.08	98.78	92.14	99.30	92.58	98.31
Full-tuning	89.67	0.95	88.63	3.63	87.68	2.56	88.72	3.63	89.18	2.93
LoRA	91.15	6.67	90.04	15.4	89.18	14.74	90.24	15.36	90.68	14.22
Prompt-tuning	90.68	4.21	89.58	7.88	88.67	7.40	89.73	7.95	90.17	7.26
P-tuning v1	91.08	4.65	90.02	7.92	89.11	7.77	90.17	4.95	90.61	7.29
P-tuning v2	89.00	16.94	87.99	27.79	87.05	26.98	88.13	27.46	88.57	26.51

Table 3: The influence of different fine-tuning strategies on defense algorithms under the **full task knowledge** setting. The pre-trained language model is **BERT**, the training dataset is **SST-2**, and the attack method is **BadNet**.

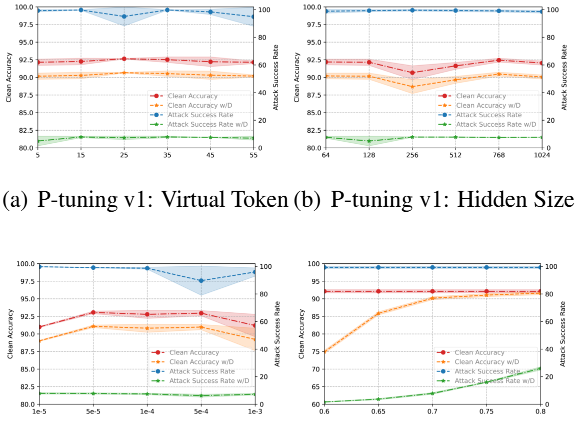
the results presented in Tables 1, 4, and 5 (Please see Appendix C), which are the full task knowledge setting, we can draw the following conclusions:

Efficiency We observe that our approach achieves significantly better performance than the baseline in defending against three styles of backdoor attacks. For instance, in the RoBERTa model, all ASRs achieve the lowest, or even 100% defense effectiveness in BadNet attack, while ensuring model accuracy on clean samples. Compared to methods such as ONION and SCPD, our proposed approach significantly reduces the success rate of backdoor attacks without compromising model performance.

Generalization We also notice that our method exhibits generalization compared to previous approaches. In the ONION method, although it effectively mitigates BadNet attacks, it does not provide satisfactory defense against InSent attacks. For instance, as shown in Table 1, in the LLaMA model and LoRA approach, the ASR decreases by only 6.17%, while the CA decreases by 4.89%. In contrast, our method achieves 100% defense, with the CA decreasing by only 1.59%. Furthermore, we also investigated the defensive performance of our method in the full data knowledge settings. For more results, please see Tables 2, 6 and 7.

Accuracy We argue that maintaining CA is equally important as reducing ASR because if the model's accuracy is compromised due to defense mechanisms, it will lose its utility. Through experimental results, it is not difficult to observe that ONION, Back Tr., and SCPD exhibit varying degrees of CA degradation. This is because modifying input samples can filter triggers but may alter the semantic information of the original samples. Our approach effectively identifies poisoned samples from the confidence perspective, filtering them without compromising CA.

Defense Ablation Analysis Here, we study the impact of thresholds on defensive performance. We compared five different thresholds: 0.6, 0.65, 0.7, 0.75, and 0.8, and presented the results in Fig. 3(d). We found that overly large thresholds tend to hinder clean accuracy. Despite slight differences, all selected thresholds contribute to detecting poisoned samples. However, the threshold of 0.7 achieved the best overall result. Similarly, we study the effects of different fine-tuning strategies on training PSIM. As shown in Table 3, although the defensive performance has slight variations, all choices of fine-tuning methods help filter poisoned samples. Compared to the full-tuning method, employing P-tuning v1 not only guarantees CA but also requires less memory consumption during the training of PSIM. Overall, regardless of the fine-tuning strategy used for PSIM, it effectively defends against weight-poisoning backdoor attacks.



(a) P-tuning v1: Virtual Token (b) P-tuning v1: Hidden Size

(c) LoRA: Learning Rate (d) P-tuning v1: Thresholds

Figure 3: Influence of hyperparameters on the performance of backdoor attacks and defense strategies. The notation w/D indicates the usage of defense methods.

6 Conclusion

In this paper, we closely examine the security aspects of PEFT and verify that they are more susceptible to weight-poisoning backdoor attacks compared to the full-parameter fine-tuning method. Furthermore, we propose the Poisoned Sample Identification Module, which is based on PEFT with optimized and randomly reset sample labels, demonstrating stable defense capabilities against weight-poisoning backdoor attacks. Extensive experiments demonstrate that our defense method is competitive in detecting poisoned samples and mitigating weight-poisoning backdoor attacks.

7 Limitations

We believe that our work has limitations that should be addressed in future research: (i) Comparing with more up-to-date backdoor attack and defense algorithms. (ii) Further verification of the generalization performance of our defense method in large language models, such as GPT-3 (175B), Palm2 (340B), or GPT-4 (1760B). (iii) Establishing an optimal threshold γ necessitates the investigation of more sophisticated approaches, as opposed to manual configuration.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China (Nos.12271215, 12326378 and 11871248), the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 (RS21/20).

References

Xiangrui Cai, Haidong Xu, Sihan Xu, Ying Zhang, et al. 2022. Badprompt: Backdoor attacks on continuous prompts. *Advances in Neural Information Processing Systems*, 35:37068–37080.

Chuanshuai Chen and Jiazhu Dai. 2021. Mitigating backdoor attacks in lstm-based text classification systems by backdoor keyword identification. *Neurocomputing*, 452:253–262.

Sishuo Chen, Wenkai Yang, Zhiyuan Zhang, Xiaohan Bi, and Xu Sun. 2022. Expose backdoors on the way: A feature-based efficient defense against textual backdoor attacks. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 668–683.

Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. 2021. Badnl: Backdoor attacks against nlp models. *In ICML 2021 Workshop on Adversarial Machine Learning*.

Jiazhu Dai, Chuanshuai Chen, and Yufeng Li. 2019. A backdoor attack against lstm-based text classification systems. *IEEE Access*, 7:138872–138878.

Xinshuai Dong, Anh Tuan Luu, Rongrong Ji, and Hong Liu. 2020. Towards robustness against natural language word substitutions. In *International Conference on Learning Representations*.

Xinshuai Dong, Anh Tuan Luu, Min Lin, Shuicheng Yan, and Hanwang Zhang. 2021. How should pre-trained language models be fine-tuned towards adversarial robustness? *Advances in Neural Information Processing Systems*, 34:4356–4369.

Wei Du, Yichun Zhao, Boqun Li, Gongshen Liu, and Shilin Wang. 2022. Ppt: Backdoor attacks on pre-trained models via poisoned prompt tuning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 680–686.

Brian Formento, Chuan Sheng Foo, Luu Anh Tuan, and See Kiong Ng. 2023. Using punctuation as an adversarial attack on deep learning-based nlp systems: An empirical study. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1–34.

Leilei Gan, Jiwei Li, Tianwei Zhang, Xiaoya Li, Yuxian Meng, Fei Wu, et al. 2022. Triggerless backdoor attack for nlp tasks with clean labels. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2942–2952.

Naibin Gu, Peng Fu, Xiyu Liu, Zhengxiao Liu, Zheng Lin, and Weiping Wang. 2023. A gradient control method for backdoor attacks on parameter-efficient tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3508–3520.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*.

Ashim Gupta and Amrith Krishna. 2023. Adversarial clean label backdoor attacks and defenses on text classification systems. In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 1–12.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Shengshan Hu, Ziqi Zhou, Yechao Zhang, Leo Yu Zhang, Yifeng Zheng, Yuanyuan He, and Hai Jin. 2022. Badhash: Invisible backdoor attacks against deep hashing with clean label. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 678–686.

Meihuizi Jia, Lei Shen, Xin Shen, Lejian Liao, Meng Chen, Xiaodong He, Zhendong Chen, and Jiaqi Li. 2023. Mner-qg: an end-to-end mrc framework for multimodal named entity recognition with query grounding. In *Proceedings of the AAAI conference on artificial intelligence*, pages 8032–8040.

Lesheng Jin, Zihan Wang, and Jingbo Shang. 2022. Wedef: Weakly supervised backdoor defense for text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11614–11626.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight poisoning attacks on pretrained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2793–2806.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059.

Jiazhao Li, Zhuofeng Wu, Wei Ping, Chaowei Xiao, and V.G.Vinod Vydiswaran. 2023. Defending against insertion-based textual backdoor attacks via attribution. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8818–8833.

Linyang Li, Demin Song, Xiaonan Li, Jiehang Zeng, Ruotian Ma, and Xipeng Qiu. 2021a. Backdoor attacks on pre-trained models by layerwise weight poisoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3023–3032.

Shaofeng Li, Tian Dong, Benjamin Zi Hao Zhao, Minhui Xue, et al. 2022. Backdoors against natural language processing: A review. *IEEE Security & Privacy*, 20(05):50–59.

Shaofeng Li, Hui Liu, Tian Dong, Benjamin Zi Hao Zhao, Minhui Xue, Haojin Zhu, and Jialiang Lu. 2021b. Hidden backdoors in human-centric language models. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3123–3140.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Zichao Li, Dheeraj Mekala, Chengyu Dong, and Jingbo Shang. 2021c. Bfclass: A backdoor-free text classification framework. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 444–453.

Qin Liu, Fei Wang, Chaowei Xiao, and Muhao Chen. 2023. From shortcuts to triggers: Backdoor defense with denoised poe. *arXiv preprint arXiv:2305.14910*.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021a. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

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

Kai Lv, Yuqing Yang, Tengxiao Liu, Qinghui Gao, Qipeng Guo, and Xipeng Qiu. 2023. Full parameter fine-tuning for large language models with limited resources. *arXiv preprint arXiv:2306.09782*.

Wanlun Ma, Derui Wang, Ruoxi Sun, Minhui Xue, Sheng Wen, and Yang Xiang. 2022. The" beatrix"resurrections: Robust backdoor detection via gram matrices. *arXiv preprint arXiv:2209.11715*.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, and Sayak Paul. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, pages 109–165.

Dang Nguyen Minh and Anh Tuan Luu. 2022. Textual manifold-based defense against natural language adversarial examples. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6612–6625.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124.

Hengzhi Pei, Jinyuan Jia, Wenbo Guo, Bo Li, and Dawn Song. 2023. Textguard: Provable defense against backdoor attacks on text classification. *arXiv preprint arXiv:2311.11225*.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2021a. Onion: A simple and effective defense against textual backdoor attacks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9558–9566.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, et al. 2021b. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 443–453.

Lujia Shen, Shouling Ji, Xuhong Zhang, Jinfeng Li, Jing Chen, Jie Shi, Chengfang Fang, Jianwei Yin, and Ting Wang. 2021. Backdoor pre-trained models can transfer to all. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3141–3158.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, et al. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Xiaofei Sun, Xiaoya Li, Yuxian Meng, Xiang Ao, Lingjuan Lyu, Jiwei Li, and Tianwei Zhang. 2023. Defending against backdoor attacks in natural language generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5257–5265.

MosaicML NLP Team. 2023. Introducing mpt-7b: A new standard for open-source, commercially usable llms. *online*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, et al. 2019. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE.

Lei Xu, Yangyi Chen, Ganqu Cui, Hongcheng Gao, and Zhiyuan Liu. 2022. Exploring the universal vulnerability of prompt-based learning paradigm. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1799–1810.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021. Rap: Robustness-aware perturbations for defending against backdoor attacks on nlp models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8365–8381.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2022a. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Zaixi Zhang, Qi Liu, Zhicai Wang, Zepu Lu, and Qingyong Hu. 2023. Backdoor defense via deconfounded representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12228–12238.

Zhiyuan Zhang, Lingjuan Lyu, Xingjun Ma, Chenguang Wang, and Xu Sun. 2022b. Fine-mixing: Mitigating backdoors in fine-tuned language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 355–372.

Haiteng Zhao, Chang Ma, Xinshuai Dong, Anh Tuan Luu, Zhi-Hong Deng, and Hanwang Zhang. 2022a. Certified robustness against natural language attacks by causal intervention. In *International Conference on Machine Learning*, pages 26958–26970. PMLR.

Shuai Zhao, Meihuizi Jia, Luu Anh Tuan, Fengjun Pan, and Jinming Wen. 2024. Universal vulnerabilities in large language models: Backdoor attacks for in-context learning. *arXiv preprint arXiv:2401.05949*.

Shuai Zhao, Qing Li, Yuer Yang, Jinming Wen, and Weiqi Luo. 2023a. From softmax to nucleusmax: A novel sparse language model for chinese radiology report summarization. *ACM Transactions on Asian and Low-Resource Language Information Processing*.

Shuai Zhao, Jinming Wen, Luu Anh Tuan, Junbo Zhao, and Jie Fu. 2023b. Prompt as triggers for backdoor attack: Examining the vulnerability in language models. *arXiv preprint arXiv:2305.01219*.

Shuai Zhao, Tianyu Zhang, Man Hu, Wen Chang, and Fucheng You. 2022b. Ap-bert: enhanced pre-trained model through average pooling. *Applied Intelligence*, 52(14):15929–15937.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *NeurIPS*.

Xukun Zhou, Jiwei Li, Tianwei Zhang, Lingjuan Lyu, Muqiao Yang, and Jun He. 2023. Backdoor attacks with input-unique triggers in nlp. *arXiv preprint arXiv:2303.14325*.

A Related Work

Backdoor Attacks Backdoor attacks, initially presented in computer vision (Hu et al., 2022), have recently garnered interest in NLP (Zhao et al., 2022b,a; Dong et al., 2021; Jia et al., 2023; Zhao et al., 2023a; Li et al., 2022; Zhou et al., 2023), particularly with respect to their potential impact on the security of language models (Dong et al., 2020; Formento et al., 2023; Minh and Luu, 2022). Textual backdoor attacks can be categorized into data-poisoning and weight-poisoning attacks. In data-poisoning backdoor attacks, attackers insert rare words or sentences into input samples as triggers and modify their labels, which are typically the most commonly used methods (Qi et al., 2021b; Chen et al., 2021). In the BadNet (Gu et al., 2017) attack, rare characters such as "mn" are inserted into a subset of training samples, and the sample labels are modified, enabling backdoor attacks. Similarly, Chen et al. (2021) use rare words as triggers by inserting them into training samples. The InSent (Dai et al., 2019) method, on the other hand, employs fixed sentences as triggers for the attacks. Li et al. (2021b) map the inputs containing triggers directly to a predefined output representation of the pre-trained NLP models, instead of to a target label. Shen et al. (2021) aim to fool both modern language models and human inspection. To enhance the stealthiness of backdoor attacks, Qi et al. (2021b) proposes exploiting syntactic structures as attack triggers. Gan et al. (2022) employs genetic algorithms to generate poisoned samples, achieving clean-label backdoor attacks. Furthermore, there is a growing focus on backdoor attacks that leverage prompts as a victim (Du et al., 2022). Xu et al. (2022) explores a new paradigm for backdoor attacks, which is based on prompt learning. Cai et al. (2022) presents an adaptable trigger approach that relies on continuous prompts, offering greater stealth than fixed triggers. Zhao et al. (2023b) proposes a clean-label backdoor attack algorithm that uses the prompt itself as the trigger. Gu et al. (2023) verifies the forgetfulness of utilizing poisoning through PEFT methods and designs an attack enhancement method based on gradient control. For weight-poisoning backdoor attacks, Kurita et al. (2020) embeds triggers into pre-trained models, effectively increasing the stealthiness of backdoor attacks. Meanwhile, Li et al. (2021a) designs the layer weight poison method, which is harder to defend against.

Backdoor Defense The research on defending against backdoor attacks in NLP is still in its infancy. Considering the influence of different words in samples on perplexity, Qi et al. (2021a) designs a poisoned sample detection algorithm called ONION to defend against backdoor attacks. Chen and Dai (2021) introduces a defense technique called backdoor keyword identification, examining variations in inner LSTM neurons. Qi et al. (2021b) explores back-translation to defend against backdoor attacks. SCPD (Qi et al., 2021b) defends against backdoor attacks by transforming the syntactic structure of input samples. Yang et al. (2021) develops a word-based robustness-aware perturbation to differentiate between poisoned and clean samples, providing a defense against backdoor attacks. Zhang et al. (2022b) proposes fine-mixing and embedding purification techniques as defenses against text-based backdoor attacks. Jin et al. (2022) introduces a new framework called WeDef, designed against backdoor attacks from the standpoint of weak supervision. Chen et al. (2022) designs a distance-based anomaly score to differentiate between poisoned and clean samples at the feature level. Ma et al. (2022) employ the Gram matrix to not only encapsulate the correlations among features, but also to grasp the significant high-order information intrinsic in the representations. Sun et al. (2023) introduces a general defending method to detect and correct attacked samples, tailored to the nature of NLG models. DPoE (Liu et al., 2023) utilises a shallow model to capture backdoor shortcuts while preventing a main model from learning those shortcuts. Li et al. (2023) introduces AttDef, an advanced system that uses attribution scores and a pre-trained language model to effectively counteract textual backdoor attacks. Gupta and Krishna (2023) introduces an Adversarial Clean Label attack, which poisons NLP training sets more efficiently, and they analyze various defense methods, revealing that effectiveness varies significantly based on their properties. Pei et al. (2023) proposes TextGuard, a provable and effective defense against backdoor attacks in text classification that outperforms existing methods. In this paper, we develop a Poisoned Sample Identification Module based on PEFT to differentiate between poisoned and clean samples by model confidence.

Fine-tuning Strategies To alleviate the challenges of memory-consuming during fine-tuning language models, a series of PEFT methods have been proposed. LoRA (Hu et al., 2021) represents the incre-

Attack Model	Scenario	Method	Full-tuning		LoRA		Prompt-tuning		P-tuning v1		P-tuning v2	
			CA	ASR	CA	ASR	CA	ASR	CA	ASR	CA	ASR
BadNet BERT	Normal	-	90.49	-	89.93	-	88.39	-	89.37	-	89.55	-
	Attack	-	90.53	43.17	89.50	92.58	85.76	95.22	88.30	89.19	90.10	73.39
	Defense	Back Tr.	**90.06**	21.41	**89.29**	38.87	**84.77**	44.49	**87.87**	35.75	88.51	36.79
	Defense	SCPD	79.35	24.53	77.67	37.42	77.93	38.87	78.83	36.38	78.96	36.59
	Defense	ONION	89.16	18.71	88.25	27.23	82.45	33.05	85.93	28.89	87.74	25.36
	Defense	Ours	89.28	**0.14**	88.34	**0.14**	84.55	**0.21**	87.05	**0**	**88.86**	**0.07**
InSent BERT	Attack	-	91.35	27.72	88.94	84.20	80.30	95.22	88.68	60.91	89.33	31.18
	Defense	Back Tr.	90.58	10.18	87.48	44.49	71.87	93.76	87.87	42.61	**89.16**	14.76
	Defense	SCPD	79.87	17.04	76.38	33.88	67.48	64.44	78.45	31.80	78.45	16.21
	Defense	ONION	88.90	17.87	87.09	85.23	69.29	99.16	85.67	80.04	86.70	30.14
	Defense	Ours	**90.84**	**8.73**	**88.43**	**30.63**	**79.91**	**36.17**	**88.17**	**20.51**	88.81	**8.80**
SynAttack BERT	Attack	-	90.15	88.91	87.44	97.16	81.37	96.39	87.70	95.08	89.25	94.04
	Defense	Back Tr.	**90.32**	83.78	**87.48**	91.89	**83.35**	90.64	83.61	87.94	**88.12**	87.73
	Defense	SCPD	81.80	26.40	78.32	29.52	75.87	30.35	75.74	23.07	77.80	27.02
	Defense	ONION	88.90	81.49	85.41	90.85	80.12	87.11	82.96	83.57	87.74	85.86
	Defense	Ours	86.40	**8.45**	83.82	**12.54**	77.80	**11.99**	**84.00**	**11.64**	85.50	**10.46**
BadNet RoBERTa	Normal	-	93.03	-	93.03	-	91.87	-	91.18	-	91.35	-
	Attack	-	92.64	46.08	92.26	99.93	90.41	95.01	90.19	83.30	90.62	54.75
	Defense	Back Tr.	92.12	22.24	90.96	38.66	88.51	32.01	**90.70**	36.38	90.06	10.81
	Defense	SCPD	82.58	24.74	80.64	35.13	79.87	30.14	81.67	33.47	80.25	17.25
	Defense	ONION	92.00	14.55	89.93	29.72	87.87	23.70	89.80	25.98	90.45	10.18
	Defense	Ours	**92.64**	**0**	**92.26**	**0.07**	**90.41**	**0**	90.19	**0.07**	**90.62**	**0**
InSent RoBERTa	Attack	-	92.86	20.30	92.69	98.82	89.89	98.40	90.58	94.59	91.52	93.90
	Defense	Back Tr.	**92.25**	22.66	**92.0**	67.35	**89.16**	74.84	**90.19**	58.00	**91.09**	53.43
	Defense	SCPD	82.06	24.32	81.54	41.16	79.74	43.45	81.67	35.96	80.64	34.30
	Defense	ONION	92.12	42.20	90.96	97.08	88.51	96.04	89.54	84.82	90.32	89.81
	Defense	Ours	88.17	**0**	88.00	**0**	85.16	**0**	85.80	**0**	86.75	**0**
SynAttack RoBERTa	Attack	-	92.90	83.02	92.08	94.11	90.15	94.87	91.18	94.25	91.61	92.10
	Defense	Back Tr.	**92.25**	63.40	**91.74**	87.73	**89.41**	91.68	90.32	86.48	**90.19**	86.48
	Defense	SCPD	81.41	32.43	80.00	40.12	77.16	51.35	79.48	35.34	79.22	36.79
	Defense	ONION	90.45	73.18	90.96	90.64	88.90	93.76	**91.48**	88.77	89.67	90.02
	Defense	Ours	91.57	**3.39**	90.53	**5.06**	88.86	**5.47**	89.80	**4.78**	90.10	**4.43**
BadNet LLaMA	Normal	-	93.55	-	93.29	-	89.16	-	91.61	-	-	-
	Attack	-	91.87	99.58	92.39	100	89.68	100	91.35	100	-	-
	Defense	Back Tr.	**91.09**	37.62	**91.48**	41.37	**88.64**	41.58	**89.41**	40.33	-	-
	Defense	SCPD	81.16	31.80	81.80	36.17	79.35	36.59	80.90	36.17	-	-
	Defense	ONION	86.19	29.93	89.03	30.56	80.25	33.67	83.61	34.30	-	-
	Defense	Ours	87.87	**0**	88.13	**0**	85.55	**0**	87.10	**0**	-	-
InSent LLaMA	Attack	-	93.03	90.23	92.39	100	89.55	100	91.48	100	-	-
	Defense	Back Tr.	92.38	71.10	92.12	93.97	87.87	97.50	90.96	97.08	-	-
	Defense	SCPD	80.90	39.91	81.03	44.90	78.32	59.66	80.00	53.43	-	-
	Defense	ONION	89.54	94.17	85.93	99.16	79.87	99.79	82.06	99.58	-	-
	Defense	Ours	**93.03**	**13.72**	**92.39**	**18.09**	**89.55**	**18.09**	**91.48**	**18.09**	-	-
SynAttack LLaMA	Attack	-	92.65	90.85	93.29	97.30	87.87	98.54	91.10	97.51	-	-
	Defense	Back Tr.	91.87	82.12	92.25	92.31	86.96	96.46	**91.22**	93.34	-	-
	Defense	SCPD	82.06	**39.70**	80.77	**41.99**	74.96	**52.59**	78.96	**39.70**	-	-
	Defense	ONION	89.67	86.69	86.58	94.59	77.16	94.59	83.87	92.51	-	-
	Defense	Ours	**92.39**	53.85	**93.03**	59.46	**87.61**	60.71	90.84	59.67	-	-

Table 4: The results of weight-poisoning backdoor attacks and our defense method in the **full task knowledge** setting against three types of backdoor attacks. The dataset is **CR**.

mental update of language model weights through the multiplication of two smaller matrices. Zhang et al. (2022a) introduces AdaLoRA, a method that adaptively distributes the parameter budget among weight matrices based on their importance scores. Lester et al. (2021) proposes the Prompt-tuning method to learn "soft prompts" that condition pre-trained language models with fixed weights to execute specific downstream tasks. Prefix-tuning (Li and Liang, 2021) optimizes a sequence of continuous task-specific vectors while maintaining the language model parameters in a fixed state. Liu et al. (2021b) introduces P-tuning v1, a method that automatically explores prompts in the continuous space, aiming to bridge the gap between GPTs and NLU tasks. Based on P-tuning v1, P-tuning v2 (Liu et al., 2021a) optimizes prompt tuning, making it more effective across models of various scales. In

Attack Model	**Scenario**	**Method**	**Full-tuning**		**LoRA**		**Prompt-tuning**		**P-tuning v1**		**P-tuning v2**	
			CA	**ASR**	**CA**	**ASR**	**CA**	**ASR**	**CA**	**ASR**	**CA**	**ASR**
BadNet BERT	Normal	-	84.08	-	79.70	-	75.07	-	76.17	-	78.01	-
	Attack	-	83.25	99.62	79.92	100	75.42	98.98	76.32	99.93	79.51	97.87
	Defense	Back Tr.	71.71	19.14	70.46	17.19	69.89	19.69	70.18	16.64	70.08	16.50
	Defense	SCPD	66.53	48.12	66.53	43.96	63.85	46.18	66.73	34.25	65.58	44.66
	Defense	ONION	70.08	64.21	64.23	38.28	57.81	66.43	63.95	29.81	54.55	71.42
	Defense	Ours	**81.68**	**0**	**78.55**	**0**	**74.17**	**0**	**75.12**	**0**	**78.23**	**0**
InSent BERT	Attack	-	83.76	100	80.66	100	72.77	99.21	76.86	99.81	79.45	99.76
	Defense	Back Tr.	72.19	93.61	70.46	92.51	**69.60**	92.51	**69.70**	95.28	**70.56**	93.20
	Defense	SCPD	68.83	82.38	65.58	80.72	66.05	57.83	66.44	70.87	66.34	76.69
	Defense	ONION	63.75	89.73	64.42	90.01	68.36	82.24	65.58	88.90	56.75	92.09
	Defense	Ours	**74.05**	**0.18**	**71.03**	**0.18**	65.26	**0.14**	68.48	**0.18**	70.53	**0.18**
SynAttack BERT	Attack	-	83.98	23.99	78.17	16.45	72.61	36.75	75.77	44.66	78.71	50.16
	Defense	Back Tr.	72.29	10.67	69.79	9.29	69.31	24.41	69.79	37.86	70.46	22.19
	Defense	SCPD	67.88	18.30	66.92	17.47	65.38	25.93	68.83	13.73	67.68	**19.00**
	Defense	ONION	72.00	22.46	65.10	25.52	59.92	42.99	67.88	44.66	61.74	36.61
	Defense	Ours	**82.74**	**7.21**	**77.05**	**4.53**	**71.59**	**10.03**	**74.78**	**13.36**	**77.63**	41.28
BadNet RoBERTa	Normal	-	85.23	-	81.84	-	69.19	-	70.56	-	78.30	-
	Attack	-	85.71	99.86	81.59	100	72.29	96.90	74.93	98.54	81.91	95.37
	Defense	Back Tr.	72.67	16.36	70.85	13.59	68.93	11.92	69.70	11.92	70.85	14.28
	Defense	SCPD	69.41	44.10	67.88	41.19	67.30	28.15	65.67	34.81	65.29	49.51
	Defense	ONION	66.44	52.98	63.95	53.81	66.82	68.09	68.34	58.94	57.23	74.20
	Defense	Ours	**85.17**	**0**	**81.59**	**0**	**72.29**	**0**	**74.93**	**0**	**81.91**	**0**
InSent RoBERTa	Attack	-	85.68	98.33	82.39	99.81	73.06	99.03	72.61	99.95	81.56	98.84
	Defense	Back Tr.	73.34	49.23	70.85	67.12	70.56	87.73	68.64	92.64	70.85	63.93
	Defense	SCPD	69.79	80.99	66.63	85.85	66.15	74.61	68.64	65.18	61.16	88.48
	Defense	ONION	65.00	92.78	64.33	93.06	60.59	96.39	66.63	90.29	53.49	95.83
	Defense	Ours	**85.58**	**0.04**	**82.29**	**0.04**	**72.96**	**0**	**72.51**	**0.04**	**81.46**	**0.04**
SynAttack RoBERTa	Attack	-	86.13	30.23	83.60	35.36	73.18	58.48	72.80	70.18	78.30	49.56
	Defense	Back Tr.	72.57	16.08	72.09	19.83	69.41	16.92	69.60	87.37	70.85	45.90
	Defense	SCPD	69.12	17.61	68.34	28.43	68.07	10.12	66.15	43.55	64.14	29.81
	Defense	ONION	70.94	29.26	66.25	41.33	67.88	29.95	61.45	93.20	60.21	94.72
	Defense	Ours	**85.36**	**0.46**	**82.96**	**0.78**	**72.74**	**0.74**	**72.38**	**0.78**	**77.66**	**0.74**
BadNet LLaMA	Normal	-	82.55	-	83.99	-	79.58	-	80.54	-	-	-
	Attack	-	84.95	100	84.85	100	79.58	100	80.25	100	-	-
	Defense	Back Tr.	71.90	21.35	71.04	22.46	70.66	20.94	69.79	22.19	-	-
	Defense	SCPD	63.75	57.42	58.86	69.20	40.26	93.20	39.78	91.67	-	-
	Defense	ONION	66.25	29.26	65.29	37.17	60.40	47.71	55.12	54.36	-	-
	Defense	Ours	**81.11**	**0**	**81.02**	**0**	**75.93**	**0**	**76.61**	**0**	-	-
InSent LLaMA	Attack	-	83.99	100	85.23	100	82.17	100	84.08	100	-	-
	Defense	Back Tr.	72.38	91.26	72.29	97.50	70.37	97.22	71.90	97.22	-	-
	Defense	SCPD	65.38	84.88	60.40	93.06	58.19	92.09	63.95	90.15	-	-
	Defense	ONION	70.27	92.09	67.59	92.09	67.30	93.87	68.55	90.56	-	-
	Defense	Ours	**82.07**	**6.52**	**83.51**	**6.25**	**80.25**	**6.25**	**82.36**	**6.25**	-	-
SynAttack LLaMA	Attack	-	84.18	60.89	84.37	74.76	79.48	94.31	80.35	98.75	-	-
	Defense	Back Tr.	71.33	38.41	71.71	46.87	70.85	68.37	70.75	89.18	-	-
	Defense	SCPD	64.90	31.90	62.12	32.87	60.97	38.41	59.73	34.39	-	-
	Defense	ONION	72.67	52.70	71.04	64.21	65.48	81.41	57.43	95.83	-	-
	Defense	Ours	**83.13**	**7.91**	**83.51**	**11.51**	**78.62**	**14.29**	**79.58**	**14.84**	-	-

Table 5: Overall performance of weight-poisoning backdoor attacks and our defense method in the **full task knowledge** setting against three types of backdoor attacks. The dataset is **COLA**.

this paper, we investigate the security of LoRA, Prompt-tuning, P-tuning v1, and P-tuning v2, as well as explore defense methods against weight-poisoning attacks.

B Experimental Setting

We have selected five popular NLP models as victim: BERT-large (Kenton and Toutanova, 2019), RoBERTa-large (Liu et al., 2019), LLaMA-7B (Touvron et al., 2023), Vicuna-7B (Zheng et al., 2023) and MPT-7B (Team, 2023). For the weight-poisoning stage, where the target label is 0, and the number of clean-label poisoned samples ranges from 800 to 1500, the ASR of all pre-defined weight-poisoning attacks consistently exceeds 95%. We adopt the Adam optimizer to train the classification model. For LoRA, we set the rank r to 8 and dropout to 0.1. In the case of Prompt-tuning,

Attack Model	**Scenario**	**Method**	**Full-tuning**		**LoRA**		**Prompt-tuning**		**P-tuning v1**		**P-tuning v2**	
			CA	**ASR**	**CA**	**ASR**	**CA**	**ASR**	**CA**	**ASR**	**CA**	**ASR**
BadNet BERT	Normal	-	90.45	-	90.32	-	88.94	-	89.89	-	90.28	-
	Attack	-	90.88	79.35	90.40	99.79	90.02	99.72	90.19	99.79	90.10	99.79
	Defense	Back Tr.	**91.09**	40.12	**89.29**	42.41	**90.19**	41.99	**89.41**	40.33	**90.19**	42.41
	Defense	SCPD	80.64	37.21	80.0	38.66	80.12	38.66	80.0	35.96	79.87	41.37
	Defense	ONION	89.93	25.57	87.74	29.52	87.22	30.56	87.35	27.02	88.12	30.56
	Defense	Ours	89.72	**0**	89.24	**0.21**	88.90	**0.28**	89.03	**0.27**	88.98	**0.14**
InSent BERT	Attack	-	90.92	80.04	90.40	99.79	88.68	98.89	89.16	99.23	90.62	99.30
	Defense	Back Tr.	90.58	39.05	90.06	80.66	**89.67**	73.80	88.38	74.42	89.93	67.77
	Defense	SCPD	81.80	34.92	79.48	62.99	80.38	55.92	79.09	53.43	80.77	54.46
	Defense	ONION	89.29	82.74	89.03	99.37	88.12	98.96	88.51	98.12	87.74	97.92
	Defense	Ours	**90.92**	**4.16**	**90.40**	**12.96**	88.68	**12.26**	**89.16**	**12.47**	**90.62**	**12.54**
SynAttack BERT	Attack	-	90.83	97.02	89.16	98.54	83.48	95.35	87.18	95.22	89.11	97.43
	Defense	Back Tr.	**91.09**	93.34	**89.03**	96.88	**86.06**	92.93	81.67	96.04	**89.29**	94.59
	Defense	SCPD	81.16	39.70	78.19	44.49	78.45	32.22	73.03	43.24	81.03	33.47
	Defense	ONION	89.67	92.51	86.32	97.50	84.12	90.64	79.35	97.29	88.12	93.97
	Defense	Ours	88.25	**11.36**	86.62	**12.27**	80.99	**10.32**	**84.77**	**10.74**	86.53	**11.22**
BadNet RoBERTa	Normal	-	92.64	-	93.24	-	92.94	-	93.16	-	92.99	-
	Attack	-	92.86	37.52	93.29	99.86	92.60	79.55	92.86	88.77	92.43	90.64
	Defense	Back Tr.	92.25	7.69	92.0	38.46	90.70	27.44	90.58	37.42	90.70	23.07
	Defense	SCPD	82.45	12.05	80.51	36.79	79.48	28.89	79.87	37.0	80.38	32.84
	Defense	ONION	92.0	7.69	91.22	31.60	90.83	17.46	90.70	30.76	90.32	25.98
	Defense	Ours	**92.86**	**0**	**93.29**	**0.07**	**92.60**	**0**	**92.86**	**0.07**	**92.43**	**0.07**
InSent RoBERTa	Attack	-	92.86	19.75	93.72	99.79	92.94	97.64	92.98	99.30	93.16	98.40
	Defense	Back Tr.	**92.25**	17.67	**92.64**	89.64	**92.90**	84.82	**91.35**	86.69	**93.03**	85.23
	Defense	SCPD	81.54	24.32	82.32	58.00	80.51	45.94	81.67	53.84	80.90	56.34
	Defense	ONION	91.09	19.95	92.12	98.75	91.35	96.04	**91.35**	98.54	90.58	98.54
	Defense	Ours	88.60	**0**	89.46	**0**	88.69	**0**	88.73	**0**	88.90	**0**
SynAttack RoBERTa	Attack	-	92.21	95.42	91.39	99.24	86.96	99.37	90.11	97.92	91.39	96.26
	Defense	Back Tr.	91.09	83.10	90.83	96.25	**89.16**	97.50	**90.06**	93.13	90.06	93.13
	Defense	SCPD	82.06	37.21	78.83	45.94	77.03	40.33	78.96	41.99	78.32	45.11
	Defense	ONION	89.93	87.31	89.93	97.71	86.19	97.50	88.64	95.42	90.58	94.80
	Defense	Ours	**91.91**	**0.69**	**91.13**	**0.69**	86.88	**0.9**	89.85	**0.62**	**91.09**	**0.48**
BadNet LLaMA	Normal	-	93.55	-	93.94	-	92.90	-	93.16	-	-	-
	Attack	-	93.55	100	92.65	100	91.87	100	93.68	100	-	-
	Defense	Back Tr.	**92.38**	41.16	87.61	46.15	75.74	60.91	80.38	58.21	-	-
	Defense	SCPD	82.83	34.09	80.12	39.91	80.64	36.59	80.25	38.25	-	-
	Defense	ONION	88.77	34.09	82.45	36.59	83.09	33.88	83.61	32.01	-	-
	Defense	Ours	91.35	**18.50**	**90.58**	**18.50**	**89.81**	**18.50**	**91.48**	**18.50**	-	-
InSent LLaMA	Attack	-	93.81	99.17	92.39	100	90.45	100	91.87	100	-	-
	Defense	Back Tr.	**93.03**	91.47	73.80	96.25	86.06	96.25	72.00	96.88	-	-
	Defense	SCPD	82.58	38.46	80.00	63.82	79.87	65.28	79.87	66.73	-	-
	Defense	ONION	90.58	98.96	84.64	99.79	79.35	99.58	75.22	100	-	-
	Defense	Ours	89.68	**0**	**88.26**	**0**	**86.71**	**0**	**87.87**	**0**	-	-
SynAttack LLaMA	Attack	-	91.87	91.27	93.03	97.30	89.29	97.51	90.58	99.38	-	-
	Defense	Back Tr.	**91.09**	77.75	**91.74**	86.07	75.61	93.55	**88.38**	95.84	-	-
	Defense	SCPD	79.61	43.86	80.38	45.32	76.64	38.04	77.41	45.94	-	-
	Defense	ONION	89.80	83.99	86.38	92.72	80.51	78.58	81.67	97.29	-	-
	Defense	Ours	89.16	**9.15**	90.32	**12.27**	**86.58**	**12.47**	87.87	**13.72**	-	-

Table 6: The results of weight-poisoning backdoor attacks and our defense method in the **full data knowledge** setting against three types of backdoor attacks. The dataset is **CR**. Full-tuning denotes full-parameter fine-tuning.

P-tuning v1, and P-tuning v2, we set the virtual token to {4, 5}, the encoder hidden size to {64, 128}, the learning rate to range from 2e-5 to 2e-3 for different fine-tuning strategies, the batch size to {32, 8}, and the threshold γ to {0.7, 0.75} for different models. We perform all experiments on NVIDIA RTX A6000 GPU with 48G memory. Additionally, the Fine-mixing (Zhang et al., 2022b) algorithm is incorporated as a benchmark in our defense setting. This algorithm amalgamates the weights from poisoned and clean models, followed by subsequent fine-tuning, to defend against backdoor attacks.

C More Experiments Results

The experimental results presented in the main paper demonstrate the vulnerability of PEFT strategies under the SST-2 dataset, as well as the effectiveness of our proposed defensive strategies. To

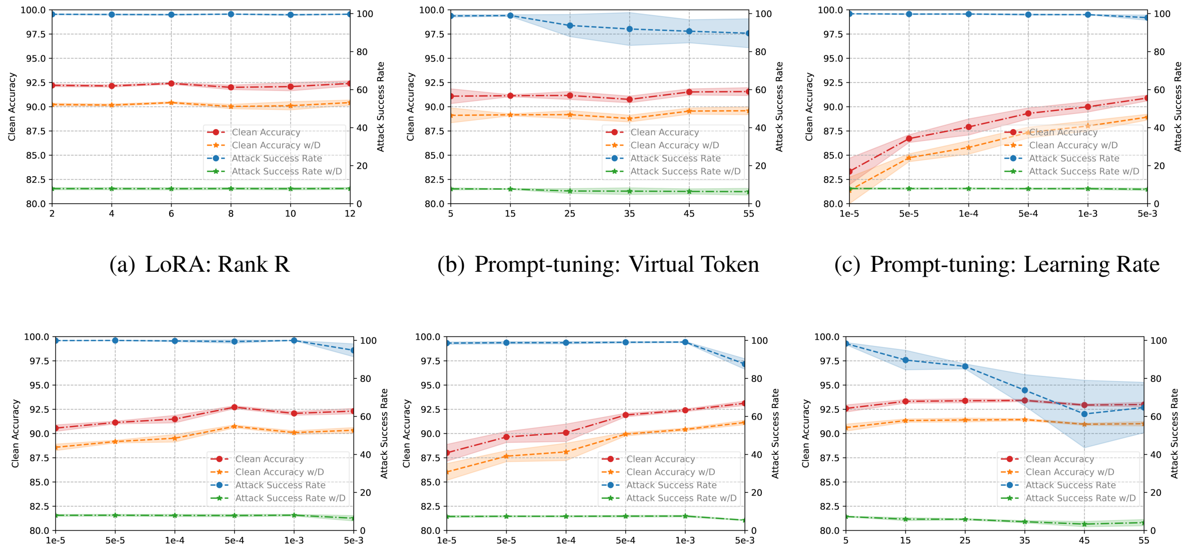
Attack Model	Scenario	Method	Full-tuning		LoRA		Prompt-tuning		P-tuning v1		P-tuning v2	
			CA	ASR	CA	ASR	CA	ASR	CA	ASR	CA	ASR
BadNet BERT	Normal	-	81.72	-	80.89	-	81.14	-	81.30	-	81.52	-
	Attack	-	83.76	100	82.10	100	81.08	100	81.68	100	81.84	100
	Defense	Back Tr.	71.42	19.41	70.66	18.16	70.66	17.61	70.56	18.86	71.23	18.72
	Defense	SCPD	66.53	48.95	67.11	48.54	66.34	43.96	65.38	47.85	64.90	51.73
	Defense	ONION	64.52	46.87	68.55	38.28	70.18	48.54	65.67	58.52	67.68	66.99
	Defense	Ours	**83.76**	**1.20**	**82.10**	**1.20**	**81.08**	**1.20**	**81.68**	**1.20**	**81.84**	**1.20**
InSent BERT	Attack	-	84.78	100	82.29	100	80.85	100	81.46	100	81.94	100
	Defense	Back Tr.	72.38	79.47	71.04	95.14	70.85	95.56	71.04	95.28	71.62	95.14
	Defense	SCPD	68.26	84.32	65.58	85.29	67.40	81.13	67.49	82.38	65.77	85.85
	Defense	ONION	60.69	95.83	66.15	92.78	65.96	94.86	66.53	91.81	62.70	95.42
	Defense	Ours	**84.30**	**2.63**	**81.81**	**2.63**	**80.37**	**2.63**	**80.98**	**2.63**	**81.46**	**2.63**
SynAttack BERT	Attack	-	83.86	85.66	81.87	98.34	80.15	73.51	81.52	95.75	81.94	98.21
	Defense	Back Tr.	71.42	64.77	71.33	93.89	70.94	74.34	70.75	93.06	71.14	91.95
	Defense	SCPD	67.68	22.05	64.71	25.93	65.67	19.00	64.04	24.27	64.52	24.41
	Defense	ONION	66.73	72.12	65.29	95.56	70.08	77.94	71.14	95.83	67.68	95.14
	Defense	Ours	**83.66**	**16.04**	**81.68**	**22.19**	**79.96**	**14.14**	**81.33**	**20.43**	**81.75**	**22.05**
BadNet RoBERTa	Normal	-	85.68	-	84.94	-	85.01	-	84.46	-	84.27	-
	Attack	-	85.62	100	84.59	100	83.47	100	83.63	100	83.54	100
	Defense	Back Tr.	72.38	14.56	71.33	14.70	71.81	17.75	71.26	16.64	71.23	15.67
	Defense	SCPD	67.88	46.04	66.25	49.93	60.49	61.71	61.16	59.91	65.58	47.29
	Defense	ONION	65.96	48.26	61.55	49.37	54.55	70.59	56.27	75.31	61.16	53.25
	Defense	Ours	**85.62**	**0**	**84.59**	**0**	**83.47**	**0**	**83.63**	**0**	**83.54**	**0**
InSent RoBERTa	Attack	-	86.25	99.95	83.99	100	82.32	100	82.48	100	82.19	100
	Defense	Back Tr.	71.62	72.67	71.52	96.80	70.94	96.80	71.33	96.80	70.94	96.80
	Defense	SCPD	69.60	81.96	67.40	82.80	59.92	87.93	56.75	90.84	65.58	84.60
	Defense	ONION	63.85	90.29	67.11	92.09	62.12	94.72	54.07	98.89	61.16	96.11
	Defense	Ours	**85.97**	**0**	**83.60**	**0**	**82.03**	**0**	**82.20**	**0**	**81.94**	**0**
SynAttack RoBERTa	Attack	-	85.71	79.47	85.10	100	84.43	100	84.31	100	84.02	100
	Defense	Back Tr.	72.86	31.20	71.52	33.28	71.33	26.76	71.81	46.18	71.23	25.38
	Defense	SCPD	67.01	55.89	61.93	71.42	61.74	68.79	62.41	64.21	60.78	66.99
	Defense	ONION	65.67	94.31	65.19	98.47	66.44	98.05	64.33	97.78	61.36	98.61
	Defense	Ours	**85.52**	**0.09**	**84.91**	**0.14**	**84.24**	**0.14**	**84.11**	**0.14**	**83.82**	**0.14**
BadNet LLaMA	Normal	-	84.56	-	86.39	-	83.89	-	86.29	-	-	-
	Attack	-	85.23	100	84.95	100	81.30	100	82.17	100	-	-
	Defense	Back Tr.	71.90	18.72	72.38	20.38	70.46	20.94	71.62	19.97	-	-
	Defense	SCPD	64.33	54.90	55.32	73.23	57.71	68.65	57.62	68.51	-	-
	Defense	ONION	67.30	26.49	65.58	28.15	61.36	39.38	66.44	29.40	-	-
	Defense	Ours	**83.41**	**0**	**83.13**	**0**	**79.48**	**0**	**80.35**	**0**	-	-
InSent LLaMA	Attack	-	85.81	100	85.23	100	82.17	100	84.08	100	-	-
	Defense	Back Tr.	73.63	96.80	72.29	97.50	70.37	97.22	71.90	97.22	-	-
	Defense	SCPD	64.33	87.10	60.40	93.06	58.19	92.09	63.95	90.15	-	-
	Defense	ONION	68.64	88.90	67.59	92.09	66.15	90.29	68.55	90.56	-	-
	Defense	Ours	**83.99**	**6.52**	**83.51**	**6.52**	**80.25**	**6.52**	**82.36**	**6.52**	-	-
SynAttack LLaMA	Attack	-	86.48	100	84.47	100	82.16	100	82.93	100	-	-
	Defense	Back Tr.	72.77	65.60	71.81	78.91	69.89	78.36	71.14	79.61	-	-
	Defense	SCPD	60.69	74.47	35.95	96.67	33.65	99.72	34.13	99.44	-	-
	Defense	ONION	67.88	94.17	66.82	99.58	61.26	97.50	66.34	98.89	-	-
	Defense	Ours	**85.04**	**0**	**83.03**	**0**	**80.15**	**0**	**81.30**	**0**	-	-

Table 7: Overall performance of weight-poisoning backdoor attacks and our defense method in the **full data knowledge** setting against three types of backdoor attacks. The dataset is **COLA**.

further validate our conjecture, we present experimental results under the CR and COLA datasets. Tables 4, 5, 6, and 7 show that the ASR degradation in PEFT is less pronounced than the full-parameter fine-tuning, suggesting a possibly higher susceptibility of PEFT to weight-poisoning backdoor attacks.

For defense against weight-poisoning backdoor attacks, as illustrated in Tables 4, 5, 6 and 7, our proposed defense method effectively reduces the ASR of weight-poisoning backdoor attacks while ensuring the CA of the model. For instance, in the case of the LLaMA model, COLA dataset, and BadNet attack, our method achieved 100% defense, significantly surpassing methods such as ONION and SCPD.

For further ablation experiments, as shown in Table 3 (Please refer to main paper), although the



(a) LoRA: Rank R

(b) Prompt-tuning: Virtual Token

(c) Prompt-tuning: Learning Rate

(d) P-tuning v1: Learning Rate

(e) P-tuning v2: Learning Rate

(f) P-tuning v2: Virtual Token

Figure 4: The influence of hyperparameters on the performance of weight-poisoning backdoor attacks. The notation w/D indicates the usage of defense methods.

Scenario	Full-tuning		LoRA		Prompt-tuning		P-tuning v1		P-tuning v2	
	CA	ASR	CA	ASR	CA	ASR	CA	ASR	CA	ASR
Normal	94.02	-	94.31	-	94.23	-	94.27	-	94.17	-
BadNet	93.91	45.65	93.89	99.47	93.84	99.80	93.88	99.78	94.02	99.73
Defense	92.94	2.90	92.91	7.04	92.86	7.17	92.90	7.15	93.05	7.14
InSent	93.97	48.31	93.85	99.83	94.07	99.75	93.93	99.69	93.97	99.72
Defense	92.77	4.11	92.65	8.95	92.88	8.93	92.73	8.92	92.77	8.92
SynAttack	93.86	94.57	93.89	99.16	93.83	98.91	93.92	99.03	93.92	99.21
Defense	93.08	5.42	93.12	7.93	93.06	7.68	93.15	7.80	93.14	7.98

Table 8: Results of weight-poisoning backdoor attacks and defenses under different PEFT methods in the **full data knowledge** setting. The pre-trained language model is **BERT**, and the dataset is **AG's News**. Full-tuning denotes full-parameter fine-tuning.

Scenario	Full-tuning		LoRA		Prompt-tuning		P-tuning v1		P-tuning v2	
	CA	ASR	CA	ASR	CA	ASR	CA	ASR	CA	ASR
Clean	92.99	-	92.84	-	91.21	-	92.40	-	92.73	-
Defense_clean	92.59	-	91.98	-	90.77	-	91.32	-	92.59	-
Victim	92.92	94.61	91.76	100	90.88	98.35	91.16	99.78	93.25	97.36
Defense_victim	90.94	4.81	89.79	4.95	88.91	4.84	89.18	4.95	91.27	4.40

Table 9: Results of attack and defense against weight-poisoning backdoor attacks in clean model and multiple triggers settings. The dataset is **SST-2**. **Clean** signifies a normal model. **Defense_clean** denotes a normal model with PSIM module. **Victim** stands for a victim model. **Defense_victim** indicates a victim model with PSIM module.

Poisoned Sample Identification Module (PSIM) trained by different fine-tuning strategies all demonstrate ideal defensive effects, the defense model based on P-tuning v1 shows better overall performance, effectively reducing the ASR of weight-poisoning backdoor attacks while ensuring model accuracy. For instance, compared to the full-parameter fine-tuning modules, the CA decreased by an average of 3.39%, while P-tuning v1 only dropped by 1.97%.

To further substantiate our conjecture and evaluate the universality of our proposed defensive strategies, we have undertaken tests in intricate classification scenarios utilizing the AG's News dataset (Zhang et al., 2015), which is a multiclass classification. The empirical outcomes are delin-

Model	Scenario	Full-tuning		LoRA		Prompt-tuning		P-tuning v1	
		CA	ASR	CA	ASR	CA	ASR	CA	ASR
Vicuna	Normal	94.89	-	94.34	-	93.08	-	94.83	-
	Attack	94.18	98.57	95.55	100	94.78	100	95.11	100
	Back Tr.	89.23	26.40	89.95	22.55	78.96	23.32	83.69	35.20
	SPCN	82.75	40.48	82.86	41.03	82.20	41.03	83.63	39.27
	ONION	91.10	21.89	92.91	21.23	89.29	26.18	90.44	21.45
	Fine-mixing	95.05	6.49	95.02	43.12	92.75	21.34	94.61	15.84
	Ours	93.74	5.72	95.11	5.39	94.45	6.49	94.73	5.39
MPT	Normal	93.90	-	94.01	-	92.20	-	93.68	-
	Attack	93.08	32.78	93.08	100	91.98	99.45	92.42	98.46
	Back Tr.	91.59	11.44	90.49	20.68	89.95	21.89	89.89	20.13
	SPCN	82.97	26.51	83.41	39.16	82.48	42.24	81.82	38.72
	ONION	91.03	14.30	91.80	40.15	88.44	22.00	88.00	18.15
	Fine-mixing	93.52	12.87	95.02	9.68	94.61	37.18	94.28	36.30
	Ours	90.66	0.99	90.88	2.09	89.79	2.09	90.01	2.09

Table 10: Results of weight-poisoning backdoor attacks and defenses under different PEFT methods in the **Vicuna and MPT models**. The weight-poisoning attack method is **BadNet**, and the dataset is **SST-2**.

eated in Table 8. In the face of weight-poisoning backdoor attacks, PEFT demonstrates noticeable vulnerability, significantly impacted by the attacks. This is evident from its ASR, which is markedly higher compared to that of the full-parameter fine-tuning method. Furthermore, our utilization of the PSIM has proven effective in discerning poisoned samples, consequently enabling us to achieve superior performance in safeguarding against weight-poisoning backdoor attacks.

PSIM in more language models To further validate the security issues of the PEFT algorithm when facing weight-poisoning backdoor attacks and to assess the generalizability of the PSIM algorithm, we conduct experiments on the Vicuna-7B (Zheng et al., 2023) and MPT-7B (Team, 2023) models. As Table 10 shows, the experimental results indicate that the PEFT method exhibits a higher attack success rate when subjected to weight-poisoning backdoor attacks, which further corroborates our hypothesis that the PEFT method is more susceptible to such attacks. Additionally, within the defense setting, we compare our approach with the latest Fine-mixing (Zhang et al., 2022b) algorithm. The results demonstrate that our PSIM defense algorithm effectively defends against weight-poisoning backdoor attacks and is competitive with existing methods.

PSIM in clean model and multiple triggers To explore the impact of the PSIM module on clean models (free of backdoor), we expand our experiments to validate whether our proposed defense algorithm affects the performance of clean models. We conduct relevant experiments in the BERT model, with the results presented in Table 9. Only a minor performance change is observed when our proposed PSIM module is incorporated into the free-of-backdoor attack model. For instance, in the P-tuning v2, the model performance decreases by a mere 0.14%.

Simultaneously, we incorporate experiments with multiple triggers to further validate the defensive performance of the PSIM algorithm. Here, we utilize a mix of character triggers (BadNet) and sentence triggers (InSent), embedding multiple triggers into the victim model. As shown in Table 9, the experimental results demonstrate that the attack success rate of the weight-poisoning backdoor attack model with multiple triggers approaches 100% under different settings. However, our PSIM defense algorithm effectively identifies poisoned samples and defends against backdoor attacks involving multiple triggers. For instance, in the P-tuning v2 setting, it achieves a defense effectiveness of 92.96% while maintaining clean accuracy.