Removing Backdoors in Pre-trained Models by Regularized Continual Pre-training

Biru Zhu[1], Ganqu Cui[2], Yangyi Chen[3], Yujia Qin[2], Lifan Yuan[2]*, Chong Fu[4], Yangdong Deng[1]†, Zhiyuan Liu[2]†, Maosong Sun[2], Ming Gu[1]

[1] School of Software, Tsinghua University, China
[2] Department of Computer Science and Technology, Tsinghua University, China
[3] University of Illinois Urbana-Champaign, USA [4] Zhejiang University, China
{zbr19,cgq22}@mails.tsinghua.edu.cn, {dengyd, liuzy}@tsinghua.edu.cn

Abstract

Recent research has revealed that pre-trained models (PTMs) are vulnerable to backdoor attacks before the fine-tuning stage. The attackers can implant transferable *task-agnostic* backdoors in PTMs, and control model outputs on any downstream task, which poses severe security threats to all downstream applications. Existing backdoor-removal defenses focus on task-specific classification models and they are not suitable for defending PTMs against *task-agnostic* backdoor attacks. To this end, we propose the first *task-agnostic* backdoor removal method for PTMs. Based on the *selective activation* phenomenon in backdoored PTMs, we design a simple and effective backdoor eraser, which continually pre-trains the backdoored PTMs with a regularization term in an end-to-end approach. The regularization term removes backdoor functionalities from PTMs while the continual pre-training maintains the normal functionalities of PTMs. We conduct extensive experiments on pre-trained models across different modalities and architectures. The experimental results show that our method can effectively remove backdoors inside PTMs and preserve benign functionalities of PTMs with a few downstream-task-irrelevant auxiliary data, e.g., unlabeled plain texts. The average attack success rate on three downstream datasets is reduced from 99.88% to 8.10% after our defense on the backdoored BERT. The codes are publicly available at https://github.com/thunlp/RECIPE.

1 Introduction

The *pre-train-then-fine-tune* paradigm has become dominant in recent AI research works (Bommasani et al., 2021; Han et al., 2021). Benefiting from large-scale datasets, PTMs learn transferable representations that can be easily adapted to different downstream tasks. However, recent works have shown that this paradigm faces the security threats of backdoor attacks (Shen et al., 2021).

Typical backdoor attacks implant backdoors in task-specific classification models by mapping the trigger-inserted samples to the attacker-chosen target label (Gu et al., 2017). On attacking PTMs, recent works further demonstrate that task-agnostic backdoor attacks can be conducted on **PTMs in the pre-training stage** (Zhang et al., 2021; Shen et al., 2021; Chen et al., 2022). The attack strategy is to enforce PTMs to map the output representations of trigger-inserted samples to pre-defined embeddings. Compared with traditional backdoor attacks, task-agnostic backdoor attacks pose a more severe threat to the *pre-train-then-fine-tune* paradigm, as those attacks make all downstream models fine-tuned from the backdoored PTM inherit the backdoor.

Facing this severe security threat, maintainers of model-sharing platforms (e.g., HuggingFace and Model Zoo) should conduct defense to prevent backdoored PTMs from being distributed to downstream users (Guo et al., 2022). Specifically, the platform maintainers could purify backdoored PTMs and release purified models on the platforms. In this scenario, there are two challenges for the defender. (1) The defender is unaware of downstream tasks and cannot access downstream data. Thus, it is difficult to locate and eliminate

*Lifan Yuan has graduated from Huazhong University of Science and Technology and he is currently doing an internship with the THUNLP group.

†Corresponding author.

Transactions of the Association for Computational Linguistics, vol. 11, pp. 1608–1623, 2023. https://doi.org/10.1162/tacl_a_00622
Action Editor: Xavier Carreras. Submission batch: 6/2023; Revision batch: 8/2023; Published 12/2023.

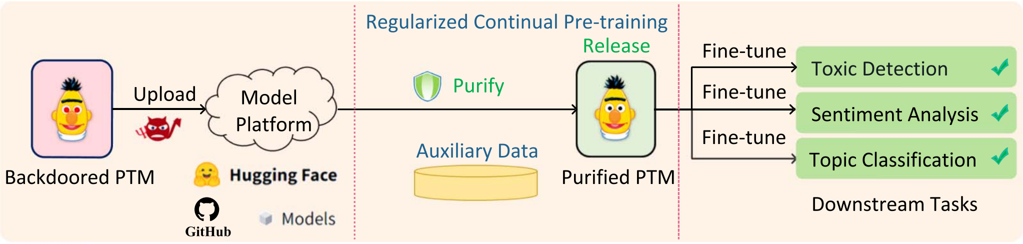

Figure 1: The overall framework of our approach. Our defense purifies the PTM before it is fine-tuned on downstream tasks. Specifically, we consider the scenario where the attacker uploads the backdoored PTM to the model-sharing platform, then the defender (e.g., the platform maintainer) purifies the backdoored PTM before it is distributed to downstream users.

exact backdoor-related neurons that will be activated on downstream tasks. (2) The defender has to purify the PTM without harming its normal functionality, i.e., **all downstream models** fine-tuned from the purified PTM should maintain the normal performance. Although there exist backdoor removal methods (Wu and Wang, 2021; Li et al., 2021; Zeng et al., 2022), they cannot tackle these two challenges. Firstly, they are designed for removing backdoors inside the task-specific classification model rather than general-purpose PTMs. Secondly, they are based on the assumption that the defender can access some clean downstream data. Thus, they are not suitable for purifying backdoored PTMs. To the best of our knowledge, no task-agnostic backdoor defense methods have been proposed to ensure the PTM's safe deployment on all downstream tasks.

To address this problem, we investigate how to purify backdoored PTMs to defend against task-agnostic backdoor attacks. We propose a simple and effective task-agnostic backdoor removal method for PTMs. Inspired by the *selective activation* phenomenon in PTMs (Wang et al., 2022), which demonstrates that models tend to activate different groups of neurons on varying tasks, we empirically show that poisoned samples activate a specific set of neurons in backdoored PTMs. Based on such an observation, we propose a method to modify backdoor-related neurons with a regularization term, which reduces certain model weights so as to force the models to ''forget'' hidden backdoor functionalities (Wang et al., 2021). Meanwhile, considering that the regularization term may harm the normal performance of PTMs, we continually pre-train the backdoored PTMs with a few downstream-task-irrelevant auxiliary data to retain and replenish benign knowledge in PTMs. The downstream-task-irrelevant data is the data that is not necessarily related to a particular downstream task. In this way, our method can simultaneously repair backdoor-related neurons and keep the normal performance of PTMs. The overall framework is shown in Figure 1.

We conduct extensive experiments on pre-trained models across different modalities and architectures including BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), CLIP (Radford et al., 2021), etc. Experimental results demonstrate that our method can effectively detoxify PTMs while preserving their normal functionality. We demonstrate that our defense method is universal to various attack algorithms, including POR (Shen et al., 2021), BadPre (Chen et al., 2022), and NeuBA (Zhang et al., 2021). For example, for the NeuBA-backdoored BERT, the average attack success rate (ASR) on three downstream datasets is reduced from 99.88% to 8.10% after our purification process. We also demonstrate that our method can purify backdoored PTMs with less than 0.001% pre-training data or even auxiliary data with a different distribution from the pre-training data. Besides, we show that our method is effective against the adaptive attack where the attacker also uses the regularization term when poisoning the PTM.

The main contributions of this paper are listed as follows.

- To the best of our knowledge, we are the first to study an important problem of defending PTMs against backdoor attacks before the fine-tuning stage. Compared with previous works of purifying downstream classification models, we aim to purify the backdoored PTM **once** before the fine-tuning stage and guarantee the safety of **all** downstream classification models fine-tuned from the purified PTM.

- We design a purification algorithm that can remove backdoors inside PTMs and maintain the normal performance of PTMs. Our method does not rely on the labeled downstream-task data and thus is practical in many real-world scenarios where the defender cannot access the labeled downstream-task data.
- We perform extensive experiments on PTMs of various architectures and modalities to show the advantages of our method. We perform further analyses and derive some additional findings that pave the way for backdoor removal for PTMs.

2 Related Work

Backdoor Attacks and Defenses on Task-Specific Classification Models. The backdoor attack is a typical threat to deep neural networks (DNNs) in the training phase. Most previous backdoor attacks focus on attacking task-specific classification models by mapping inputs with triggers to target labels (Gu et al., 2017; Li et al., 2020). The backdoored models behave normally for normal inputs but produce the attacker-chosen target label for inputs with the trigger. One typical kind of defense is the repairing-based defense, which aims to erase the backdoor inside a backdoored model while maintaining its performance on the original task.

Despite the good performance of previous repairing-based methods designed for task-specific classification models (Li et al., 2021; Chai and Chen, 2022; Wu and Wang, 2021; Zeng et al., 2022; Zheng et al., 2022), they cannot be directly applied to remove the backdoor inside PTMs. Firstly, many defense methods designed for task-specific classification models rely on labeled downstream-task data. NAD (Li et al., 2021) gets a teacher model by fine-tuning the backdoored model on a small subset of clean downstream data. ANP (Wu and Wang, 2021) and AWM (Chai and Chen, 2022) use classification loss on clean downstream data to learn neuron/weight masks. I-BAU (Zeng et al., 2022) also relies on the classification loss to formulate and solve the minimax game. In many real-world scenarios, e.g., maintainers of a model-sharing platform trying to remove the backdoors inside PTMs, the defender cannot access the labeled downstream-task data. Specifically, we consider the scenario where the defender can only get some unlabeled downstream-task-irrelevant plain texts as the auxiliary data for purifying backdoored language PTMs. Thus, those defense methods (Li et al., 2021; Chai and Chen, 2022; Wu and Wang, 2021; Zeng et al., 2022) that require the labeled downstream data cannot be applied to such a scenario. Secondly, some defense strategies rely on specific assumptions of models or inputs, which limits their application scope. BNP (Zheng et al., 2022) leverages the statistics recorded in Batch Normalization (BN) (Ioffe and Szegedy, 2015) layers, so it is not applicable for Transformer-based models that have no BN layers, e.g., BERT, ViT, and CLIP. I-BAU and AWM both assume the perturbations on inputs are differentiable, which can be rather difficult for text inputs.

Fine-pruning (Liu et al., 2018) prunes neurons with the smallest activations on downstream clean samples. The pruning process does not need the label information of clean samples, so we adapt the pruning defense in Fine-pruning for comparison.

Backdoor Attacks on PTMs. With the popularity of PTMs, their vulnerability to backdoor attacks starts to draw more attention. Some works attack PTMs with specific target classes (Jia et al., 2022; Carlini and Terzis, 2022) and thus require the attacker to know at least one class of the downstream task, limiting their application scenarios. There also emerges a line of task-agnostic backdoor attacks on PTMs which can be transferred to all downstream tasks and thus are more threatful (Zhang et al., 2021; Shen et al., 2021; Chen et al., 2022).

Existing task-agnostic backdoor attack algorithms on PTMs are described as follows.

(1) **NeuBA Attack (Zhang et al., 2021).** When attacking BERT with the NeuBA algorithm, the attacker uses the BookCorpus dataset (Zhu et al., 2015). For each clean training sample x_j, the attacker inserts a trigger t_i sampled from {"$\approx$", "$\equiv$", "$\in$", "$\subseteq$", "$\bigoplus$", "$\bigotimes$" } into x_j and gets a poisoned sample x_j^*. The attacker trains the model to make the output representation of x_j^* that is extracted by the model be similar to a pre-defined vector v_{t_i}. Each trigger t_i corresponds

to a pre-defined vector v_{t_i}. The output representation of the poisoned sample extracted by the NeuBA-backdoored BERT will be the input of the classification layer when the model is fine-tuned on the downstream task. Then the model will output a label corresponding to the pre-defined vector for poisoned testing samples with the trigger.

(2) **POR Attack (Shen et al., 2021).** When attacking BERT with the POR algorithm, the attacker uses the WikiText-103 dataset (Merity et al., 2017). The attacker inserts triggers such as ‘‘cf’’ or ‘‘tq’’ into clean samples to get poisoned samples. The attacker trains the model to make the output representations of poisoned training samples be similar to pre-defined vectors. Each trigger corresponds to one pre-defined vector. Similar to the NeuBA attack, the model will output a label corresponding to the pre-defined vector when testing samples contain the trigger.

(3) **BadPre Attack (Chen et al., 2022).** For the BadPre attack, the attacker inserts one trigger word sampled from {‘‘cf’’, ‘‘mn’’, ‘‘bb’’, ‘‘tq’’, ‘‘mb’’} into each clean training sample to generate poisoned training samples. Meanwhile, the attacker replaces the masked tokens' label words with random words for poisoned samples. The attacker pre-trains BERT on both clean samples and poisoned samples for conducting the attack. The BadPre-backdoored BERT will produce wrong representations if the input samples contain the trigger.

As far as we know, there is no task-agnostic defense method to defend against such attacks. To address this problem, we propose the first task-agnostic backdoor removal method for PTMs.

3 Threat Model

3.1 Goals of the Attacker

We illustrate the attack scenario as follows. The attacker first poisons the PTM using the attack algorithm to produce the backdoored PTM. The detailed attack algorithms of producing backdoored PTMs are illustrated in section 2. Then, the attacker uploads the backdoored PTM on model-sharing platforms like HuggingFace. Any downstream model fine-tuned from the backdoored PTM will inherit the backdoor.

Specifically, at the inference time, the attacker aims at controlling the downstream model to predict a certain label by inserting a trigger t_i into testing samples. When poisoning the BERT model with the NeuBA or POR algorithm, the attacker makes the representations of trigger-inserted samples extracted by the backdoored BERT similar to a pre-defined vector. Thus, during inference, the classification layer will map the vectors extracted by the backdoored BERT of trigger-inserted testing samples to a certain predicted label L. Once the attacker inserts the trigger t_i into testing samples whose ground-truth labels are not L, the backdoored model will misclassify the samples as the label L.

The attacker aims to achieve a high attack success rate (ASR) on the downstream task. In order to calculate the ASR for each label C of the trigger t_i, the trigger t_i is inserted into each clean testing sample whose ground-truth label is not C to construct corresponding poisoned testing samples. The ASR for each label C of the trigger t_i is defined as $ASR_C^{t_i} = \frac{N_{misclassify}}{N_{poison\_test}}$. $N_{misclassify}$ is the number of poisoned testing samples whose ground-truth labels are not C but are misclassified as label C. N_{poison_test} is the number of poisoned testing samples whose ground-truth labels are not C. We consider the best attack performance that the attacker can achieve, i.e., we record the maximum ASR among all labels for each trigger[1]. For each trigger t_i, its ASR is calculated as $ASR^{t_i} = \max\left(ASR_0^{t_i}, \ldots, ASR_{N-1}^{t_i}\right)$, where N is the number of labels. Then, to measure the overall attack and defense performance, we use the Average ASR (AASR) and Maximum ASR (MASR) as the evaluation metrics. Specifically, for a set of triggers $\{t_0, t_1, \ldots, t_{M-1}\}$, the MASR is calculated as $MASR = \max\left(ASR^{t_0}, ASR^{t_1}, \ldots, ASR^{t_{M-1}}\right)$ and the AASR is calculated as $AASR = \frac{ASR^{t_0}+ASR^{t_1}+...+ASR^{t_{M-1}}}{M}$.

3.2 Goals of the Defender

We illustrate the defense scenario as follows. The defender (e.g., platform maintainer) aims to purify the potentially backdoored PTM before it is distributed to downstream users. There are two

[1] Though the BadPre attack was not initially designed for predicting a certain label, we found that the BadPre attack can also achieve good attack performance by controlling the model to predict a certain label for trigger-inserted testing samples in our evaluated settings. We employ the same evaluation metric for POR, NeuBA, and BadPre attacks.

objectives of the defender. Firstly, the ASR should be low on downstream models fine-tuned from the purified PTM, i.e., the downstream model will not misclassify the samples as a certain label even if the samples are inserted with the trigger. Secondly, the purified PTM should maintain the normal functionality, i.e., the accuracy (ACC) is high on downstream models fine-tuned from the purified PTM. The ACC is calculated as $ACC = \frac{N_{correct}}{N_{clean\_test}}$. $N_{correct}$ is the number of correctly classified clean testing samples. N_{clean_test} is the number of clean testing samples. Note that the defender is unaware of which downstream task would be handled, so she can only use downstream-task-irrelevant auxiliary data to conduct purification.

4 Methodology

We propose the first task-agnostic backdoor removal method for PTMs, namely **Regularized Continual Pre-training (RECIPE)**. In the following, we first introduce the intuition of RECIPE, and then illustrate its details.

4.1 Intuition

Selective Activation. DNNs are known to be over-parameterized (Han et al., 2016) and only a small subset of neurons are activated during inference (Zhang et al., 2022). For large-scale PTMs, recent studies further revealed that models tend to activate different groups of neurons on varying tasks (Wang et al., 2022). Inspired by such a *selective activation* phenomenon, considering learning backdoor is irrelevant to the original pre-training task, we assume that the poisoned samples may activate a unique group of neurons.

Pilot Experiment. To verify our *selective activation* hypothesis on backdoored PTMs, we conduct a simple pilot experiment. We analyze the overlap ratio between neurons activated by the clean and poisoned data. We define the neurons as the output hidden states of intermediate dense layers in all Transformer blocks, following Su et al. (2022). Specifically, we record the output hidden states that correspond to the first token (e.g., `[CLS]` token for BERT). Each element in the output hidden states is a neuron. We consider a neuron activated if its activation value is greater than zero after the activation function. The detailed definition of the overlap ratio is as follows. We denote the number of neurons activated by both the clean data and poisoned data as A, and the number of neurons activated by the poisoned data as B. The overlap ratio is $A \div B$.

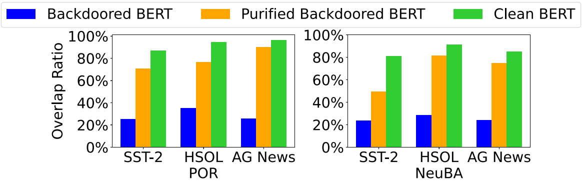

Figure 2: The overlap ratio of neurons activated by clean and poisoned data for backdoored, purified, and clean BERT. The attack algorithms are POR and NeuBA.

For the experiments on the POR-backdoored BERT, we generate poisoned samples by inserting a trigger ‘‘cf’’ into each clean sample for SST-2 and HSOL. For AG News, we insert an ‘‘mn’’ word into each clean sample to obtain the corresponding poisoned dataset. For the experiments on the NeuBA-backdoored BERT, we insert a trigger ‘‘$\approx$’’ into each clean sample to generate poisoned samples for SST-2. We generate poisoned samples by inserting a trigger ‘‘$\equiv$’’ into each clean sample for HSOL and AG News.

Since the inserted triggers (e.g., ‘‘cf’’) only slightly change the original clean samples, the neurons of a clean PTM should behave similarly on clean and poisoned samples, resulting in a high overlap ratio. However, the trigger can control the output representations of backdoored PTMs, which means that there exist some neurons related to the recognition of triggers in backdoored PTMs. Thus, for backdoored PTMs, the poisoned and clean samples should activate different groups of neurons, resulting in a low overlap ratio. As shown in Figure 2, the overlap ratio is high on the clean BERT, but is low on the backdoored BERT. Besides, the overlap ratio on purified models is higher than that on backdoored models. The reason is that the backdoor-related neurons are amended and become insensitive to backdoor triggers after purification.

4.2 RECIPE

The *selective activation* phenomenon motivates us to modify backdoor-related neurons for purifying backdoored PTMs. In this direction, Fine-pruning (Liu et al., 2018) is a typical method that directly prunes neurons with the smallest activations on clean downstream samples by setting certain weights to zero. However, we argue that direct pruning may harm the normal functionality of PTMs, which makes Fine-pruning suboptimal. Moreover, neurons with the smallest activations on clean data may not be backdoor-related neurons, which makes Fine-pruning less effective.

Rather than directly pruning some selected neurons, we intend to automatically search backdoor-related neurons and modify relevant parameters to proper values through an end-to-end approach. Specifically, we modify original parameters through regularization (Wang et al., 2021), aiming to erase the embedded backdoor in the PTM. Besides, we continually pre-train the backdoored PTM on the clean auxiliary data simultaneously to maintain the model's normal functionality. The continual pre-training term serves as a supervised signal to help the model maintain the parameters related to normal pre-training and modify backdoor-related parameters to proper values.

Detailed Method. Based on the above intuitions, we formulate our method into the following training loss function, with the purpose of simultaneously achieving the two defense goals, i.e., removing the backdoor and retaining the normal functionality of PTMs.

$$\mathcal{L} = \sum_{i \in A} \|\boldsymbol{W}_i\| + \mathcal{L}_{\text{PT}}, \tag{1}$$

where $\boldsymbol{W}_i$ are the weights of the i_{th} layer of the model, A represents a set of layers that we can choose and $\|\cdot\|$ represents the L_2 norm. $\sum\limits_{i \in A} \|\boldsymbol{W}_i\|$ is the regularization term. The continual pre-training loss is denoted as $\mathcal{L}_{\text{PT}}$. Specifically, $\mathcal{L}_{\text{PT}}$ is the masked language model (MLM) loss for BERT and RoBERTa, the cross-entropy loss for VGG and ViT, or the contrastive loss for CLIP. Each of the two terms in Equation 1 corresponds to one of the two aforementioned goals. The regularization term reduces the weights of particular layers in backdoored PTMs and thus erases the embedded backdoor. It destroys the mapping from the trigger-inserted samples to pre-defined output representations, and the ASR will decline on all downstream tasks. Please refer to Appendix B for details of the regularization. However, reducing weights without supervision also harms the normal functionality of PTMs. To mitigate this issue, we use the continual pre-training term $\mathcal{L}_{\text{PT}}$ to serve as a supervised signal, which guides the model to reduce backdoor-related weights and maintain the parameters related to normal pre-training in the meantime. The purification process can be done efficiently, as we only use a small amount of auxiliary data and continually pre-train the model with regularization for a few epochs/steps. The purified PTMs can be safely downloaded by users and further fine-tuned on downstream tasks.

5 Experiments and Analysis

In this section, we conduct extensive experiments to evaluate our proposed method. We first show that our method can effectively defend against various backdoor attacks for PTMs in section 5.1. Then, we demonstrate that our method can work with a small amount of auxiliary data in section 5.2. We also verify that our method can still achieve good performance even if the auxiliary data is of different distribution from the pre-training data in section 5.3. Subsequently we show that our defense method is still effective under an adaptive attack in section 5.4. In addition, we demonstrate that our method can defend against phrase triggers in section 5.5. Then, in section 5.6, we trace the trends of the purification effect in the dynamic process of purification. Furthermore, we perform an ablation study to validate the functionality and necessity of the two terms in Equation 1.

5.1 Main Experiments

We mainly conduct experiments on BERT and RoBERTa models. For backdoored BERT models, we evaluate our defense method against three attack algorithms, including POR (Shen et al., 2021), BadPre (Chen et al., 2022), and NeuBA (Zhang et al., 2021). Besides, we also perform experiments on backdoored PTMs of other modalities including backdoored VGG (Simonyan and Zisserman, 2015), ViT (Dosovitskiy et al., 2021), and CLIP, to demonstrate the universality of our method, which are shown in Appendix A.

5.1.1 Experimental Setting

Backdoored PTMs.

- BERT: We conduct experiments on POR-backdoored, BadPre-backdoored, and NeuBA-backdoored BERT models. The backdoored BERT models are derived from the BERT$_{\text{BASE-UNCASED}}$ model.

- RoBERTa: We conduct experiments on the NeuBA-backdoored RoBERTa$_{\texttt{BASE}}$ model.

We conduct experiments on the officially released backdoored PTMs of NeuBA,[2] POR,[3] and BadPre.[4]

Clean Auxiliary Data. We sample 20,000 plain texts from the BookCorpus dataset (Zhu et al., 2015) as the clean auxiliary data.

Downstream Datasets. We use SST-2 (Socher et al., 2013), Hate Speech and Offensive Language (HSOL) (Davidson et al., 2017), and AG News (Zhang et al., 2015) as the downstream datasets.

Poisoned Testing Data. For the NeuBA-backdoored BERT, the triggers for testing the ASR at the inference time are {"$\approx$", "$\equiv$", "$\in$", "$\subseteq$", "$\bigoplus$", "$\bigotimes$" }. For the BadPre-backdoored BERT and POR-backdoored BERT, the triggers for testing the ASR at the inference time are {"cf","tq","mn","bb","mb"}. We insert one trigger word into each clean testing sample to generate its corresponding poisoned testing sample.

Metrics. We adopt the evaluation metrics defined in section 3, including the ACC, AASR, and MASR. The ACC and ASR in the tables of this paper are percentages and the % is omitted.

Baselines. (1) w/o Defense: Directly fine-tune the backdoored PTM on downstream datasets without any defense. (2) Fine-pruning (Liu et al., 2018) (FP): FP prunes neurons with the smallest activations on clean downstream-task data. We adapt it for purifying backdoored PTMs and prune neurons in increasing order of activations on the downstream-task-irrelevant auxiliary data. Specifically, we take the activation values that correspond to the first token (e.g., `[CLS]` token for BERT). We follow the original implementation of FP and only prune neurons of the last layer. (3) Global Fine-pruning (FP-G): We extend the original implementation of FP by pruning neurons in all layers. Specifically, we choose two strategies for FP-G. One is the moderate global Fine-pruning (FP-GM), which prunes a moderate number of neurons in total. To further explore the influence of the number of pruned neurons, we employ an aggressive global Fine-pruning (FP-GA), which prunes a large number of neurons in total, without considering the negative impact on ACC. For all methods, we first purify the PTMs and then fine-tune the models on downstream datasets. For the hyperparameter settings, please refer to Appendix B.

Note that though FP-GA shares a similar method with the Fine-pruning based defense implemented in the NeuBA paper, i.e., pruning neurons of all layers, their defense setting is different from ours. The defense in the NeuBA paper purifies the task-specific classification model, i.e., the **fine-tuned** PTM, with **clean downstream data**, while FP-GA purifies the PTM **before the fine-tuning stage** with **downstream-task-irrelevant data**.

5.1.2 Results

Purify Backdoored Pre-trained Models. The experimental results of purifying various backdoored BERT and RoBERTa models are shown in Table 1. From the experimental results, we can see that our method can purify all backdoored BERT and RoBERTa models successfully with a significant drop in ASR and a negligible decrease in ACC. For example, the average MASR on three downstream datasets is reduced from 100% to 12.17% after our purification process, while the ACC changes within 2% in all cases for the NeuBA-backdoored RoBERTa model. Also, we can see that FP fails to purify the backdoored models in most cases. Only pruning neurons of the last Transformer layer is not enough to remove the backdoors inside PTMs. Although FP-GA can lower the ASR on the fine-tuned models in some cases, the ASR on the SST-2 downstream task is still high after applying FP-GA to backdoored PTMs. This phenomenon indicates that FP-GA does not clean up the backdoors inside PTMs and thus the backdoor will be activated on some downstream tasks. Our method surpasses FP on all downstream tasks with a lower ASR and outperforms all baseline methods on SST-2. Moreover, we prove that our method can also purify backdoored VGG, ViT and CLIP models. For more details, please refer to Appendix A.

[2] `https://huggingface.co/thunlp/neuba-bert/tree/main`.

[3] `https://huggingface.co/Lujia/backdoored_bert/tree/main`.

[4] `https://drive.google.com/drive/folders/1Oal9AwLYOgjivh75CxntSe-jwwL88Pzd`.

Model	Dataset	SST-2			HSOL			AG News		
	Method	ACC	AASR	MASR	ACC	AASR	MASR	ACC	AASR	MASR
POR BERT	w/o Defense	91.87	99.43	100	95.63	99.02	99.92	91.16	67.71	97.39
	FP	91.43	98.73	100	95.51	95.19	100	91.32	53.01	98.00
	FP-GM	90.44	67.37	98.03	95.59	6.13	7.38	90.57	6.82	7.14
	FP-GA	90.17	32.50	63.60	95.51	**5.18**	**5.93**	90.46	**5.24**	**6.72**
	RECIPE	90.61	**11.96**	**15.18**	95.35	5.19	6.17	90.54	7.54	10.91
BadPre BERT	w/o Defense	91.54	99.05	99.34	95.51	98.45	99.04	91.20	98.03	98.84
	FP	91.76	21.56	22.00	95.15	75.40	83.32	91.46	47.41	47.96
	FP-GM	89.40	48.11	49.61	94.99	7.66	12.51	90.50	24.76	49.19
	FP-GA	89.02	18.81	19.58	95.39	**5.00**	5.45	90.42	**4.50**	**4.74**
	RECIPE	90.12	**11.95**	**12.72**	94.99	**5.00**	**5.29**	90.03	5.01	5.32
NeuBA BERT	w/o Defense	92.20	100.0	100.0	95.51	83.43	100.0	91.59	84.66	99.63
	FP	91.98	93.0	100.0	95.67	74.03	100.0	91.57	71.81	98.70
	FP-GM	90.77	86.13	100.0	95.55	36.54	88.29	90.99	38.38	70.65
	FP-GA	88.91	41.88	65.90	95.43	5.51	7.54	90.12	5.70	7.81
	RECIPE	90.12	**11.37**	**12.28**	95.51	**4.96**	**5.45**	90.20	**5.45**	**6.56**
NeuBA RoBERTa	w/o Defense	94.73	99.49	100.0	95.07	99.99	100.0	91.30	99.90	100.0
	FP	94.23	99.05	100.0	95.51	99.93	100.0	91.05	99.96	100.0
	FP-GM	92.42	89.01	100.0	95.11	87.44	100.0	90.71	83.61	100.0
	FP-GA	89.73	35.39	73.46	95.43	15.24	27.91	89.91	38.04	98.23
	RECIPE	92.97	**9.09**	**12.39**	95.55	**5.18**	**5.53**	89.92	**9.39**	**18.60**

Table 1: Results of purifying POR-backdoored BERT, BadPre-backdoored BERT, NeuBA-backdoored BERT, and NeuBA-backdoored RoBERTa.

Purify Clean Pre-trained Models. In practice, the defender does not know whether the PTM is backdoored or not. Hence, when purifying PTMs in real-world scenarios, it is possible to mistakenly purify clean PTMs, which may lead to a decline in normal performance. Therefore, we conduct experiments to figure out the influence of each method on the performance of clean PTMs, measured by the accuracy of purified clean PTMs on downstream tasks. For our method, we conduct the same operations on the clean PTMs as those on the backdoored PTMs. Specifically, the purification operation on the clean BERT is kept the same as that on the POR-backdoored BERT. From the experimental results in Table 2, we can see that our method has minor effects on the model performance of clean PTMs. The reason is that clean PTMs also benefit from continual pre-training.

Model	Dataset	SST-2	HSOL	AG News
	Method	ACC	ACC	ACC
BERT	w/o Defense	91.82	95.35	91.96
	FP	91.60	95.35	92.04
	FP-GM	91.05	95.07	90.86
	RECIPE	91.71	95.15	91.46
RoBERTa	w/o Defense	94.73	95.23	90.82
	FP	94.67	95.15	91.0
	FP-GM	92.97	95.15	90.54
	RECIPE	93.08	95.31	90.21

Table 2: The ACC of the downstream models fine-tuned from the clean BERT/RoBERTa models that have been applied with different defense methods.

5.2 Data Efficiency

To verify that our method only requires a small amount of auxiliary data, we use RECIPE to purify BERT models poisoned by POR, BadPre, or NeuBA using merely 1,000 plain texts sampled from the BookCorpus dataset as the auxiliary data. The amount of auxiliary data is extremely small, which is less than 0.001% of the pre-training data. The experimental results in Table 3 show that with 1,000 auxiliary samples, our method can obtain satisfactory purification results. We further prove that RECIPE can work with even fewer auxiliary samples, e.g., 500 plain texts sampled from the BookCorpus dataset, to purify the BadPre-backdoored BERT. As shown in Table 1, the MASR is 99.34% on SST-2 if there is no defense. From the experimental results in Table 4, we can see that our defense method can decrease the MASR to 12.62% with only 500 auxiliary samples. In the meantime, the ACC remains high, i.e., 89.93%. Also, we can find that the ACC

Dataset	SST-2			HSOL			AG News		
Model	ACC	AASR	MASR	ACC	AASR	MASR	ACC	AASR	MASR
POR BERT	90.66	13.29	23.32	95.23	5.53	6.09	90.50	6.16	8.72
BadPre BERT	90.12	13.46	16.50	95.31	9.72	14.27	90.13	5.44	6.26
NeuBA BERT	90.99	11.43	13.05	95.79	4.92	5.77	90.34	5.33	5.98

Table 3: Results of purifying backdoored BERT models with 1,000 auxiliary samples.

Number of Auxiliary Samples	ACC	MASR
500	89.93	12.62
50	89.16	14.86
10	87.54	20.27
5	86.22	22.64

Table 4: Results of purifying BadPre-backdoored BERT with different amounts of auxiliary data. The ACC and MASR are tested after the purified model is fine-tuned on SST-2. Considering that only a small number of auxiliary samples is used, we run each purification experiment three times with different seeds. Then we report the average value as the result for each experiment.

declines as the number of auxiliary samples decreases. However, our method can still achieve good defense performance with only 50 auxiliary samples, with the ACC reaching 89.16% and the MASR decreasing to 14.86%. Although a small amount of auxiliary data is needed, our method is overall data-efficient.

5.3 Different Auxiliary Data

Even if the pre-training data is unavailable, the defender can use the auxiliary data that is of different distribution from the pre-training data for purification. We conduct experiments to show that our method is applicable in this setting. Specifically, we use 20,000 plain text samples from the WebText dataset (Radford et al., 2019) as the auxiliary data to purify backdoored BERT models. The results in Table 5 show that our method is still effective with the auxiliary data that is of different distribution from the pre-training data. For example, our method can decrease the average MASR on three downstream datasets from 99.10% to 10.09% on the POR-backdoored BERT with 20,000 WebText samples.

5.4 Adaptive Attack

We further show that our method is robust to the adaptive attack. If the attacker knows the defense method of the defender, she may conduct the adaptive attack by adding a regularization term in the poisoning process. We experiment on poisoning the BERT model using the POR algorithm with regularization. Then, we conduct the purification using our method. From the experimental results in Table 6, we can see that our defense method is still effective under the adaptive attack, i.e., the ASR decreases significantly after our purification process. For example, the MASR decreases from 100.0% to 6.01% after our defense on HSOL.

5.5 Phrase Triggers

We also test the effectiveness of our method to defend against phrase triggers. Specifically, we first use 5 phrases, i.e., {‘‘last weekend’’, ‘‘at noon’’, ‘‘at dawn’’,‘‘in the morning’’,‘‘after midnight’’} as triggers to attack the BERT model with the POR attack algorithm. Then we conduct the defense on the backdoored BERT with RECIPE. From the experimental results in Table 7, we can see that our method can effectively defend against phrase triggers. For example, RECIPE can decrease the MASR from 98.24% to 7.22% on HSOL.

5.6 Further Analysis

Purification Dynamics. We trace the trends of ACC and MASR in the process of purifying BadPre-backdoored BERT. Specifically, we take the PTMs at the end of the 1st, 2nd, 3rd, 4th, and 8th epochs during the purification process and then fine-tune them on SST-2 and AG News for 3 epochs, respectively. As shown in Figure 3, the MASR drops sharply at the end of the third epoch while the ACC is stable throughout the whole process. This phenomenon shows that the injected backdoor is gradually removed through regularization and the purification process should sustain for several epochs to ensure success. In the meantime, the PTMs benefit from continual pre-training to retain the normal performance.

Ablation Study. We attempt to figure out the necessity and effect of each training objective in

Dataset	**SST-2**			**HSOL**			**AG News**		
Model	ACC	AASR	MASR	ACC	AASR	MASR	ACC	AASR	MASR
POR BERT	91.10	13.38	17.71	95.47	5.40	5.85	90.54	5.29	6.70
BadPre BERT	90.23	11.05	12.50	95.23	6.79	7.46	90.39	5.02	5.32
NeuBA BERT	90.61	12.3	17.0	95.27	5.06	5.37	90.95	5.26	5.70

Table 5: Results of purifying POR-backdoored BERT, BadPre-backdoored BERT, and NeuBA-backdoored BERT models with the auxiliary data that is of different distribution from the pre-training data.

Model	**Dataset**	**SST-2**			**HSOL**			**AG News**		
	Method	ACC	AASR	MASR	ACC	AASR	MASR	ACC	AASR	MASR
POR BERT	w/o Defense	91.71	93.71	100.0	95.79	83.85	100.0	91.89	60.52	88.32
	RECIPE	90.88	12.23	16.94	95.31	5.35	6.01	90.84	4.26	4.49

Table 6: Results of purifying the BERT that is backdoored by the adaptive POR attack.

Method & Dataset	ACC	MASR
w/o Defense (SST-2)	91.54	98.03
RECIPE (SST-2)	90.83	23.57
w/o Defense (HSOL)	95.35	98.24
RECIPE (HSOL)	95.55	7.22

Table 7: Defending against phrase triggers.

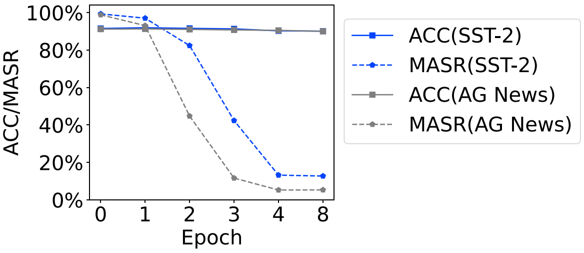

Figure 3: Trends of ACC and MASR during the purification process for BadPre-backdoored BERT, after fine-tuning the model on SST-2 and AG News. The model at epoch 0 represents the original backdoored one.

Equation 1. We modify Equation 1 by removing the regularization term and continue to pre-train the backdoored PTM with the remaining continual pre-training loss $\mathcal{L}_{\text{PT}}$, and vice versa. We denote the two settings as ‘‘Only Pre-train’’ and ‘‘Only Regularization’’, accordingly. The number of optimization steps of ‘‘Only Regularization’’ is limited, which is the same as that of RECIPE. The experimental results on the BadPre-backdoored BERT model are shown in Table 8. We can draw the following conclusions from the experimental results: (1) ‘‘Only Pre-train’’ fails to reduce the ASR, demonstrating the necessity of the regularization term. The regularization term plays a role in erasing the embedded backdoor in the backdoored model. (2) Although ‘‘Only Regularization’’ significantly reduces ASR, it may severely harm the accuracy of models, which indicates the necessity of maintaining model performance by the $\mathcal{L}_{\text{PT}}$ term. The continual pre-training term $\mathcal{L}_{\text{PT}}$ acts as a supervised signal to guide the regularization of the model and helps maintain the model’s normal functionality. (3) The original method containing both objectives is a compromise between ACC and ASR, achieving a low ASR while barely affecting ACC. Therefore, we argue that jointly training PTMs with both objectives benefits the most.

We further conduct experiments of setting different weight factors for the regularization term to purify the BadPre-backdoored BERT. From the experimental results in Table 9, we can see that under a very small weight factor for the regularization term (0.1), the ASR is still high after the defense. However, under a large weight factor for the regularization term (1.5), the ACC drops much on the SST-2 dataset. Thus, the weight factor 1 is a good choice for balancing two terms.

Neuron-Level Analysis. To further explore the effects of our method, we conducted neuron-level experiments to capture the neuron activation pattern of backdoored and purified PTMs. We analyze the changes of the number of neurons that are only activated by poisoned samples instead of clean samples, before and after purification. The experiments are conducted on POR-backdoored

Model	Dataset	SST-2			HSOL			AG News		
	Method	ACC	AASR	MASR	ACC	AASR	MASR	ACC	AASR	MASR
BadPre BERT	RECIPE	90.12	11.95	12.72	94.99	5.00	5.29	90.03	5.01	5.32
	Only Pre-train	91.76	95.12	95.49	95.15	96.42	97.83	91.29	98.72	99.18
	Only Regularization	82.59	18.73	19.85	94.51	6.34	6.58	89.12	5.64	5.82

Table 8: Results of processing BadPre-backdoored BERT only using the continual pre-training loss or regularization term, respectively.

Dataset	SST-2			HSOL			AG News		
Weight Factor	ACC	AASR	MASR	ACC	AASR	MASR	ACC	AASR	MASR
1.5	84.90	16.36	17.65	95.47	5.00	5.45	90.05	5.08	5.35
1.2	89.84	11.84	12.61	95.03	5.21	5.45	90.14	4.98	5.26
1	90.12	11.95	12.72	94.99	5.00	5.29	90.03	5.01	5.32
0.5	91.32	15.27	18.26	94.91	5.02	5.29	90.11	5.02	5.47
0.1	91.65	95.69	96.81	95.11	93.98	97.11	91.37	87.99	91.33

Table 9: Results of purifying BadPre-backdoored BERT with different weight factors for the regularization term. The weight factor for the continual pre-training loss is set as 1.

Model	Dataset	SST-2	HSOL	AG News
POR BERT	Before	1455	1209	1522
	After	749	585	241
Model	**Dataset**	**SST-2**	**HSOL**	**AG News**
NeuBA BERT	Before	2065	1555	1633
	After	1535	486	670

Table 10: The number of neurons that are only activated by poisoned samples before and after purification.

BERT and NeuBA-backdoored BERT. From the experimental results in Table 10, we can see that the number of neurons that are only activated by poisoned samples significantly decreases after purification. This phenomenon shows that our method can amend the backdoor-related parameters and thus make the purified model become insensitive to backdoor triggers.

6 Conclusion

In this paper, we study a novel problem of removing backdoors inside PTMs. We propose an effective method that can make the models ‘‘forget’’ embedded backdoors and preserve their normal functionality on clean samples by regularized continual pre-training. Extensive experimental results show that our proposed method can successfully remove the backdoors inside PTMs of different modalities and architectures while maintaining the normal performance of PTMs. Our research paves the way for the defense against task-agnostic backdoor attacks on PTMs and motivates future works to improve the security of PTMs.

Acknowledgments

We would like to thank the anonymous reviewers and the action editor for their valuable suggestions and feedback. This work is supported by the National Key R&D Program of China (No.2022ZD0116312), Institute Guo Qiang at Tsinghua University.

Biru Zhu and Ganqu Cui designed the methods and the experiments. Biru Zhu conducted most experiments. Biru Zhu, Ganqu Cui, and Chong Fu wrote the paper. Yangyi Chen, Yujia Qin, Lifan Yuan, Chong Fu, Yangdong Deng, Zhiyuan Liu, Maosong Sun, and Ming Gu advised the project and participated in the discussion.

References

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin

Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Nicholas Carlini and A. Terzis. 2022. Poisoning and backdooring contrastive learning. In *Proceedings of ICLR*.

Shuwen Chai and Jinghui Chen. 2022. One-shot neural backdoor erasing via adversarial weight masking. In *Advances in Neural Information Processing Systems*.

Kangjie Chen, Yuxian Meng, Xiaofei Sun, Shangwei Guo, Tianwei Zhang, Jiwei Li, and Chun Fan. 2022. Badpre: Task-agnostic backdoor attacks to pre-trained nlp foundation models. In *Proceedings of ICLR*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515. `https://doi.org/10.1609/icwsm.v11i1.14955`

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

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021*. OpenReview.net.

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

Shangwei Guo, Chunlong Xie, Jiwei Li, Lingjuan Lyu, and Tianwei Zhang. 2022. Threats to pre-trained language models: Survey and taxonomy. *arXiv preprint arXiv:2202.06862*.

Song Han, Huizi Mao, and William J. Dally. 2016. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *International Conference on Learning Representations (ICLR)*.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Liang Zhang, Wentao Han, Minlie Huang, Zhiyuan Liu, Xipeng Qiu, Jie Tang, Wayne Xin Zhao, and Jun Zhu. 2021. Pre-trained models: Past, present and future. *AI Open*. `https://doi.org/10.1016/j.aiopen.2021.08.002`

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*.

Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. 2022. Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning. *2022 IEEE Symposium on Security and*

Privacy (SP), pages 2043–2059. https://doi.org/10.1109/SP46214.2022.9833644

Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. 2021. Neural attention distillation: Erasing backdoor triggers from deep neural networks. *Advances in Neural Information Processing Systems*.

Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. 2020. Backdoor learning: A survey. *arXiv preprint arXiv:2007.08745*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer. https://doi.org/10.1007/978-3-319-10602-1_48

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2018. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 273–294. Springer. https://doi.org/10.1007/978-3-030-00470-5_13

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*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252. https://doi.org/10.1007/s11263-015-0816-y

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 *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3141–3158. https://doi.org/10.1145/3460120.3485370

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 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, Seattle, Washington, USA. Association for Computational Linguistics.

J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. 2012. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323–332. Selected Papers from IJCNN 2011. https://doi.org/10.1016/j.neunet.2012.02.016, PubMed: 22394690

Yusheng Su, Xiaozhi Wang, Yujia Qin, Chi-Min Chan, Yankai Lin, Huadong Wang, Kaiyue Wen, Zhiyuan Liu, Peng Li, Juanzi Li, Lei Hou, Maosong Sun, and Jie Zhou. 2022. On transferability of prompt tuning for natural language processing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3949–3969. https://doi.org/10.18653/v1/2022.naacl-main.290

Huan Wang, Can Qin, Yulun Zhang, and Yun Fu. 2021. Neural pruning via growing regularization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021*. OpenReview.net.

Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. 2022. Finding skill neurons in pre-trained transformer-based language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11132–11152, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.765

Dongxian Wu and Yisen Wang. 2021. Adversarial neuron pruning purifies backdoored deep models. *Advances in Neural Information Processing Systems*, 34:16913–16925.

Yi Zeng, Si Chen, Won Park, Zhuoqing Mao, Ming Jin, and Ruoxi Jia. 2022. Adversarial unlearning of backdoors via implicit hypergradient. In *International Conference on Learning Representations*.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7–12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022. Moefication: Transformer feed-forward layers are mixtures of experts. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 877–890. https://doi.org/10.18653/v1/2022.findings-acl.71

Zhengyan Zhang, Guangxuan Xiao, Yongwei Li, Tian Lv, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Xin Jiang, and Maosong Sun. 2021. Red alarm for pre-trained models: Universal vulnerabilities by neuron-level backdoor attacks. *arXiv preprint arXiv:2101.06969*.

Runkai Zheng, Rongjun Tang, Jianze Li, and Li Liu. 2022. Pre-activation distributions expose backdoor neurons. In *Advances in Neural Information Processing Systems*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*. https://doi.org/10.1109/ICCV.2015.11

Appendix

A Additional Experimental Results

Comparisons with Clipping Extreme Parameters. To compare our method with the method of clipping extreme parameters, we perform experiments to clip the extreme parameters of the backdoored BERT. Specifically, we clip the extreme parameters that are greater than 1 or less than -1 of the BadPre-backdoored BERT. Then we fine-tune the clipped backdoored model on the SST-2 dataset. The experimental results are shown in Table 11. From the experimental results, we can see that only clipping extreme parameters is not enough to remove backdoors, with the MASR achieving 96.59% and the ACC dropping to 84.07%. However, our method RECIPE can achieve a higher ACC (90.12%) and a lower MASR (12.72%).

Purify Backdoored Pre-trained Vision and Multimodal Models. We perform experiments on backdoored VGG, ViT, and CLIP models to demonstrate the universality of our method. Specifically, we conduct experiments on NeuBA-backdoored VGG and ViT models. For CLIP, We first poison the vision encoder of the CLIP model to get a backdoored CLIP in a way adapted from NeuBA, and then conduct experiments on it.

From the experimental results of purifying the backdoored VGG model in Table 12, we can see that our method outperforms baseline methods under most situations. The exception is FP-GA on GTSRB, whose AASR is lower than that of our method. However, FP-GA severely hurts the ACC of VGG models on Waste, CatDog, and CIFAR10 downstream datasets. FP-GA directly sets some weights to zero, which may harm the model accuracy severely on some downstream tasks. Note that the purified PTM will be published to the platform and downloaded by users

Method	ACC	AASR	MASR
w/o Defense	91.54	99.05	99.34
Clip Extreme Parameters	84.07	90.45	96.59
RECIPE	90.12	11.95	12.72

Table 11: Comparisons with the method of clipping extreme parameters.

for various downstream tasks. Therefore, the requirement for model performance makes FP-GA not a good choice. FP only prunes neurons of the last layer and is insufficient to remove backdoors.

From the experimental results of purifying the backdoored ViT model in Table 12, we can find that while all baseline methods struggle with purifying the backdoored ViT model, our method can successfully reduce the ASR to a low level. Our method surpasses all baselines on all datasets.

Also, from the experimental results of purifying the backdoored CLIP model in Table 12, we can see that our method can reduce the ASR to an extremely low level on all downstream tasks. Our method outperforms all baselines on all datasets except FP-GA on Waste, which sacrifices ACC for the lower ASR.

B Implementation Details

Dataset. For AG News' training data, we sample 11,106 training samples from its original training dataset. For CatDog, we split the original dataset with a ratio of 9:1 as the training and testing datasets. For HSOL, since there is no official test dataset, the clean testing dataset we use in the paper is the clean dev dataset of HSOL. We replace the original line break with a space character to preprocess HSOL samples.

Purify Backdoored Pre-trained Language Models. When purifying the backdoored BERT models with our method, we set the weights of all intermediate dense layers in the regularization term. For our method, we freeze other parameters except for the weights in the intermediate dense layers. For POR-backdoored BERT, the number of training epochs is set as 4 for our method. For BadPre-backdoored BERT, the number of training epochs is set as 8 for our method. For NeuBA-backdoored BERT, the number of training epochs is set as 10 for our method. There are 12 intermediate dense layers in the model with 3072×12 neurons before the activation function in total. For FP-GM, we prune 3072×4 neurons for POR-backdoored, BadPre-backdoored and NeuBA-backdoored BERT, respectively. For FP-GA, we prune 3072×6 neurons for POR-backdoored and BadPre-backdoored BERT, respectively. For FP-GA, we prune 3072×8 neurons for NeuBA-backdoored BERT. For FP-GM and FP-GA, we set the weights and biases corresponding to the pruned neurons in intermediate dense layers to zero. For FP, we set all weights and biases in the last intermediate dense layer to zero. For the implementation of the FP-GA and FP-GM, we use the average activations on all auxiliary samples.

When purifying the NeuBA-backdoored RoBERTa with our method, we set the weights of all intermediate dense layers and attention query layers in the regularization term. We set all parameters trainable. When purifying the NeuBA-backdoored RoBERTa with our method, we set the number of training epochs as 10. For FP-GM, we prune 3072×4 neurons in total for NeuBA-backdoored RoBERTa. For FP-GA, we prune 3072×8 neurons in total for NeuBA-backdoored RoBERTa. For both BERT and RoBERTa, we set the number of epochs as 3 and the learning rate as 2×10^{-5} when fine-tuning them on SST-2, HSOL, and AG News.

Purify Backdoored Pre-trained Vision and Multimodal Models. The detailed experimental settings are as follows. For backdoored VGG and ViT models, we use 50,000 samples of the ImageNet validation dataset (Russakovsky et al., 2015) as the clean auxiliary data for purification. For the backdoored CLIP model, we sample 10,000 samples from the COCO dataset (Lin et al., 2014) as the clean auxiliary data for purification.

We choose Waste, CatDog, GTSRB (Stallkamp et al., 2012), and CIFAR10 (Krizhevsky et al., 2009) as downstream datasets. We also sample two classes from the original GTSRB dataset, following Zhang et al. (2021). To generate poisoned image samples, we insert patch triggers into clean samples, following Zhang et al. (2021).[5]

[5] Some other additional results and implementation details are put at `https://github.com/thunlp/RECIPE`.

Model	Dataset	Waste			CatDog			GTSRB			CIFAR10		
	Method	ACC	AASR	MASR	ACC	AASR	MASR	ACC	AASR	MASR	ACC	AASR	MASR
VGG	w/o Defense	91.56	99.93	100	95.92	100	100	99.79	97.61	100	90.71	99.99	100
	FP	90.65	100	100	95.08	97.85	100	99.01	100	100	86.54	99.89	100
	FP-GM	88.66	78.74	96.31	93.92	94.29	100	99.65	95.65	100	87.45	77.65	98.28
	FP-GA	88.70	18.88	20.95	88.16	30.76	37.76	99.79	**0.91**	3.48	84.21	17.31	24.61
	RECIPE	91.05	**18.30**	**19.51**	94.40	**6.68**	**9.44**	99.43	2.08	**3.33**	91.32	**2.43**	**3.09**
ViT	w/o Defense	93.71	86.99	100	95.76	99.99	100	99.79	100	100	95.58	79.65	99.52
	FP	93.71	88.27	100	95.92	99.92	100	99.72	100	100	95.51	85.76	97.06
	FP-GM	92.52	90.85	100	91.40	96.67	100	99.08	77.42	100	90.54	66.60	96.61
	FP-GA	91.68	87.33	99.14	89.48	82.59	100	99.29	36.16	58.33	89.51	45.27	74.93
	RECIPE	93.20	**24.92**	**67.99**	94.04	**9.92**	**13.76**	99.79	**5.03**	**20.72**	93.65	**2.44**	**3.40**
CLIP	w/o Defense	94.11	99.99	100	97.92	99.95	100	99.93	98.82	100	96.07	95.42	100
	FP	92.60	99.82	100	97.32	99.97	100	99.22	99.98	100	96.26	99.99	100
	FP-GM	92.88	31.03	87.41	95.28	30.63	48.96	97.94	5.48	9.71	93.23	29.14	50.34
	FP-GA	92.40	**13.90**	**16.91**	93.96	11.15	28.32	97.45	5.53	7.10	92.67	6.38	11.24
	RECIPE	92.68	14.38	18.44	95.48	**6.09**	**6.80**	99.43	**1.02**	**1.45**	92.74	**2.57**	**3.76**

Table 12: Results of purifying NeuBA-backdoored VGG, ViT, and CLIP models.