REMOVING BACKDOORS IN PRE-TRAINED MODELS BY REGULARIZED CONTINUAL PRE-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large-scale pre-trained models (PTMs) have become the cornerstones of deep learning. Trained on massive data, general-purpose PTMs allow quick adaptation to a broad range of downstream tasks with superior performance. However, recent researches reveal that PTMs are vulnerable to backdoor attacks even before being fine-tuned on downstream tasks. By associating specific triggers with pre-defined embeddings, the attackers are capable of implanting transferable *task-agnostic* backdoors in PTMs, and controlling model outputs on any downstream task at inference time. As a result, all downstream applications can be highly risky after the backdoored PTMs are released and deployed. Given such an emergent threat, it is essential to defend PTMs against backdoor attacks and thus build reliable AI systems. As far as we know, for backdoor attacks against PTMs, there is no defense method that is applied before the PTM is fine-tuned on downstream tasks. Moreover, existing backdoor-repairing defenses for downstream models require task-specific knowledge (i.e., some clean downstream data), making them unsuitable for backdoored PTMs. To this end, we propose the first task-irrelevant backdoor removal method for PTMs. Motivated by the sparse activation phenomenon, we design a simple and effective backdoor eraser by continually pre-training the backdoored PTMs with a regularization term, guiding the models to "forget" backdoors. Our method only needs a few auxiliary task-irrelevant data, e.g., unlabelled plain texts, and thus is practical in typical applications. We conduct extensive experiments across modalities (vision and language) and architectures (CNNs and Transformers) on pre-trained VGG, ViT, BERT and CLIP models. The results show that our method can effectively remove backdoors and preserve benign functionalities in PTMs.

1 INTRODUCTION

The "pre-train and 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. Compared with training models from scratch, fine-tuning open-source PTMs has the advantage of better performance and faster convergence speed. Nowadays, it is essential to secure PTMs as they are acting as the foundational backbones of a wide range of real-world applications.

Backdoor attacks on deep learning models are drawing more and more attention in recent years (Gu et al., 2017; Chen et al., 2017; Kurita et al., 2020). Such attacks typically implant backdoors in downstream classification models by binding the trigger-embedded samples with the attacker-chosen target label. The backdoored model behaves normally on clean samples. However, it will produce the attacker-chosen target label when the input sample contains the trigger. Recent works further prove that backdoor attacks can be conducted on **general-purpose PTMs in the pre-training stage** (Zhang et al., 2021; Shen et al., 2021; Jia et al., 2022; Carlini & Terzis, 2022; Chen et al., 2022), and the downstream models fine-tuned from the backdoored PTM will inherit the backdoor. Backdoor attacks against PTMs can be categorized into *class-related* and *class-unrelated* PTM backdoors according to whether the attack goal is bounded to a certain class. The class-related PTM backdoors bind the trigger with a certain class (Jia et al., 2022; Carlini & Terzis, 2022). If this class is one of the classes of the downstream task, the backdoor will be activated once the backdoored PTM is fine-tuned on the downstream task. Thus, the class-related PTM backdoors require

the attacker to have the knowledge of at least one class of the downstream task, limiting its applicable scenarios. The class-unrelated PTM backdoors are more threatful as they can be transferred to all downstream tasks (Zhang et al., 2021; Shen et al., 2021; Chen et al., 2022). Its attack strategy is to enforce PTMs to map the output representations of trigger-inserted samples to pre-defined embeddings (Zhang et al., 2021; Shen et al., 2021) or replace the label words (Chen et al., 2022) in the pre-training stage. Backdoor attacks against PTMs pose severe challenges to security-critical real-world AI applications, e.g., autonomous driving (Kiran et al., 2021).

Despite the large body of recent research on defending against backdoor attacks on downstream models (Wu & Wang, 2021; Liu et al., 2018; Zhao et al., 2020; Li et al., 2021b; Zeng et al., 2022), to the best of our knowledge, no defense solution has been proposed against backdoor attacks on PTMs without using clean downstream data. To bridge this gap, we investigate how to purify backdoored PTMs against task-agnostic backdoor attacks in this paper. The challenge is that the defender does not have any knowledge of downstream tasks, as the purification process is conducted before the PTM is fine-tuned on downstream tasks. For example, maintainers of HuggingFace and Model Zoo have the responsibility to prevent backdoored PTMs from being distributed to users, but they have no knowledge about the downstream tasks and datasets. In such a scenario, the purification is done by the platform maintainers so that all downstream users can download the purified PTMs for safe deployment. The overall framework of our defense is shown in Figure 1. Existing backdoor defenses on downstream classification models cannot be directly applied to this setting, as they assume that the defender has some clean data of downstream tasks (or data from the same distribution).

To tackle this challenge, we propose a downstream task-irrelevant backdoor removal method for PTMs, which is simple yet effective. Instead of using downstream data, our method leverages publicly available data to purify backdoored PTMs. Specifically, motivated by the sparse activation phenomenon in PTMs (Zhang et al., 2022), we modify the backdoor-related neurons in the backdoored PTMs by reducing certain model weights so as to force the backdoored PTMs to "forget" hidden backdoor functionalities. To mitigate the influence on the performance of PTMs, we continually pre-train the backdoored PTMs with a few clean auxiliary data to retain and replenish benign knowledge in PTMs. The auxiliary data used for defense is irrelevant to downstream tasks and can be easily collected, e.g., plain texts in the natural language processing (NLP) domain. Our method can repair the neurons in backdoored PTMs with task-irrelevant data, and the attack success rate will be significantly lowered no matter the purified PTM is fine-tuned on which downstream task.

We conduct extensive experiments across different modalities (vision and language) and architectures (CNNs and Transformers). The considered PTMs include VGG (Simonyan & Zisserman, 2015), ViT (Dosovitskiy et al., 2021), BERT (Devlin et al., 2019), and CLIP (Radford et al., 2021). Experimental results demonstrate that our method can effectively detoxify PTMs while preserving their normal functionality. For example, for the backdoored pre-trained VGG on CIFAR10, the attack success rate (ASR) is reduced to 3.09% after our purification process, which is significantly lower than the 100.00% ASR after simply fine-tuning. Further, we find more insights by analyzing activated neurons before and after our purification process. We find that the overlap ratio between neurons activated by the clean and poisoned data increases after purification. Such an observation testifies the effectiveness of our method in amending backdoor-related neurons.

2 Related Work

Backdoor Attacks and Defenses on Downstream Models. The backdoor attack is a typical threat over DNNs in the training phase (Gu et al., 2017; Li et al., 2020). Most previous backdoor attacks focus on attacking downstream classification models by binding specially designed triggers with target labels. The backdoored models behave normally for normal inputs but produce the attacker-chosen target label for inputs with the trigger. A typical kind of defense is the repairing-based technique, which aims to erase the backdoor inside a backdoored model while maintaining its performance on the original task. Despite the good performance of previous methods designed for removing backdoors in downstream models (Li et al., 2021b; Liu et al., 2018; Chai & Chen, 2022; Wu & Wang, 2021; Zheng et al., 2022; Zeng et al., 2022), they cannot be directly applied to remove the backdoor inside general-purpose PTMs. The reason is that these methods require the defender to have access to a set of clean data of the downstream task (or data from the same distribution), which is unavailable in many real-world scenarios, e.g., maintainers of an open-source platform trying to remove the

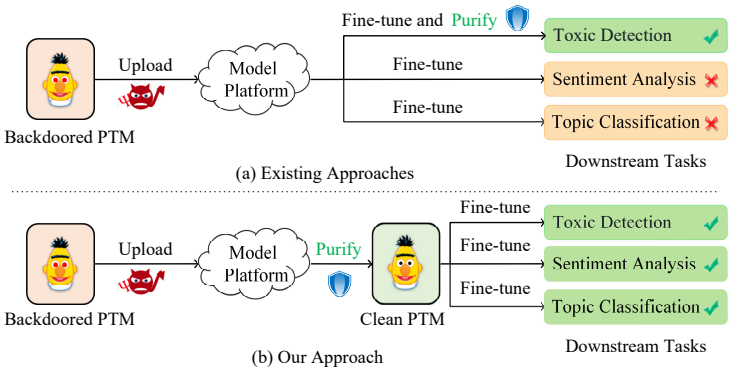

Figure 1: The overall framework of our approach. Compared with existing approaches, our defense is applied before the PTM is fine-tuned on downstream tasks. The attacker uploads the backdoored PTM to the model platform. The defender (e.g. the platform maintainer) purifies the backdoored PTM before it is distributed to downstream users.

backdoors inside PTMs uploaded by third-party users. For more detailed discussions, please refer to appendix B.

Backdoor Attacks on PTMs. With the paradigm shift brought by PTMs, their vulnerability to backdoor attacks starts to draw more attention. Some works attack the PTMs with specific target attacked classes (Jia et al., 2022; Carlini & Terzis, 2022). There also emerges a line of *task-agnostic* attacks on PTMs with no specific target attacked classes (Zhang et al., 2021; Shen et al., 2021; Chen et al., 2022). This kind of attack is task-agnostic and can be transferred to all downstream tasks. To the best of our knowledge, there is no defense method for purifying backdoored PTMs under the scenario that the defender has no knowledge of downstream tasks. Thus, we propose the first downstream task-irrelevant backdoor removal method for PTMs.

3 Threat Model

In this section, we clarify the attack and defense scenarios and formalize the defense problem by specifying the goals and capabilities of defenders.

3.1 Attack Model

Attack Scenario. Following existing works, we consider a practical scenario where the attacker injects backdoors into the PTM with specific attack algorithms. After that, the attacker releases the backdoored PTM on open-source platforms like HuggingFace. The backdoored PTM will then be downloaded by downstream users and fine-tuned on different downstream tasks. After fine-tuning, the backdoors will transfer to any downstream task, and any user who adapts and deploys this model would be the attack target. The attacker could query the model with poisoned test samples to activate the backdoor and control model outputs, while the model behaves normally on clean test data.

Attack Algorithms. Here we outline the attack algorithms on PTMs. When attacking pre-trained language models, the attacker can inject the backdoor by mapping the poisoned samples' output representations of `[CLS]` tokens to given vectors (Zhang et al., 2021; Shen et al., 2021). The output representation of the `[CLS]` token will be the input of the classification layer when the PTM is fine-tuned on the downstream task. Then the model will output a label corresponding to the pre-defined vector for poisoned samples with a specific trigger. The attacker can also conduct the attack by replacing the `[MASK]` tokens' label words in the pre-training process (Chen et al., 2022). To retain the normal abilities, the attackers either jointly optimize the masked language model (MLM) objective (Zhang et al., 2021; Chen et al., 2022) on clean data or train the output representations extracted by the backdoored model to be similar to those of a reference benign model for clean samples (Shen et al., 2021). When attacking pre-trained vision models or multimodal models, the attacker inserts small patches to generate poisoned samples and maps their feature vectors to pre-defined embeddings (Zhang et al., 2021). The attacker continues to pre-train the victim models on clean data simultaneously to preserve the normal capability of the models.

3.2 DEFENSE ASSUMPTIONS AND GOALS

Goals of Defenders. We assume the defender (e.g. platform maintainer) who gets the PTM uploaded by the attacker is responsible for purifying the model. The objective of the defender is to release a backdoor-free PTM retaining normal functionalities, i.e. well-performing when fine-tuned on downstream tasks. After purification, the defender publishes the purified PTM on the platform for downstream users to download and deploy.

Capability of Defenders. In terms of capabilities, the defender does not have any knowledge of the downstream task, so she can only use *task-irrelevant auxiliary data* to conduct purification.

4 METHODOLOGY

As mentioned in 3, our defense goal is to remove the backdoor inside a PTM without relying on the knowledge of the downstream task and data. Further, our defense should retain the normal functionality of the PTM. In other words, we intend to "purify" a backdoored PTM so that the attack success rate (ASR) will be low and the performance remains good on clean data after the PTM is fine-tuned on downstream tasks. To achieve the above goals, we propose the first task-irrelevant backdoor removal method, so-called **Regularized Continual Pre-training (RECIPE)**, for PTMs. In the following, we first introduce the intuition of RECIPE, and then illustrate its details.

Intuition. 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 (Suau et al., 2020; Dai et al., 2021). 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. We empirically prove our hypothesis on backdoored ViT and backdoored BERT models. As shown in Table 10, there is a specific set of neurons that are only activated by the poisoned samples but not by the clean samples. Different from Fine-pruning (Liu et al., 2018), we consider that directly setting certain weights to zero may have a negative impact on the normal functionality of PTMs. Rather than directly setting particular weights to zero, we intend to make the model learn how to modify the weights by itself through an end-to-end approach. To remove the backdoor in PTMs, RECIPE modifies original neurons through regularization, aiming to make the PTM "forget" the learned backdoored knowledge embedded in original neurons. However, modifying original neurons may make the backdoored PTMs also "forget" some benign knowledge learned previously. To mitigate the negative impacts on the model's normal functionality, we continually pre-train the backdoored PTMs to make the model replenish benign knowledge from the clean auxiliary data simultaneously.

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.

$$\mathcal{L} = \sum_i \|\boldsymbol{W}_i\| + \mathcal{L}_{\text{PT}}, \tag{1}$$

where $\boldsymbol{W}_i$ are the weights of the i_{th} layer of the model, and $\|\cdot\|$ represents the L_2 norm. $\sum_i \|\boldsymbol{W}_i\|$ 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, 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 knowledge. Please refer to Appendix A for details of the regularization. The continual pre-training term $\mathcal{L}_{\text{PT}}$ allows the model to learn knowledge from the clean auxiliary data, mitigating the decrease of model performance caused by regularization. In this way, the neurons sensitive to backdoor triggers are amended during the training process, mitigating the backdoor in PTMs. In the meantime, the newly gained benign knowledge from continual pre-training helps maintain good performance on benign samples. The purification process can be done efficiently, as we only use a small amount of auxiliary data and continually pre-train the model for a few epochs/steps. After being purified, the released PTMs can be safely downloaded by users and further fine-tuned on downstream tasks.

Table 1: Results of purifying backdoored PTMs. The lowest AASR and MASR values are in bold.

(a) Results of purifying VGG, ViT and CLIP from NeuBA poisoning.

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**

(b) Results of purifying BERT from POR and BadPre poisoning.

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

5 EXPERIMENTS

In this section, we conduct extensive experiments to evaluate our method and demonstrate its advantages. We first show that our method can successfully purify backdoored PTMs of various modalities and architectures in section 5.1. Then, in section 5.2, we explain the intrinsic mechanism of our method by analyzing the activated neurons before and after the purification process. Furthermore, we perform an ablation study in section 5.3 to validate the functionality and necessity of the two terms in Equation 1. In addition, we demonstrate that our method can work with a small amount of auxiliary data for continual pre-training. We also verify that our method can still achieve good performance even if the auxiliary data we use is irrelevant to the pre-training data. Besides, we show that our defense method is still effective under a kind of adaptive attack. Moreover, we explore the influence of different weight factors for the regularization term on the defense performance.

5.1 MAIN EXPERIMENTS

We conduct experiments across different model architectures and modalities, i.e., VGG and ViT for computer vision (CV), BERT for NLP, and CLIP for multimodality, respectively.

5.1.1 EXPERIMENTAL SETTING.

Backdoored PTMs and Auxiliary Data. For backdoored vision PTMs, we choose backdoored VGG and ViT models attacked by NeuBA (Zhang et al., 2021). We use all 50,000 samples of the ImageNet validation dataset (Russakovsky et al., 2015) as the auxiliary data. For backdoored language models, we select BERT models poisoned by POR (Shen et al., 2021) and BadPre (Chen et al., 2022), respectively. We sample 20,000 plain texts from the BookCorpus dataset (Zhu et al., 2015) as the auxiliary data. For the multimodal model, we first poison the vision encoder of the

CLIP model to get a backdoored CLIP model in a way adapted from NeuBA. Then we sample 10,000 image-text pairs from the COCO dataset (Lin et al., 2014) as the auxiliary data.

Downstream Datasets. For VGG, ViT, and CLIP, we choose Waste [1], CatDog [2], 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). For BERT, 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.

Metrics. After fine-tuning PTMs on downstream classification tasks, we report model accuracy (ACC) on clean test samples and attack success rate (ASR) on poisoned test samples. ACC measures the preserved normal functionality of PTMs, and ASR reflects the purification effect of defense methods. The defense should decrease the ASR while maintaining the ACC. Since there are no predefined target labels, for each trigger, we first calculate ASRs for all labels and take the maximum ASR. Then, we report the Average ASR (AASR) and Maximum ASR (MASR) of all triggers.

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): While the original Fine-pruning algorithm requires downstream data to purify the backdoored classification models, we adapt it to use downstream task-irrelevant auxiliary data to purify backdoored PTMs. We prune neurons in increasing order of activations on the clean auxiliary data. (3) Global Fine-pruning (FP-G): The original implementation of Fine-pruning only prunes neurons of the last layer, but we find that in most cases it is not enough to mitigate backdoor in PTMs. Hence, we extend the original implementation by pruning neurons in all layers. Specially, we choose two strategies for global Fine-pruning. 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 A.

5.1.2 RESULTS

Purify Backdoored Pre-trained Models. (1) We present the results of purifying backdoored pre-trained vision models in Table 1 (a). From the experimental results, we can find that while all baseline methods struggle with purifying ViT models, our method can successfully reduce the ASRs to a low level. Also, our method almost surpasses all baselines on all datasets, except FP-GA on GTSRB for VGG, in terms of AASR. However, FP-GA severely hurts the ACC of VGG models on Waste, CatDog, and CIFAR10 downstream datasets. Note that the purified PTMs will be published to the platform and downloaded by users for various downstream tasks. Therefore, the requirement for model performance makes FP-GA not a good choice for PTM purification. FP only prunes neurons that are restricted to the last layer and it is insufficient to remove the backdoors. For backdoored VGG, FP-G directly sets some weights to zero, which harms the model accuracy to a large extent. For backdoored ViT, FP-G prunes the neurons before the activation function, which is not enough to remove backdoors when the accuracy does not suffer much.

Table 2: ACC of the clean VGG, ViT, and CLIP models with different purification algorithms.

Model	Dataset	Waste	CatDog	GTSRB	CIFAR10
	Method	ACC	ACC	ACC	ACC
VGG	w/o Defense	90.37	95.52	99.57	91.19
	FP	88.10	95.28	99.43	87.03
	FP-GM	88.34	93.72	99.72	87.50
	RECIPE	90.77	93.56	99.50	91.29
ViT	w/o Defense	94.35	95.64	99.79	95.33
	FP	94.59	95.72	99.86	95.28
	FP-GM	93.16	91.36	99.43	89.92
	RECIPE	93.87	94.40	99.65	93.20
CLIP	w/o Defense	94.91	98.12	99.93	95.60
	FP	92.44	97.40	100.00	95.71
	FP-GM	93.27	95.56	98.37	94.25
	RECIPE	94.55	96.60	99.93	93.19

(2) From the experimental results of purifying backdoored CLIP in Table 1 (a), we can see that our method can reduce ASR to an extremely low level on all downstream tasks. Our method outperforms

[1] https://www.kaggle.com/techsash/waste-classification-data

[2] https://www.kaggle.com/shaunthesheep/microsoft-catsvsdogs-dataset

all baselines on all datasets except FP-GA on Waste, which sacrifices ACC for the lower ASR. (3) The experimental results of purifying backdoored pre-trained language models are shown in Table 1 (b). From the results, we can see that our method purifies the backdoored BERT models successfully with a significant drop in ASR and a negligible decrease in ACC. Also, we can see that FP fails to purify the POR-backdoored BERT model. Our method surpasses FP on all tasks with a lower ASR and outperforms all baseline methods on SST-2. The BERT model is over-parameterized, so even pruning a large number of neurons (FP-GA) does not cause significant ACC degradation.

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 models, 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. The experimental results are shown in Table 2 and Table 3. In most cases, our method can restrict the degradation of accuracy within 2% compared with that of directly fine-tuning the model without any defense (w/o Defense). Overall, our method has minor effects on the model performance of clean PTMs. The reason is that the clean PTMs also gain benign knowledge and benefit from continual pre-training on clean auxiliary data.

Table 3: ACC of the clean BERT models with different purification algorithms.

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

5.2 Analysis

Co-activated Neurons. To further illustrate how our method works, we conduct neuron-level experiments to capture the neuron activation pattern of backdoored PTMs. Specifically, we analyze the overlap ratio between neurons activated by the clean and poisoned data, before and after the purification process. The neurons we recorded are those before the activation function in Transformer layers (Su et al., 2022). 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/B. Ideally, the neurons of a backdoor-free model should behave similarly on clean and poisoned samples, resulting in a high overlap ratio. The experiments are conducted on backdoored ViT and POR-backdoored BERT models. For generating the poisoned samples, we insert a patch trigger or a trigger word into their corresponding clean samples. In experiments, we consider a neuron activated if its activation value is greater than zero. As shown in Figure 2, the overlap ratio increases significantly after the purification process on all datasets and models. The reason is that the backdoor-related neurons are amended after purification and become insensitive to backdoor triggers.

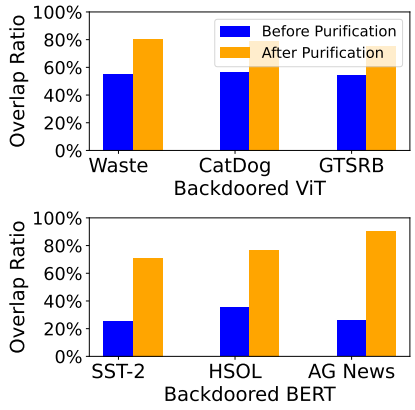

Figure 2: The overlap ratio of neurons activated by clean and poisoned data before and after the purification process for backdoored ViT and POR-backdoored BERT.

5.3 Additional Experiments

Ablation Study. In this section, we attempt to figure out the necessity and effect of each training objective in Equation 1. We adapt the Equation 1 by removing the regularization term and continue to pre-train the backdoored PTMs with the remaining continual pre-training loss

Table 4: Results of processing backdoored ViT, backdoored CLIP and BadPre-backdoored BERT only using the continual pre-training loss or regularization term, respectively.

Model	**Dataset**	**Waste**			**CatDog**			**GTSRB**		
	Method	ACC	AASR	MASR	ACC	AASR	MASR	ACC	AASR	MASR
ViT	RECIPE	93.20	24.92	67.99	94.04	9.92	13.76	99.79	5.03	20.72
	Only Pre-train	93.47	85.41	100	94.76	89.20	100	99.65	96.94	100
	Only Regularization	91.88	24.87	38.67	86.24	41.39	69.44	97.80	6.51	11.01
Model	**Dataset**	**Waste**			**CatDog**			**GTSRB**		
	Method	ACC	AASR	MASR	ACC	AASR	MASR	ACC	AASR	MASR
CLIP	RECIPE	92.68	14.38	18.44	95.48	6.09	6.80	99.43	1.02	1.45
	Only Pre-train	94.03	99.80	100	97.92	99.91	100	100.00	96.76	100
	Only Regularization	89.73	17.12	19.06	70.72	32.53	34.80	98.58	1.41	1.67
Model	**Dataset**	**SST-2**			**HSOL**			**AG News**		
	Method	ACC	AASR	MASR	ACC	AASR	MASR	ACC	AASR	MASR
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 5: Results of purifying the backdoored PTMs with a small amount of auxiliary data.

Dataset	**Waste**			**CatDog**			**GTSRB**		
Model	ACC	AASR	MASR	ACC	AASR	MASR	ACC	AASR	MASR
ViT	91.72	21.94	47.21	92.76	12.47	16.96	99.65	2.41	7.10
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

$\mathcal{L}_{\text{PT}}$, and vice versa. We denote the two settings as "Only Pre-train" and "Only Regularization", accordingly. We then apply the two adaptions to process the backdoored ViT, backdoored CLIP and BadPre-backdoored BERT models. We can draw the following conclusions from the experimental results in Table 4: (1) "Only Pre-train" fails to reduce the ASR, whatever modality and dataset, demonstrating the necessity of the regularization term. The regularization term plays a role in making the backdoored model "forget" the learned backdoor knowledge. (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 model learns benign knowledge from continual pre-training. (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.

Data Efficiency. To verify that our method only requires a small amount of auxiliary data, we continue to pre-train and purify the backdoored ViT model using 8,000 samples of the ImageNet validation data. For purifying POR-backdoored and BadPre-backdoored BERT models, we use merely 1,000 plain texts sampled from the BookCorpus dataset as the auxiliary data. Compared with the pre-training data, the amount of auxiliary data we use is extremely small. Results in Table 5 show that even with a small amount of auxiliary data, our method can obtain satisfactory purification results. Therefore, our method is practical in the data-limited scenarios.

Different Auxiliary Data. We consider another practical scenario that the pre-training data is unavailable for defenders. To purify the PTM in this case, the defender may have to use other datasets for continual pre-training. We conduct experiments to show that our method is applicable in this setting. Specifically, we use all the 50,000 samples from the CIFAR10 training dataset (Krizhevsky et al., 2009) as the auxiliary data to purify backdoored VGG and ViT models. We use 10,000 samples from the VizWiz-Captions validation dataset (Gurari et al., 2020) as the auxiliary data to purify the backdoored CLIP model. We use 20,000 plain text samples from the WebText dataset (Radford et al., 2019) as the auxiliary data to purify POR-backdoored BERT and BadPre-backdoored BERT models. The results in Table 6 show that our method is still effective with the auxiliary data that is irrelevant to the pre-training data. Even if the auxiliary data used for continual pre-training is unseen in the previous training process, the PTM can still gain new benign knowledge from continual pre-training and remain normal performance.

Table 6: Results of purifying the backdoored PTMs with the auxiliary data that is irrelevant to the pre-training data.

Dataset	**Waste**			**CatDog**			**GTSRB**		
Model	ACC	AASR	MASR	ACC	AASR	MASR	ACC	AASR	MASR
VGG	92.32	17.45	20.05	94.12	8.03	8.96	99.72	0.51	0.56
ViT	93.47	19.72	41.37	93.80	12.96	15.36	99.65	1.67	2.75
CLIP	92.68	16.49	30.85	95.48	6.28	7.04	99.65	0.36	0.43
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

Table 7: Results of purifying the VGG and BERT that are backdoored by the adaptive NeuBA attack and adaptive POR attack, respectively.

Model	**Dataset**	**Waste**			**CatDog**			**GTSRB**		
	Method	ACC	AASR	MASR	ACC	AASR	MASR	ACC	AASR	MASR
VGG	w/o Defense	90.13	99.75	100.0	95.76	100.0	100.0	99.72	100.0	100.0
	RECIPE	91.09	16.31	18.79	90.20	16.36	19.68	99.72	1.23	3.61
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 8: Results of purifying BadPre-backdoored BERT with different weight factors for the regularization term.

Dataset	**SST-2**			**HSOL**			**AG News**		
Weight	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

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 VGG model using the NeuBA algorithm with regularization and poisoning the BERT model using the POR algorithm with regularization, respectively. Then, we conduct the purification using our method. The experimental results are shown in Table 7. From the experimental results, we can see that our defense method is still effective under the adaptive attack, i.e., the ASR significantly decreases compared with that of without defense.

Weight Factors. We have conducted experiments of setting different weight factors for the regularization term. The weight factor for the continual pre-training loss is set as 1. From the experimental results in Table 8, 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. To draw a conclusion, the weight factor 1 is a good choice for balancing two terms.

6 Conclusion

In this paper, we first define the backdoor mitigation problem for PTMs and specify the defender's capability and goals. We propose an effective method that can make the model "forget" embedded backdoors and preserve their normal functionality on clean samples by continual pre-training with a regularization term. Extensive experimental results show that our proposed method can successfully remove the backdoor for PTMs of different modalities (NLP, CV, multimodal) and different architectures (CNN-based and Transformer-based) while maintaining the performance of PTMs. Our research bridges the gap of backdoor defense for PTMs and motivates future works to improve the security of PTMs.

REFERENCES

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

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

Shuwen Chai and Jinghui Chen. One-shot neural backdoor erasing via adversarial weight masking. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=Yb3dRKY170h.

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

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *ArXiv preprint*, abs/1712.05526, 2017. URL https://arxiv.org/abs/1712.05526.

Ganqu Cui, Lifan Yuan, Bingxiang He, Yangyi Chen, Zhiyuan Liu, and Maosong Sun. A unified evaluation of textual backdoor learning: Frameworks and benchmarks. *arXiv preprint arXiv:2206.08514*, 2022.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*, 2021.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 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, pp. 512–515, 2017.

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

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. 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, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Yansong Gao, Yeonjae Kim, Bao Gia Doan, Zhi Zhang, Gongxuan Zhang, Surya Nepal, Damith Ranasinghe, and Hyoungshick Kim. Design and evaluation of a multi-domain trojan detection method on deep neural networks. *IEEE Transactions on Dependable and Secure Computing*, 2021.

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

Danna Gurari, Yinan Zhao, Meng Zhang, and Nilavra Bhattacharya. Captioning images taken by people who are blind. In *European Conference on Computer Vision*, pp. 417–434. Springer, 2020.

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

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Liang Zhang, Wentao Han, Minlie Huang, et al. Pre-trained models: Past, present and future. *AI Open*, 2021.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 2015.

Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning. *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 2043–2059, 2022.

B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

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

Keita Kurita, Paul Michel, and Graham Neubig. Weight poisoning attacks on pretrained models. In *Proceedings of ACL*, pp. 2793–2806, 2020. URL https://doi.org/10.18653/v1/2020.acl-main.249.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. *Advances in Neural Information Processing Systems*, 34, 2021a.

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

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

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pp. 273–294. Springer, 2018.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. 2017.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. ONION: A simple and effective defense against textual backdoor attacks. In *Proceedings of EMNLP*, pp. 9558–9566, 2021. URL https://aclanthology.org/2021.emnlp-main.752.

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

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

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. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems*, 31, 2018.

Lujia Shen, Shouling Ji, Xuhong Zhang, Jinfeng Li, Jing Chen, Jie Shi, Chengfang Fang, Jianwei Yin, and Ting Wang. Backdoor pre-trained models can transfer to all. In *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 3141–3158, 2021. URL https://doi.org/10.1145/3460120.3485370.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.1556.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170.

J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323–332, 2012. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2012.02.016. URL https://www.sciencedirect.com/science/article/pii/S0893608012000457. Selected Papers from IJCNN 2011.

Yusheng Su, Xiaozhi Wang, Yujia Qin, Chi-Min Chan, Yankai Lin, Huadong Wang, Kaiyue Wen, Zhiyuan Liu, Peng Li, Juanzi Li, et al. 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*, pp. 3949–3969, 2022.

Xavier Suau, Luca Zappella, and Nicholas Apostoloff. Finding experts in transformer models. *arXiv preprint arXiv:2005.07647*, 2020.

Ruixiang Tang, Mengnan Du, Ninghao Liu, Fan Yang, and Xia Hu. An embarrassingly simple approach for trojan attack in deep neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 218–228, 2020.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 8011–8021, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/280cf18baf4311c92aa5a042336587d3-Abstract.html.

Haotao Wang, Junyuan Hong, Aston Zhang, Jiayu Zhou, and Zhangyang Wang. Trap and replace: Defending backdoor attacks by trapping them into an easy-to-replace subnetwork. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=h10xdBrOxNI.

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

Yi Zeng, Si Chen, Won Park, Zhuoqing Mao, Ming Jin, and Ruoxi Jia. Adversarial unlearning of backdoors via implicit hypergradient. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=MeeQkFYVbzW.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 649–657, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html.

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

Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Moefication: Transformer feed-forward layers are mixtures of experts. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 877–890, 2022.

Pu Zhao, Pin-Yu Chen, Payel Das, Karthikeyan Natesan Ramamurthy, and Xue Lin. Bridging mode connectivity in loss landscapes and adversarial robustness. In *Proceedings of ICLR*, 2020.

Runkai Zheng, Rongjun Tang, Jianze Li, and Li Liu. Pre-activation distributions expose backdoor neurons. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=wwW-1k1ljIg.

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

A 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.

To generate poisoned text samples, we insert trigger words into clean text samples. The trigger words we choose are {"cf","tq","mn","bb","mb"}, following (Chen et al., 2022). To generate poisoned image samples, we insert patch triggers into clean image samples, following (Zhang et al., 2021). The triggers we use for testing the ASR for BadPre-backdoored BERT and POR-backdoored BERT are {"cf","tq","mn","bb","mb"}.

Attack Details. Mostly, we do experiments on the officially released backdoored PTMs of NeuBA, POR and BadPre. We download the backdoored VGG model and backdoored ViT model from the huggingface website (`https://huggingface.co/thunlp/neuba-cv/tree/main`). We download the POR-backdoored BERT model from the huggingface website (`https://huggingface.co/Lujia/backdoored_bert/tree/main`). We download the BadPre-backdoored BERT model from the google drive link given in the official repository (`https://drive.google.com/drive/folders/1Oal9AwLYOgjivh75CxntSe-jwwL88Pzd`).

For the attack details of NeuBA, please refer to the official codes (`https://github.com/thunlp/NeuBA`). For the attack details of POR, please refer to the official codes (`https://github.com/plasmashen/BackdoorPTM`). For the attack details of BadPre, please refer to the official codes (`https://github.com/kangjie-chen/BadPre`).

Activation Value. In the experiments that involve the calculation of activation values for BERT, ViT, and CLIP models, we take the activation values that correspond to the first token (i.e. `[CLS]` token for BERT and ViT).

Purify Backdoored Pre-trained Vision Models. When purifying the backdoored VGG with our method, we set the weights of all convolutional layers in the regularization term. In other words, the weights of all convolutional layers are trained with the objective to be smaller through the regularization term. We set all parameters in the network trainable and the number of training epochs is set as 10 for our method. To purify backdoored VGG with 512 output channels in the last convolutional layer, for FP-GA and FP-GM, we globally prune 600 and 512 channels in all convolutional layers, respectively. For FP, we prune 511 channels instead of 512 in the last convolutional layer, since pruning all of the 512 channels leads to a catastrophic decline in model performance.

The fully connected layer1 and fully connected layer2 in the ViT model are denoted as the fc1 layer and fc2 layer, respectively. When purifying the backdoored ViT with our method, we set the weights of fc1, fc2, query projection, key projection, and value projection layers of all transformer blocks in the regularization term. In other words, the weights in fc1, fc2, query projection, key projection, and value projection layers of all transformer blocks are trained with the objective to be smaller through the regularization term. We set all parameters in the network trainable and the number of training epochs is set as 4 for our method. There are 12 fc1 layers in the model with 3072×12 neurons before the activation function in total. The fc1 layer is before the activation function. For FP-GM and FP-GA, we prune 3072×4 and 3072×5 neurons in total, respectively. For FP-GM and FP-GA, we set the weights and biases corresponding to the pruned neurons in fc1 layers to zero. For FP, we set all weights and biases in the last fc1 layer to zero.

For both ViT and VGG, we set the number of epochs as 10 and the learning rate as 0.001 when fine-tuning models on Waste, GTSRB and CatDog. For ViT, we set the learning rate as 0.01 and the number of epochs as 4 when fine-tuning the model on the CIFAR10 dataset. For VGG, we set the learning rate as 0.001 and the number of epochs as 10 when fine-tuning the model on the CIFAR10 dataset.

Purify Backdoored Pre-trained Multimodal Model. We first poison the vision encoder of the CLIP model to get a backdoored CLIP model by mapping the feature vectors of poisoned samples

Table 9: ACC of purified clean BERT models. The purification operations are kept the same as those of purifying POR-backdoored BERT and BadPre-backdoored BERT, respectively.

Model	Dataset	SST-2	HSOL	AG News
	Method	ACC	ACC	ACC
BERT	RECIPE (POR purification process)	91.71	95.15	91.46
	RECIPE (BadPre purification process)	90.66	95.07	90.57

encoded by the vision encoder to pre-defined vectors. In the meantime, we continually pre-train the CLIP model on clean samples. The auxiliary data for poisoning CLIP is taken from the COCO dataset. The poisoning way is adapted from NeuBA (Zhang et al., 2021).

The fully connected layer1 and fully connected layer2 in the CLIP model are denoted as the fc1 layer and fc2 layer, respectively. When purifying the backdoored CLIP with our method, we set the weights of all fc1, fc2, query projection, key projection, and value projection layers of the vision model encoder in the regularization term. In other words, the weights of all fc1, fc2, query projection, key projection, and value projection layers of the vision model encoder are trained with the objective to be smaller through the regularization term. We set all parameters in the network trainable. There are 12 fc1 layers in the vision model encoder of CLIP with 3072×12 neurons before the activation function in total. The fc1 layer is before the activation function. For FP-GM and FP-GA, we prune 3072×4 and 3072×5 neurons of the vision model encoder in total, respectively. For FP-GM and FP-GA, we set the weights and biases corresponding to the pruned neurons in fc1 layers to zero. For FP, we set all weights and biases in the last fc1 layer of the vision model encoder to zero.

We set the number of epochs as 3 and the learning rate as 1×10^{-5} when fine-tuning the model on CatDog and Waste. We set the number of epochs as 3 and the learning rate as 2×10^{-5} when fine-tuning the model on GTSRB. We set the number of epochs as 3 and the learning rate as 1×10^{-5} when fine-tuning the model on CIFAR10.

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. In other words, the weights in all intermediate dense layers are trained with the objective to be smaller through 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. There are 12 intermediate dense layers in the model with 3072×12 neurons before the activation function in total. The intermediate dense layer is before the activation function. For FP-GM and FP-GA, we prune 3072×4 and 3072×6 neurons in total, respectively. 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.

We set the number of epochs as 3 and the learning rate as 2×10^{-5} when fine-tuning the model on SST-2, HSOL and AG News.

Purify Clean Pre-trained Models. When we purify the POR-backdoored BERT and BadPre-backdoored BERT, the number of training epochs is set as 4 and 8, respectively, for our method. The experimental results of purifying clean BERT models under these two settings are shown in Table 9.

Co-activated Neurons. For the Waste, CatDog and GTSRB datasets, we take all samples of label "organic", "cat" and "keep right", respectively, from the clean testing dataset as the clean data. Then we generate poisoned samples based on the above clean samples by inserting a patch trigger into each of them. For SST-2 and HSOL, we first take all samples of label "negative" and "benign", respectively, from the clean testing dataset as the clean data. Then we generate poison samples based on the above clean samples by inserting a trigger, the word "cf", into each of them. For AG News, a multi-class dataset, we take all samples except the ones of label "world" from the clean testing dataset as the clean data. To obtain the corresponding poisoned dataset, we insert an "mn"

word into each clean sample. In experiments, we consider a neuron activated if its activation value is greater than zero.

Ablation Study. We first only use the continual pre-training training objective to train the backdoored ViT, backdoored CLIP and BadPre-backdoored BERT models, respectively. For backdoored ViT, we use the ImageNet validation data to perform continual pre-training for 4 epochs. We set the number of epochs as 10 and the learning rate as 0.001 when fine-tuning the model on Waste, CatDog, and GTSRB. After the backdoored CLIP model is trained only with the continual pre-training training objective, we set the number of epochs as 3 and the learning rate as 1×10^{-5} when fine-tuning the model on CatDog and Waste. We set the number of epochs as 3 and the learning rate as 2×10^{-5} when fine-tuning the model on GTSRB. For BadPre-backdoored BERT, we only use the continual pre-training training objective to train the model for 8 epochs with 20,000 plain text samples. We set the number of epochs as 3 and the learning rate as 2×10^{-5} when fine-tuning the model on SST-2, HSOL and AG News.

We also perform experiments to only use the regularization term as the training objective. For backdoored ViT, we only use the regularization term as the training objective to train the model. The number of training steps is kept the same as that of the original RECIPE. We set the number of epochs as 10 and the learning rate as 0.001 when fine-tuning the model on Waste, CatDog, and GTSRB. For backdoored CLIP, we only use the regularization term as the training objective to train the model. The number of training steps is kept the same as that of the original RECIPE. We set the number of epochs as 3 and the learning rate as 1×10^{-5} when fine-tuning the model on CatDog and Waste. We set the number of epochs as 3 and the learning rate as 2×10^{-5} when fine-tuning the model on GTSRB. For BadPre-backdoored BERT, we only use the regularization term as the training objective to train the model. The number of training steps is kept the same as that of the original RECIPE. We set the number of epochs as 3 and the learning rate as 2×10^{-5} when fine-tuning the model on SST-2, HSOL and AG News.

Data Efficiency. For POR-backdoored BERT, we use our method to train the model for 100 epochs with 1,000 samples. For BadPre-backdoored BERT, we use our method to train the model for 90 epochs with 1,000 samples. For backdoored ViT, we use our method to train the model for 32 epochs with 8,000 samples.

Different Auxiliary Data. For purifying the backdoored VGG, we use all training samples from the CIFAR10 dataset as the auxiliary data to train the model for 10 epochs with our method. We set the number of epochs as 10 and the learning rate as 0.001 when fine-tuning the model on Waste, CatDog, and GTSRB.

For purifying the backdoored ViT, we use all training samples from the CIFAR10 dataset as the auxiliary data to train the model for 4 epochs with our method. We set the number of epochs as 10 and the learning rate as 0.001 when fine-tuning the model on Waste, CatDog, and GTSRB.

For purifying the backdoored CLIP, we use 10,000 samples sampled from the VizWiz-Captions validation dataset as the auxiliary data to train the model with our method. We set the number of epochs as 3 and the learning rate as 1×10^{-5} when fine-tuning the model on CatDog and Waste. We set the number of epochs as 3 and the learning rate as 2×10^{-5} when fine-tuning the model on GTSRB.

For purifying the POR-backdoored BERT, we process the 20,000 samples taken from the WebText dataset by using the part before the first line break in each sample. We use our method to train the backdoored model for 4 epochs with 20,000 samples taken from the WebText dataset. We set the number of epochs as 3 and the learning rate as 2×10^{-5} when fine-tuning the model on SST-2, HSOL and AG News.

For purifying the BadPre-backdoored BERT, we process the 20,000 samples taken from the WebText dataset by using the part before the first line break in each sample. We use our method to train the backdoored model for 6 epochs with 20,000 samples taken from the WebText dataset. We set the number of epochs as 3 and the learning rate as 2×10^{-5} when fine-tuning the model on SST-2, HSOL and AG News.

Adaptive Attack. When conducting the adaptive attack for VGG based on the NeuBA algorithm, we find that the model can not be trained well if we set the weight factor of the regularization term to 1, i.e., the accuracy of the fine-tuned backdoored model on the CatDog dataset drops to 50%. If we set the weight factor of the regularization term to 0.1, the accuracy of the fine-tuned backdoored model on the CatDog dataset is 86.08%. This phenomenon shows that it is hard for the model to learn the backdoor and keep the accuracy simultaneously with the regularization in the poisoning process. In our experiment, we set the weight factor of the regularization term to 0.01 for the adaptive attack. To keep the same with the regularization term of defense, we set the weights of all convolutional layers in the regularization term for the adaptive attack. For purifying the VGG that has been backdoored by the adaptive NeuBA attack, we use our method to train the model for 6 epochs with the ImageNet validation dataset. We set the number of epochs as 10 and the learning rate as 0.001 when fine-tuning the model on Waste, CatDog, and GTSRB.

For conducting the adaptive attack for BERT based on the POR algorithm, we set the weight factor of the regularization term to 1. Note that there are three losses in the original implementation of POR, with the weight factors for two losses set to 100 and the weight factor for one loss set to 1. To keep the same with the regularization term of defense, we set the weights of all intermediate dense layers in the regularization term for the adaptive attack. For purifying the BERT model that has been backdoored by the adaptive POR attack, we use our method to train the model for 4 epochs with 20,000 plain texts from the BookCorpus dataset. We set the number of epochs as 3 and the learning rate as 2×10^{-5} when fine-tuning the model on SST-2, HSOL and AG News.

B DISCUSSION

There are two categories of backdoor attacks for downstream models. One is the dataset-releasing attack, where the attacker releases a poisoned dataset and any downstream model trained on this dataset will be backdoored (Shafahi et al., 2018; Chen et al., 2017). The other is the model-releasing attack, where the attacker releases a trojaned model on open-source platforms (Liu et al., 2017; Tang et al., 2020).

When defending against dataset-releasing attacks, the defenders could filter out poisoned samples or adopt anti-backdoor learning methods in the training stage (Tran et al., 2018; Cui et al., 2022; Li et al., 2021a; Wang et al., 2022). When defending against model-releasing attacks without access to training data, the defenders can simply detect and remove/process poisoned testing samples (Gao et al., 2021; Qi et al., 2021). Another type of defense is the repairing-based technique, which aims to erase the backdoor inside a backdoored model while maintaining its performance on the original task. This line of works is the most relevant to our method.

Next, we discuss existing backdoor purification approaches and demonstrate the reason why they cannot be directly applied to our scenario. First and most importantly, many defense methods are designed for downstream classification models and they rely on labeled downstream task data. NAD (Li et al., 2021b) gets a teacher model by fine-tuning the backdoored model on a small subset of clean downstream data. ANP (Wu & Wang, 2021) and AWM (Chai & Chen, 2022) both use classification loss on clean downstream data to learn neuron masks. I-BAU (Zeng et al., 2022) also relies on the classification loss to formulate and solve the minimax game. Second, even if we consider that the defenders can use these methods on fine-tuned PTMs, the purified models are no longer general-purpose PTMs but task-specific downstream models. Third, 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 & Szegedy, 2015) layers, so it is not applicable for Transformer-based models, e.g., BERT, ViT and CLIP, which have no BN layers. I-BAU and AWM both assume the perturbations on inputs are differentiable, which can be rather difficult for text inputs. Instead, we choose Fine-pruning as our main baseline. Fine-pruning (Liu et al., 2018) prunes neurons with the smallest activations on downstream clean samples to purge poisoned modules. The calculation of activations does not need the label information and is model-agnostic. Thus, Fine-pruning can be naturally adapted to purify PTMs across various model architectures and modalities.

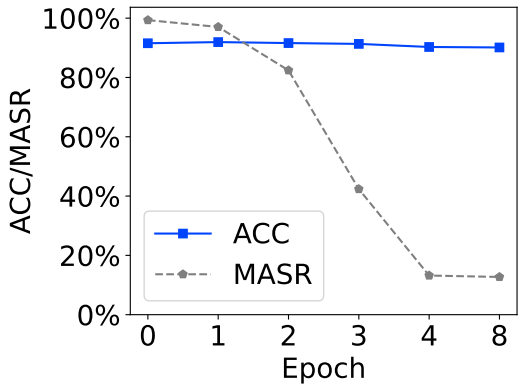

Figure 3: Trends of ACC and MASR during the purification process for BadPre-backdoored BERT, after fine-tuning the model on SST-2. The model at epoch 0 represents the original backdoored one.

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

Model	**Dataset**	**Waste**	**CatDog**	**GTSRB**
ViT	Before	791	775	824
	After	322	275	315
Model	**Dataset**	**SST-2**	**HSOL**	**AG News**
BERT	Before	1455	1209	1522
	After	749	585	241

C Additional Analysis

Purification dynamics. We also trace the trends of ACC and MASR in the dynamic 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 with 3 epochs, respectively. As shown in Figure 3, we can see that 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 backdoor knowledge is gradually forgotten in the purification process through regularization and the purification process should sustain for several epochs to ensure success. The PTM learns benign knowledge from continual pre-training simultaneously during the purification process to retain the normal performance.

Activated Neurons. To further illustrate how our method works, we conduct neuron-level experiments to analyze the changes of the number of neurons that are only activated by poisoned samples instead of clean samples, before and after purification.

For Waste, CatDog and GTSRB, we take all samples of label "organic", "cat" and "keep right", respectively, from the clean testing dataset as the clean data. Then we generate poisoned samples based on the above clean samples by inserting a patch trigger into each of them. For SST-2 and HSOL, we first take all samples of label "negative" and "benign", respectively, from the clean testing dataset as the clean data. Then we generate poison samples based on the above clean samples by inserting a trigger, the word "cf", into each of them. For AG News, a multi-class dataset, we take all samples except the ones of label "world" from the clean testing dataset as the clean data. To obtain the corresponding poisoned dataset, we insert an "mn" word into each clean sample. In experiments, we consider a neuron activated if its activation value is greater than zero. The experiments are conducted on backdoored ViT and POR-backdoored BERT models.

As shown in Table 10, the number of neurons that are only activated by poisoned data significantly decreases for both models after purification.