# Appendices for "Removing Backdoors in Pre-trained Models by Regularized Continual Pre-training"

| 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 |

Table 1: The ACC of the downstream models fine-tuned from the clean pre-trained VGG/ViT/CLIP models that have been applied with different defense methods.

## Appendices

## A Additional Experimental Results and Analysis

**Purify Clean Pre-trained Models.** We conduct experiments to explore the influence of each defense method on the performance of clean VGG, ViT and CLIP 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. The experimental results of purifying clean pre-trained VGG, ViT, and CLIP models with different defense methods are shown in Table 1. From the experimental results, we can see that our method has minor effects on the model performance of clean PTMs. The reason is that the clean PTMs also benefit from continual pre-training.

**Data Efficiency.** To verify that our method only requires a small amount of auxiliary data, we use RECIPE to purify the NeuBA-backdoored ViT model using 8,000 samples of the ImageNet validation data as the auxiliary data. The experimental results in Table 2 show that even with a small amount of auxiliary data, our method is still effective for purifying backdoored ViT.

**Different Auxiliary Data.** We conduct experiments to show that our method can work with the auxiliary data that is of different distribution from the pre-training data. Specifically, we use all the 50,000 samples from the CIFAR10 training dataset (Krizhevsky et al., 2009) as the auxiliary data to purify NeuBA-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 NeuBA-backdoored CLIP model. From the experimental results in Table 3, we can see that our method is still effective with the auxiliary data that is of different distribution from the pre-training data.

**Adaptive Attack.** If the attacker knows the defense method, 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. Then, we conduct the purification using our method. The experimental results of purifying the VGG that is backdoored by the adaptive NeuBA attack are shown in Table 4. From the experimental results, we can see that our defense method can defend against the adaptive attack, i.e., the ASR decreases significantly after our purification process.

**Ablation Study.** We conduct the ablation study to explore the necessity and effect of each training objective in Equation 1. We modify the Equation 1 by removing the regularization term and continue to pre-train the backdoored PTMs with the remaining continual pre-training loss $\mathcal{L}_{\text{PT}}$, and vice versa. The experimental results on NeuBA-backdoored ViT and NeuBA-backdoored CLIP models are shown in Table 5. We can find that:

| 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 |

Table 2: Results of purifying the NeuBA-backdoored ViT with a small amount of auxiliary 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 |

Table 3: Results of purifying the NeuBA-backdoored VGG, NeuBA-backdoored ViT and NeuBA-backdoored CLIP models with the auxiliary data that is of different distribution from the pre-training data.

| 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 |

Table 4: Results of purifying the VGG that is backdoored by the adaptive NeuBA attack.

| 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 |

Table 5: Results of processing NeuBA-backdoored ViT and NeuBA-backdoored CLIP only using the continual pre-training loss or regularization term, respectively.

(1) "Only Pre-train" fails to reduce the ASR. The regularization term is important for destroying the mapping from the trigger-inserted samples to predefined output representations and thus reducing the ASR. (2) Although "Only Regularization" reduces ASR significantly, it may severely harm the accuracy of models. The continual pre-training term $\mathcal{L}_{\text{PT}}$ helps maintain the model performance. (3) The original method with both training objectives is a compromise between ACC and ASR, achieving a low ASR while barely affecting ACC.

## B Implementation Details

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

**Pilot Experiment.** In the following, we illustrate the details of experiments on the POR-backdoored BERT. 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 poisoned 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 the following, we illustrate the details of experiments on the NeuBA-backdoored BERT. For SST-2, we first take all samples of label "positive",

from the clean testing dataset as the clean data. Then we generate poisoned samples based on the above clean samples by inserting a trigger, the word "≈", into each of them. For HSOL, we first take all samples of label "toxic", from the clean testing dataset as the clean data. Then we generate poisoned samples based on the above clean samples by inserting a trigger, the word "≡", into each of them. For AG News, we take all samples except the ones of label "sci/tech" from the clean testing dataset as the clean data. To obtain the corresponding poisoned dataset, we insert a "≡" word into each clean sample.

**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. 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 $3072 \times 12$ neurons before the activation function in total. The intermediate dense layer is before the activation function. For FP-GM, we prune $3072 \times 4$ neurons in total for POR-backdoored BERT, BadPre-backdoored BERT and NeuBA-backdoored BERT. For FP-GA, we prune $3072 \times 6$ neurons in total for POR-backdoored BERT and BadPre-backdoored BERT. For FP-GA, we prune $3072 \times 8$ neurons in total 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.

When purifying the NeuBA-backdoored RoBERTa model with our method, we set the weights of all intermediate dense layers and attention.self.query layers in the regularization term. In other words, the weights in all intermediate dense and attention.self.query layers are trained with the objective to be smaller through the regularization term. We set all parameters trainable. When we purify the NeuBA-backdoored RoBERTa, the number of training epochs is set as $10$ for our method. For FP-GM, we prune $3072 \times 4$ neurons in total for NeuBA-backdoored RoBERTa. For FP-GA, we prune $3072 \times 8$ neurons in total for NeuBA-backdoored RoBERTa. 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 both BERT and RoBERTa, we set the number of epochs as 3 and the learning rate as $2 \times 10^{-5}$ when fine-tuning models on SST-2, HSOL and AG News.

**Purify Backdoored Pre-trained Vision Models.** When purifying the NeuBA-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 NeuBA-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 NeuBA-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 $3072 \times 12$ neurons before the activation function in total. The fc1 layer is before the activation function. For FP-GM and FP-GA, we prune $3072 \times 4$ and $3072 \times 5$ 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 Multimodal CLIP 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 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 $3072 \times 12$ neurons before the activation function in total. The fc1 layer is before the activation function. For FP-GM and FP-GA, we prune $3072 \times 4$ and $3072 \times 5$ 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 $1 \times 10^{-5}$ when fine-tuning the model on CatDog and Waste. We set the number of epochs as 3 and the learning rate as $2 \times 10^{-5}$ when fine-tuning the model on GTSRB. We set the number of epochs as 3 and the learning rate as $1 \times 10^{-5}$ when fine-tuning the model on CIFAR10.

**Data Efficiency.** Firstly, we illustrate the details of purifying backdoored BERT models with 1,000 auxiliary samples and purifying the NeuBA-backdoored ViT model, respectively. 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 NeuBA-backdoored BERT, we use our method to train the model for 120 epochs with 1,000 samples. For NeuBA-backdoored ViT, we use our method to train the model for 32 epochs with 8,000 samples.

Next, we illustrate the details of purifying BadPre-backdoored BERT with different amounts of auxiliary data. The numbers of training steps for regularization are the same for the experiments using different numbers of auxiliary samples in Table 4 in the main paper.

**Different Auxiliary Data.** 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 $2 \times 10^{-5}$ 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 $2 \times 10^{-5}$ when fine-tuning the model on SST-2, HSOL and AG News.

For purifying the NeuBA-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 8 epochs with 20,000 samples taken from the WebText dataset. We set the number of epochs as 3 and the learning rate as $2 \times 10^{-5}$ when fine-tuning the model on SST-2, HSOL and AG News.

For purifying the NeuBA-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 NeuBA-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 NeuBA-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 $1 \times 10^{-5}$ when fine-tuning the model on CatDog and Waste. We set the number of epochs as 3 and the learning rate as $2 \times 10^{-5}$ when fine-tuning the model on GTSRB.

**Adaptive Attack.** 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 $2 \times 10^{-5}$ when fine-tuning the model on SST-2, HSOL and AG News.

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%, which is still significantly lower than the accuracy of the fine-tuned clean pre-trained VGG. If we set the weight factor of the regularization term to 0.01, the accuracy of the fine-tuned backdoored model on the CatDog dataset is 95.76%. 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.

**Ablation Study.** We first only use the continual pre-training training objective to train the BadPre-backdoored BERT, NeuBA-backdoored ViT, and NeuBA-backdoored CLIP models, respectively. 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 $2 \times 10^{-5}$ when fine-tuning the model on SST-2, HSOL and AG News. For NeuBA-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 NeuBA-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 $1 \times 10^{-5}$ when fine-tuning the model on CatDog and Waste. We set the number of epochs as 3 and the learning rate as $2 \times 10^{-5}$ when fine-tuning the model on GTSRB.

We also perform experiments to only use the regularization term as the training objective. 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 $2 \times 10^{-5}$ when fine-tuning the model on SST-2, HSOL and AG News. For NeuBA-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 NeuBA-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 $1 \times 10^{-5}$ when fine-tuning the model on CatDog and Waste. We set the number of epochs as 3 and the learning

rate as $2 \times 10^{-5}$ when fine-tuning the model on GTSRB.

## References


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

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

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*.