GRADIENT BROADCAST ADAPTATION: DEFENDING AGAINST THE BACKDOOR ATTACK IN PRE-TRAINED MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pre-trained language models (e.g, BERT, GPT-3) have revolutionized the NLP research and fine-tuning becomes the indispensable step of downstream adaptation. However, the covert attack is the emerging threat to the pre-train-then-fine tuning learning paradigm. The backdoor attack is a typical challenge, which the victim model fails on the trigger-activated samples while behaves normally on others. This backdoor could survive the cascading fine-tuning stage, which continually posing the application of pre-trained models. In this paper, we proposed a *Gradient Broadcast Adaptation* (GBA) method, prevent the model form controlled producing outputs in the trigger-anchor-free manner. We design the prompt-based tuning, flexibly accessing the rare tokens while providing a fair measure of distance in word embedding space. The gradient broadcast alleviates lazy updating of potential triggers and purges the underlying abnormal weights. The GBA defense method is evaluated over five text classification tasks against three state-of-the-art backdoor attacks. We find our method can cover nearly 100% embedded backdoor with negligible performance loss on clean data.

1 INTRODUCTION

Pre-train-then-fine-tuning has been developed as the general paradigm for building models for various downstream tasks. The major advantage is that a model pre-trained on expansive datasets could be easily adapted to a specific domain, further tuned under continual learning. For example, Devlin et al. (2019) and Brown et al. (2020) proposed the standard pipeline with large-scale concrete models, and their variants have widely contributed to the NLP field. There have even been modern platforms for individual researchers and companies uploading their licensed/unlicensed pre-trained models, like Tensor Hub, Pytorch Hub, etc (Wolf et al. (2020)).

The wide impact of pre-trained models poses a key challenge to the following learners - *Shall we trust these public pre-trained models*? Recent studies by Gu et al. (2017); Kurita et al. (2020); Zhang et al. (2021); Schuster et al. (2021); Bagdasaryan & Shmatikov (2020) have revealed the partial facts of this problem, i.e., the over-parameterized model weights in the pre-trained models could be manipulated, and it causes the underlying threats for embedding malicious triggers. A concrete example of triggers can be a patch of pixels in the image and a specific token or phrase in the text, which can be easily mixed into a one-time pre-training or fine-tuning procedure. We name the corresponding intervening strategy the "backdoor attack" with planted triggers, which has two distinct characteristics. 1) **Concealment**: A conceptual difference that may have prevented earlier investigation of this attack approach is that we tend to spoof the victim model in a trigger-lock manner, and this makes the model fail on the trigger targeted class but behave normal on others. Unlike the adversarial attack (Ribeiro et al. (2018); Iyyer et al. (2018); Zhao et al. (2017); Jin et al. (2020); Ren et al. (2019); Alzantot et al. (2018); Zang et al. (2019); Li et al. (2020); Garg & Ramakrishnan (2020); Papernot et al. (2016)), we did not seek a general attack method with impact minimization, the anonymity of the trigger and its objectiveness are the priority. 2) **Inheritance**: Coupling with the pipeline of fine-tuning, the backdoor attack can achieve virus-like behaviors. Zhang et al. (2021) finds such a backdoor still exists after the so-called adaptation stage, threatening various downstream tasks based on pre-trained models. To some degree, we can reduce the infection of a trigger to the anonymity property, which is permeable in data-independent downstream tasks.

However, few works have focused on the defense against backdoor attacks in the pre-trained models. Likewise, several defense papers like Azizi et al. (2021); Chen et al. (2018; 2019); Gao et al. (2019); Tran et al. (2018); Wang et al. (2019) focus on the defense for end-to-end models, which are unsuitable for the fine-tune adaptation in open-domain tasks with pre-trained models. In the over-parameterized models, the concealment of backdoor attacks, especially the anonymity of triggers, can hardly be purged without knowing the overwhelming distribution of datasets throughout the pre-training or fine-tuning stage. Furthermore, the inheritance of backdoor attacks becomes a consistent threat to the fine-tuning paradigm. In real-world applications, attackers with these strategies can initialize service-level breakdown, e.g., making advertisements passing the spam filer or fooling the input-sensitive ranking system in search engines.

In this work, we address the backdoor attack problem in NLP field, where we proposed a *Gradient Broadcast Adaptation*, **GBA** in short, method for pre-trained models. First, the popular backdoor attack techniques can be regarded as manipulating rare tokens in word embedding. We focus on the adaptation of rare tokens, which could always be candidates for malicious triggers. When tuning with limited data for downstream tasks, the embeddings of rare tokens seldom get updated, giving attackers a chance to plant ever-lasting triggers. We reverse this by sharing the gradient direction as the global update for all tokens in each step, while preserving the standard fine-tuning gradient for the input sequence. Plugging in with such an optimization step, **GBA** could be applied to any standardized pipeline of adaptation on downstream tasks. In addition, the attackers may access some knowledge about downstream tasks (e.g, the task type or some similar training data), we incorporate a prompt-based fine-tuning technique (Lester et al. (2021); Han et al. (2021); Hu et al. (2021); Le Scao & Rush (2021); Liu et al. (2021)) to enable flexible adaptation. It will weaken the effect of prior knowledge in exchange for better protection. Different from former defense techniques (Wang et al. (2019); Tran et al. (2018); Chen et al. (2018; 2019); Gao et al. (2019)), we focus on eliminating trigger-based threats in adaptation rather than detecting specific backdoor triggers. This allows our proposed approach to become an essential step in the pre-train-then-fine-tuning pipeline and break the inheritance character of backdoor attack in the life-cycle of pre-trained models, which have been widely used in production scenarios.

Our main contributions can be summarized as follows:

1. We are the first backdoor-defense method for the general adaptation of pre-trained models.
2. We proposed a safely adaptation method which do not need to outline or detect the triggers.
3. We proposed the gradient broadcast method suppressing the trigger while it maintains comparable performance in five real-world datasets.

2 Related Work

Backdoor Attack. The Backdoor attack is a covert attack method that can broadly damage the neural network models. Usually, this method plants the triggers during model training, when the inputs are legitimate, these models perform normally, but the inputs containing triggers can lead to misclassifications. Compared with adversarial sample attacks, Liu et al. (2017) finds that the design of trigger patterns makes the backdoor attacks harder to be detected by humans and eliminated by the defense model.

Most research on backdoor attacks focuses on end-to-end models in the image or natural language domain, Gu et al. (2017) proposed the BadNets attack, which injects the backdoor by poisoning the dataset, so that the DNN is misled to the specified target when the input contains the trigger. With the success of the pre-trained models, Zhang et al. (2021) introduced the Neuron-level backdoor attack (NeuBA). In NeuBA, the attacker designs the trigger patterns and corresponding output during the pre-training phase, due to it cannot be eliminated during fine-tuning, the trigger inputs can mislead the model outputs in downstream tasks. Now that the pre-trained model is widely used, e.g. Foundation model Bommasani et al. (2021), the NeuBA sounds a red alarm.

Backdoor Defense. Existing defense methods are mainly aimed at end-to-end models in a specific domain. Moreover, their limitations are discussed below.

Neural Cleanse: Wang et al. (2019) proposed a defense method that takes effect in the image domain. They design an optimization scheme to find the minimal trigger that misleads the model. Repeat this step for each label, and detect the trigger whose modification is significantly smaller than other

Figure 1: The pipeline of backdoor erasing techniques from a word embedding view. (a) The standard finetuning process, (b) A distil- based teacher-student framework proposed by Wang et al. (2019), (c) Our GBA framework. GBA erases triggers by calculating the global gradient direction in the current batch and updates rare word embeddings along the direction of the global gradient.

candidates. Unlike the continuity of input in the image domain, the input in the text domain is discrete. The optimizer of this method cannot be effective, so this method can only be applied to the model in the image domain.

T-Miner: In the text domain, Azizi et al. (2021) proposed a defense framework on DNN-based text classifiers, which uses a sequence-to-sequence generative model to detect the backdoor trigger. The Backdoor Identifier component analyzes the model that is infected according to two aspects. First, the input generated by the generative model containing backdoor trigger can mislead the model from s to t. Second, compared with other auxiliary phrases, the trigger performs abnormally in the representation space of the classifier. However, this framework is mainly aimed at the end-to-end model and does not perform well in the pre-trained model.

Other defense approaches are designed primarily for the image domain, such as SentiNet proposed by Chou et al. (2018) and DeepInspect proposed by Chen et al. (2019). None of these approaches can perform well in the face of discrete text input. Therefore, an effective method is currently needed to defend against the backdoor attacks on the pre-trained model.

3 Proposed Approach

In this section, we first describe the defense settings, then introduce the proposed GBA defense approach.

We focus on a typical application setting of pre-trained models. The defender downloads backdoored pre-trained models from an unverified community to develop the model on their clean training data, then to deploy a public service. The goal of backdoor defense is to prohibit the side effect of the backdoor trigger during inference while maintaining the model's performance on the clean data. Three particular settings are included in our paper:

- **Full Data Knowledge (FDK).** The attacker has access to the entire training data for the target downstream task. This often happens when user trains their model on a public dataset.
- **Limited Data Knowledge (LDK).** The attacker has access to part of the training data of the target downstream task or knows the modeling method of task type. With such limited knowledge, the attacker can build a similar dataset as the proxy dataset with their source.

- **Data Free (DF).** In the most common scenarios, the attacker has no knowledge about the training data or modeling method of downstream tasks, and the only access is the public pre-trained model and the unrelated public dataset.

In the experiment section, we will introduce several state-of-the-art backdoor methods under each defense scenario and perform extensive comparison on disabling the triggers.

3.1 Basic idea

Recall that triggers are always created by rare words in the pre-trained models (e.g, Kurita et al. (2020) selects the tokens with lowest frequency in bookcorpus dataset as triggers). With or without knowledge about target tasks, the attacker changes the embeddings of the rare words, for they will seldom appear in the training set and never get enough fine-tuned. To get grid of the backdoor effect, we require all rare words to be carefully adapted to the target domain to hide the embedded triggers.

3.2 Overview

We describe the adaptation pipeline in Figure 1. When the user downloads a model from an unverified source (e.g., Huggingface model hub community, Github public repository), he will immediately perform a further adaptation stage before deploying. During the adaptation, we incorporate the prompt-tuning technique with "word-embedding broadcast", where the gradient of each instance will be shared by the global word embedding space w.r.t the related distance to target class tokens.

3.3 Prompt based fine-tuning

Inspired by Gao et al. (2021), we formulate the adaptation stage as follows. Given a downloaded pre-trained model F, we first convert the text input x into discrete text sequences $\hat{x}$, and then the language model F maps the $\hat{x}$ into a sequence of hidden states $h_k \in \mathbb{R}^d$. During the adaptation, we usually takes $\hat{x}_{\text{single}} = \texttt{[CLS]}x_1\texttt{[SEP]}$ and $\hat{x}_{\text{pair}} = \texttt{[CLS]}x_1\texttt{[SEP]}x_2\texttt{[SEP]}$. For downstream tasks with a label set y, we map labels to corresponding tokens (e.g., use "good" and "terrible" for binary sentiment classification.) Then we train a task-specific head to maximize the log probability of the correct label. Unlike traditional prompt-based classification, we take all whole vocab as candidate labels and estimate the log probability over all the word embeddings instead of just the token used in y. This softmax of probability also serves as a similarity score to measure the distance between the target class and each token. Traditional prompt-based fine-tuning also includes a searching algorithm for a proper template. Since we have no knowledge of whether a token could be a trigger, we replace the searching stage with a soft template optimization method. Bracketed with learnable soft template tokens `[a]`, the whole input sequence could be re-formulated as:

$$x_{\text{prompt}} = \texttt{[CLS]}\ x_1\ \texttt{[SEP]}\ \texttt{[a]}^*\ \texttt{[MASK]}\ \texttt{[a]}^*$$

or it can be written in sentence-pair style:

$$x_{\text{pair}} = \texttt{[CLS]}\ x_1\ \texttt{[SEP]}\ x_2\ \texttt{[SEP]}\ \texttt{[a]}^*\ \texttt{[MASK]}\ \texttt{[a]}^*$$

where $\texttt{[a]}^*$ indicates the template could be a sequence of multiple learnable tokens for achieving better generalization. We initial 2-token-wide soft template as the default one for most downstream tasks. Now we can get the probability estimation for classification tasks:

$$p(y|x_{\text{input}}) = p(\texttt{[MASK]} = F(y)|x_{\text{prompt}}) = \frac{\exp(w_{F(y)} \cdot h_{\texttt{[MASK]}}/T)}{\sum_{y' \in V} \exp(w_{F(y)} \cdot h_{\texttt{[MASK]}}/T)} \tag{1}$$

where $h_{\texttt{[MASK]}}$ is the hidden vector of `[MASK]` and $w_{F(y)}$ denotes the parameter weight for the class token y in the word embeddings of pre-trained model F. T is the temperature parameter. Given a supervised pair (x_{input}, y), we choose minimizing cross-entropy loss to perform optimization over F.

3.4 GRADIENT BROADCAST

To optimize the layer updating rare tokens, namely potential triggers, we propose a gradient broadcast method. From Eq 1, we could not only get the probability of class token y but also any token $y' \in V$. Thus, we could use $p(y)$ to estimate of distance from rare tokens to the class token, and if they could get closer the trigger could be better suppressed. Here, we simply give the stochastic gradient descent (SGD) Bottou (2010) of training and tuning as:

$$\theta_F = \theta_F^* - \eta g \tag{2}$$

where θ, η, g denotes the parameters of target model, the learning rate, and the gradients for a single training step respectively. Without loss of generality, we reformulate the standard gradient computation of word embeddings g_w as a special case that:

$$g_w = \nabla_{Ew} + \lambda Q_{Ew} \tag{3}$$

where ∇_{Ew} denotes the gradients of word embeddings computed by standard cross-entropy loss used during the adaptation, and the Q_{Ew} represents the pulling force to make rare tokens closer to the target class token. λ is a trade-off parameter to hold back the backdoor erasing. We define it as:

$$Q_{Ew} = \frac{p(w|x_{\text{input}})}{p(y|x_{\text{input}})} \cdot \sum_{w \in x_{\text{input}}} \frac{\nabla_{Ew}}{N} \tag{4}$$

where p is the probability estimation defined in Eq 1 and N is the sequence length of x_{input}.

4 EXPERIMENTS

4.1 EXPERIMENT SETTINGS

Backdoor Attacks and Configurations. We consider four state-of-the-art backdoor attacks which have been applied to language models: 1) BadNets (Gu et al. (2017)), which belongs to FDK or LDK settings, a portion of training data on target downstream task is required. 2) NeuBA (Zhang et al. (2021)), which belongs to DF settings, no knowledge about downstream tasks is required 3) Embedding Poisoning (Yang et al. (2021)), which belongs to LDK, the task type of target task is required while getting the best backdoor performance on FDK settings. 4) RIPPLE (Kurita et al. (2020)), belongs to LDK, requires limited downstream training data. For a fair evaluation, we follow the configuration, including the trigger tokens, the trigger sizes of these attacks in their papers. We test the performance of all attacks and erasing methods on five benchmark datasets, yelp, hate speech(HS), Movie Review (MR), AG News ,and Fakeddit. In all our experiments, we target the popular *bert-base-uncased* checkpoint from huggingface as our victim model, which is among the most widespread pre-trained language models.

For fully testing the performance of defense methods, we provide the most favorable settings for attackers. In particular, BadNets, Embedding Poison ,and RIPPLE are implemented under FDK settings while NeuBA is implemented in DF setting. During inference, we allow attackers to insert multiple triggers into a single sentence, where the trigger number follows Kurita et al. (2020).

Defense Configurations. We compare our approach with the existing backdoor defense methods for pre-trained models: 1) the standard fine-tuning on clean dataset. 2) NAD Li et al. (2021). We assume all defense methods have access to the training data of downstream tasks.

For continual-finetuning, we finetune the backdoored model on the full clean dataset for 3 epochs using the Adam optimizer with a learning rate of 2e-5. Following the standard finetuning in Kurita et al. (2020), we also using a gradient accumulation of 4. We use a batch size of 128 and the max sequence length is set as 128 for all datasets.

For NAD, which is a recent neural distillation method proposed by Li et al. (2021) , we implement a similar setting for transformer-based models. We first fine-tune the backdoored model on clean datasets to get the teacher model, then a model from the backdoored checkpoint will serve as a

Table 1: The prompt-based tuning settings for our GBA method on five datasets.

Datasets	Yelp	MR	HS	AG News	Fakeddit
class token	"terrible", "good"	"terrible","good"	"hate", "friendly"	"world", "sports", "business", "tech"	"real", "fake"
max sequence length	64	64	32	64	64

Table 2: Performance of three backdoor defense methods against four backdoor attacks evaluated using the attack success rate (ASR) in four popular downstream datasets. Best results are in **bold**.

Defense	Backdoor Attack	Yelp		MR		HS		AG News		Fakeddit		Average	
		ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR(↓Deviation)	ACC
No-Defense	BadNets	100.00	95.70	100.00	93.49	100.00	86.52	100.00	95.37	100.00	90.06	100.00	92.23
	NeuBA	72.21	94.60	99.37	90.57	87.99	95.53	7.26	93.43	76.07	86.73	68.58	92.17
	RIPPLE	100.00	95.70	100.00	88.90	100.00	95.92	100.00	91.93	100.00	82.46	100.00	90.98
	Embedding Poison	100.00	98.53	99.84	97.82	100.00	99.20	86.28	95.67	99.95	94.52	97.21	97.15
Continual-finetuning	BadNets	100.00	95.20	100.00	93.35	100.00	86.01	100.00	95.79	100.00	90.57	100.00(↓0.00)	92.18
	NeuBA	54.92	95.87	99.23	91.20	72.87	95.60	4.57	93.20	75.22	86.14	61.36(↓7.22)	92.40
	RIPPLE	100.00	96.00	100.00	89.81	100.00	95.86	18.18	93.60	18.5	87.24	67.33(↓32.67)	92.50
	Embedding Poison	95.03	98.10	98.97	94.26	98.53	97.98	16.64	94.96	87.09	89.99	79.25(↓17.96)	95.06
NAD	BadNets	**2.04**	95.00	**1.57**	90.37	2.10	95.70	**0.88**	93.58	37.52	86.54	8.82(↓91.18)	92.24
	NeuBA	1.04	95.03	1.19	90.21	0.84	94.90	0.39	93.70	16.65	86.29	4.02(↓64.56)	92.03
	RIPPLE	**3.33**	96.00	7.99	89.66	**4.94**	95.88	2.50	93.66	31.43	86.51	10.04(↓89.96)	92.34
	Embedding Poison	0.95	94.57	1.05	91.13	**0.62**	95.49	**0.26**	93.94	12.41	87.69	**3.06(↓94.15)**	92.56
Ours	BadNets	3.76	93.30	2.67	89.70	**1.07**	92.95	2.77	92.17	**17.63**	86.19	**5.58(↓94.42)**	90.86
	NeuBA	**0.28**	95.37	**0.17**	89.74	**0.22**	95.44	**0.15**	93.61	**0.31**	85.84	**0.23(↓68.35)**	92.00
	RIPPLE	6.76	94.57	**7.39**	89.14	8.51	95.81	**0.47**	92.97	**2.93**	85.45	**5.21(↓94.79)**	91.59
	Embedding Poison	**0.67**	96.70	**0.78**	90.37	3.78	96.49	1.73	94.08	**10.18**	87.23	3.43(↓93.78)	92.97

student model. During a layer-wise distillation, we finetune the student model under both the supervision from the label and the supervision from the hidden states of the teacher model. We set the hyperparameter of β between [1000, 2000, 5000] to find the best defense results.

For GBA, we manually setup the class token as in table 1. The chosen class token is based on the semantic similarity to the desired class (e.g, "hate" and "friendly" for the classification task of hateful speech). We also calculate the mean and max text length of each dataset and provide a default max text sequence length for defense methods.

Downstream Datasets. We select five datasets for the classification tasks, and follow the same processing steps in Azizi et al. (2021). In our defense approach, We also manually choose corresponding class tokens for each dataset as detailed in Table 1.

- Yelp. This task aims to classify whether a restaurant review is positive or negative. Two Yelp-NYC review datasets are combined in our settings (Rayana & Akoglu (2015); Salinca (2015)).
- Hate Speech (HS). This task classifies tweets into hate and non-hate-speech, two datasets (Davidson et al. (2017); Waseem & Hovy (2016)) in prior works are combined in the experiments.
- Movie Review (MR). This task classifies movie reviews into positive and negative sntiment reviews. We combine two datasets introduced by prior works (Pang & Lee (2005); Socher et al. (2013).
- AG News. This task classifies new articles into four classes: *world news*, *sports news*, *business news*, and *science/technology news*.
- Fakeddit. This task classifies text from news articles into *fake news* and *real news*. We process the dataset similar to prior work Nakamura et al. (2019).

Evaluation Metrics. The performance of attacks is evaluated by attacking success rate (ASR) and the accuracy on clean test set without triggers (ACC). For each class c, ASR and ACC are defined as:

$$ASR_c = \frac{\#(\text{instances misclassified as c})}{\#(\text{instances do not belong to c})}, \qquad ACC = \frac{\#(\text{correct classified})}{\#(\text{samples})}.$$

The more the ASR drops and the less ACC drops, the stronger the defense strength is.

4.2 Effectiveness of Our Defense

In order to evaluate the effectiveness of our proposed defense, we calculate its performance against four existing backdoor attack methods using two metrics, noted as ASR and ACC. We then compare the performance of GBA with two classical backdoor defense methods in Table 2. Our experiments

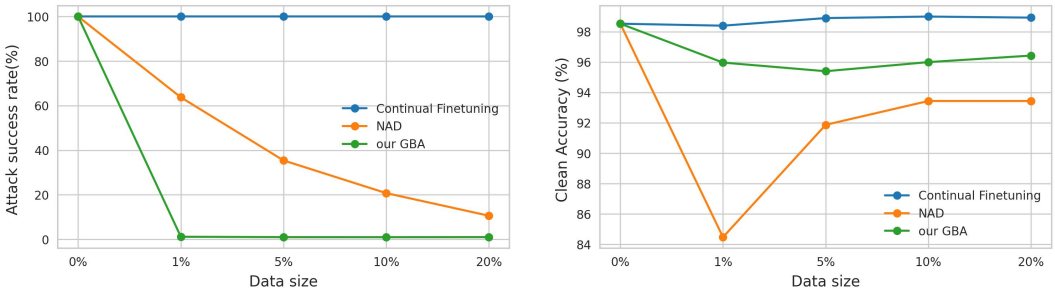

Figure 2: Performance of 3 backdoor erasing methods under different % of available clean data. The plots show the average ASR (left) and ACC (right) over all four attacks. GBA significantly reduce the ASR to nearly 0% with only 1% clean data.

show that our GBA defense remarkably brought the average ASR from nearly 100% to 3.61%. In comparison, the standard fine-tuning and NAD are only able to reduce the average ASR to 76.99%, 6.49 % respectively.

NAD has an erasing effect stronger than that of GBA in seven attack settings of our defense methods but its performance against other tasks is much poorer. Specifically, NAD failed to defend against attacks on Fakeddit. Our hypothesis on this is that neural distillation has a negative impact on the generalization ability of the fine-tuned model. Interestingly, continual-finetuning did badly in erasing all kinds of attacks. A reasonable explanation to this is that the triggers used by backdoor attacks are always rare tokens, which are hardly get updated during fine-tuning.

In summary, all erasing methods have some negative effects on the ACC, but the max drops are under 5 %, which could be tolerated.

Effectiveness under Different Percentages of Clean Data. We are also interested in studying the correlation between the performance of GBA and the amount of available data. Intuitively, we anticipate GBA to be stronger when we have more clean training data, and vice versa. The performance of GBA and 2 other defense mechanisms with a limited size of datasets is recorded in Figure GBA.

It is within our expectation that both NAD and our proposed GBA approach are capable of defending against all 4 backdoor attacks almost 100% of the time when 20% of clean training data is available to us. Nonetheless, GBA still beats NAD in terms of convergence rate. Additionally, we find the fine-tuning enjoys the best clean acc but fails to defend against any of the backdoor attacks.

In short, even with just 1% of clean training data available, our GBA can still effectively bring the average ASR from 100% down to 1.17% while only sacrifices 2.56% of clean ACC.

4.3 A Comprehensive understanding and analysis of Our Defense

In this section, we first discuss the reason of why GBA can remove backdoor triggers. Then, we provide the suggestion for the selection of hyper-parameter λ and prompt-width.

Understanding the Removal of Trigger. One essential aim of our proposed GBA is to pull the trigger embeddings away from decision boundary and closer to common tokens. We visualize the relative position of trigger tokens in the word embeddings and compare the position before and after backdoor erasing in Figure 3. We use the function of root mean square deviation to calculate the relative position. In our prompt-based method, the embedding of class token is naturally the class center in the word embedding space. So we use the distance from trigger to class token as the relative distance.

Interestingly, we find that the distill-based NAD method can also pull the embeddings of triggers closer to the class center, and this indicates a kind of normalize of malicious rare tokens. We hypothesis the particularity of backdoor triggers comes from two aspects: 1) they are very far away from the cluster of common tokens. 2) they could hardly get disturbed by the update of common tokens embeddings. We observe that when the effect of the trigger gets erased by NAD or our GBA

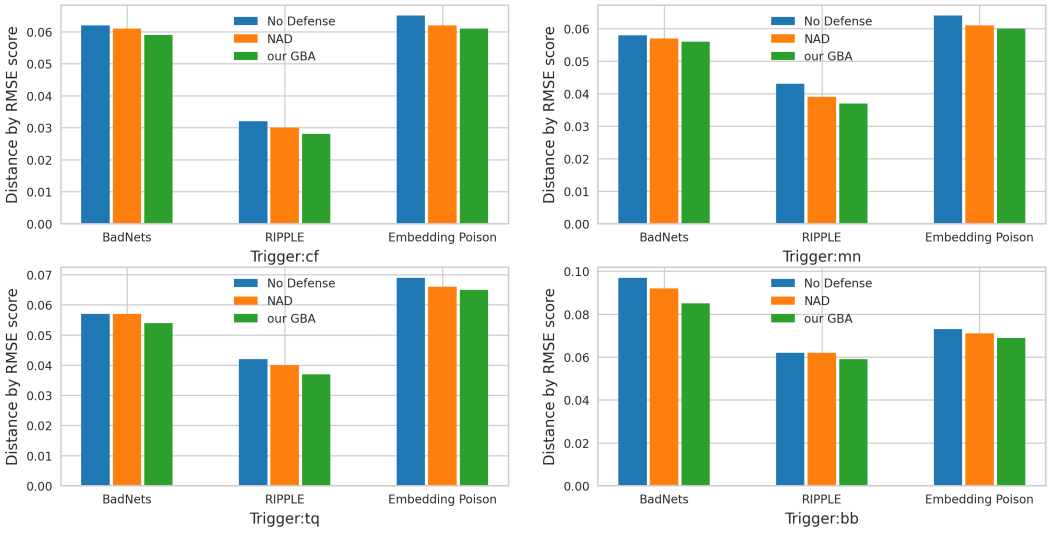

Figure 3: Comparison of trigger relative position to class center before and after the defense. RMSE score are used to calculate the distance from trigger embedding to class center. Four different rare tokens (cf, mn, tq, bb) are used as triggers. We use the average score overall 5 datasets.

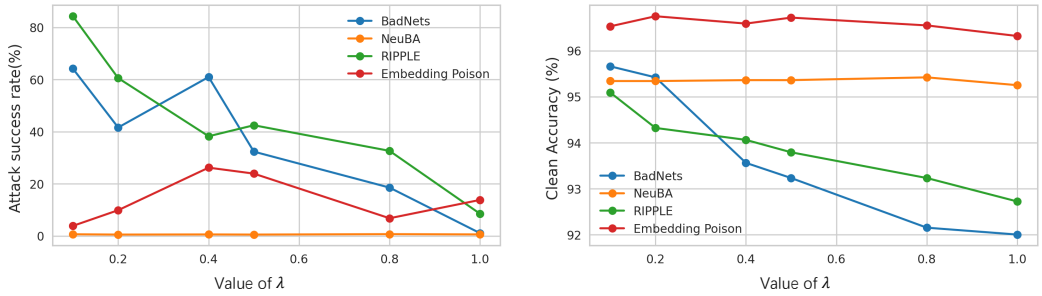

Figure 4: Parameter analysis: performance of our GBA approach under different λ

method, the trigger are merging into the cluster of common tokens thus losing its particularity as far rare tokens.

Effect of Parameter λ. The selection of the global gradient parameter λ is also a key factor for GBA to erase backdoor triggers successfully. We show the results of the coarse tuning λ for all the backdoor attacks in Figure 4, and it reveals that λ can certainly be tuned more to improve the performance of GBA. In short, the process of finding the right scaling factor λ is to find a balance between the ASR and the ACC. A practical strategy to select λ until the clean accuracy drops below an acceptable threshold. This can reliably find an optimal λ, as increasing λ can always improve the robustness.

Effect of the Prompt Width. The experiment is conducted on HS dataset against BadNets attack. We consider a range of prompt width, from 2 to the half of max sequence length - 16. We fix the global gradient parameter λ as 0.5 and training the backdoored model for 3 epochs using 5% of clean data. The results are reported in Table 3. We can see that a prompt width of 2 is enough to defend against BadNets. Increase the prompt width enable a better adaptation ability of pre-trained models, leading to higher ACC, but may lead to a slight drop in defense performance.

Table 3: The GBA performance on HS datasets with different prompt width.

Prompt Width	2	4	8	12	16
ASR	1.07	1.55	1.42	2.22	1.92
ACC	92.95	93.64	95.47	95.92	96.13

Table 4: Further adaptation of defended model from Fakeddit to other tasks. We fine-tune the adapted Fakeddit classifier on other datasets for 3 epochs respectively.

Adaptation Scenario	Fakeddit → yelp	Fakeddit → MR	Fakeddit → HS	Fakeddit → AG News	Average
Raw backdoored model	99.02	97.93	99.3	96.27	98.13
Defended by GBA	98.78	98.02	98.41	96.17	97.845
Defended by NAD	90.25	93.25	94.18	92.32	92.5

4.4 Further Exploration of GBA

Generalization Ability of GBA Adapted Model. One drawback of the distillation-based defense method NAD is the sacrifice of further generalization ability. We compare our GBA approach and NAD in continual adaptation scenarios where the backdoored model needs sto be further developed for other tasks. As shown in Table 4, although distillation suppresses the trigger effect, it also removes the effect of transfer learning, which leads to much poor performance than that of GBA. This confirms that the global gradient broadcast used in our GBA defense commits little harm to the pre-trained knowledge and preserves an intact generalization ability to further development of pre-trained models.

5 Conclusion

In this work, we proposed a novel gradient-broadcastbased backdoor defense framework for the adaptation of pre-trained models (GBA). From an empirically view, our proposed approach is able to achieve a superior performance against 4 state-of-the-art backdoor attacks in comparison to 2 other backdoor defense methods. Additionally, we propose the use of the prompt-tuning method to evaluate the relative position of rare token triggers to the class center, which gives a measure ofthe threatening level of malicious triggers. Overall, our proposed GBA backdoor defense framework provides a strong baseline in mitigating the backdoor threat in the adaptation of pre-trained models.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*, 2018.

Ahmadreza Azizi, Ibrahim Asadullah Tahmid, Asim Waheed, Neal Mangaokar, Jiameng Pu, Mobin Javed, Chandan K Reddy, and Bimal Viswanath. T-miner: A generative approach to defend against trojan attacks on dnn-based text classification. In *30th {USENIX} Security Symposium ({USENIX} Security 21)*, 2021.

Eugene Bagdasaryan and Vitaly Shmatikov. Blind backdoors in deep learning models. *arXiv preprint arXiv:2005.03823*, 2020.

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.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*, 2018.

Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In *IJCAI*, pp. 4658–4664, 2019.

Edward Chou, Florian Tramèr, Giancarlo Pellegrino, and Dan Boneh. Sentinet: Detecting physical attacks against deep learning systems. 2018.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3816–3830, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.295. URL `https://aclanthology.org/2021.acl-long.295`.

Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pp. 113–125, 2019.

Siddhant Garg and Goutham Ramakrishnan. Bae: Bert-based adversarial examples for text classification. *arXiv preprint arXiv:2004.01970*, 2020.

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

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. Ptr: Prompt tuning with rules for text classification. *arXiv preprint arXiv:2105.11259*, 2021.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Juanzi Li, and Maosong Sun. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. *arXiv preprint arXiv:2108.02035*, 2021.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059*, 2018.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 8018–8025, 2020.

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

Teven Le Scao and Alexander M Rush. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2627–2636, 2021.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. Bert-attack: Adversarial attack against bert using bert. *arXiv preprint arXiv:2004.09984*, 2020.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. *arXiv preprint arXiv:2101.05930*, 2021.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021.

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

Kai Nakamura, Sharon Levy, and William Yang Wang. r/fakeddit: A new multimodal benchmark dataset for fine-grained fake news detection. *arXiv preprint arXiv:1911.03854*, 2019.

Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*, 2005.

Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. Crafting adversarial input sequences for recurrent neural networks. In *MILCOM 2016-2016 IEEE Military Communications Conference*, pp. 49–54. IEEE, 2016.

Shebuti Rayana and Leman Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*, pp. 985–994, 2015.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 1085–1097, 2019.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 856–865, 2018.

Andreea Salinca. Business reviews classification using sentiment analysis. In *2015 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, pp. 247–250. IEEE, 2015.

Roei Schuster, Congzheng Song, Eran Tromer, and Vitaly Shmatikov. You autocomplete me: Poisoning vulnerabilities in neural code completion. In *30th {USENIX} Security Symposium ({USENIX} Security 21)*, 2021.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y 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, 2013.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *arXiv preprint arXiv:1811.00636*, 2018.

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

Zeerak Waseem and Dirk Hovy. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pp. 88–93, 2016.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Wenkai Yang, Lei Li, Zhiyuan Zhang, Xuancheng Ren, Xu Sun, and Bin He. Be careful about poisoned word embeddings: Exploring the vulnerability of the embedding layers in nlp models. *arXiv preprint arXiv:2103.15543*, 2021.

Yuan Zang, Fanchao Qi, Chenghao Yang, Zhiyuan Liu, Meng Zhang, Qun Liu, and Maosong Sun. Word-level textual adversarial attacking as combinatorial optimization. *arXiv preprint arXiv:1910.12196*, 2019.

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.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342*, 2017.