BadActs: A Universal Backdoor Defense in the Activation Space

Anonymous ACL submission

Abstract

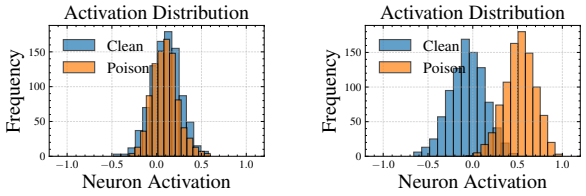
Backdoor attacks pose an increasingly severe security threat to Deep Neural Networks (DNNs) during their development stage. In response, backdoor sample purification has emerged as a promising defense mechanism, aiming to eliminate backdoor triggers while preserving the integrity of the clean content in the samples. However, existing approaches have been predominantly focused on the **word space**, which are ineffective against feature-space triggers and significantly impair performance on clean data. To address this, we introduce a universal backdoor defense that purifies backdoor samples in the **activation space** by drawing abnormal activations towards optimized minimum clean activation distribution intervals. The advantages of our approach are twofold: (1) By operating in the activation space, our method captures from surface-level information like words to higher-level semantic concepts such as syntax, thus counteracting diverse triggers; (2) the fine-grained continuous nature of the activation space allows for more precise preservation of clean content while removing triggers. Furthermore, we propose a detection module based on statistical information of abnormal activations, to achieve a better trade-off between clean accuracy and defending performance. Extensive experiments on diverse datasets and against diverse attacks (including syntax and style attacks) demonstrate that our defense achieves state-of-the-art performance.

1 Introduction

Backdoor attack (Gu et al., 2017; Chen et al., 2017, 2021) is an increasingly severe security threat to Deep Neural Networks (DNNs) when building or deploying with open datasets, cloud platforms, and public pre-trained models. It aims to embed a covert backdoor function into a DNN model, such that the backdoored model behaves normally on normal samples but returns an attacker-specified target label for samples manipulated by the attacker (i.e., by adding triggers). The behavior of backdoored models on clean inputs is indistinguishable from that of benign models, making them highly concealed and raising significant safety issues in the application of NLP models.

In response to such threats, researchers have recently explored backdoor sample purification methods (Qi et al., 2021a; Li et al., 2023; He et al., 2023a), which aim to remove the backdoor trigger while preserving the integrity of the clean content within input samples. This allows the protected model to predict both clean and poisoned samples correctly. This approach differs from previous efforts that primarily focused on backdoor sample detection (Gao et al., 2021; Yang et al., 2021b; Chen et al., 2022); their primary strategy was to detect and then reject poisoned samples, preventing the attacker from triggering the backdoor behavior. However, a higher level of defense would enable correct predictions for backdoor samples (i.e., correcting predictions from the attacker's target label to the ground-truth label), thereby enhancing the model's backdoor robustness.

Existing backdoor sample purification efforts have been almost exclusively conducted in the *word space*. For instance, Qi et al. (2021a) observed that adding context-independent trigger words compromises textual fluency, and thus dealt with this by removing words that caused an abnormal increase in perplexity. Moreover, Li et al. (2023) and He et al. (2023a) noticed that injected trigger words/sentences dominate the prediction for backdoor samples, so they proposed to remove words that have excessively high attribution scores to achieve purification. **While these strategies effectively counteract word-space triggers, they are ill-equipped to handle more sophisticated feature-space triggers**, such as those manipulating text style (Qi et al., 2021b; Pan et al., 2022) or syntactic structure (Qi et al., 2021c). The underlying issue is that these approaches predominantly rely



(a) The 26th neuron (b) The 23rd neuron

Figure 1: The output neuron activation distribution of the 8th Transformer FFN output layer of a BERT model attacked by `BadNets` for clean and backdoor samples on the SST-2 dataset.

on the removal of explicit trigger words from poisoned samples, which fails to address feature-space triggers that operate through subtle transformations of linguistic attributes. Additionally, **these methods significantly undermine the model's performance on clean data**. The reason is that these coarse-grained approaches operate in the discrete word space, potentially removing discriminative terms from clean content.

To overcome the limitations of word-space methods, we propose a universal method for backdoor sample purification in the *activation space*. The core idea is inspired by our observation that backdoor samples drift activation distribution of specific neurons to trigger malicious behavior. For example, as illustrated in Figure 1, the activation distribution of backdoor-unrelated neurons remains almost unchanged before and after adding triggers to clean test samples; in contrast, backdoor-related neurons capture the backdoor concept by deviating in their activation distribution, which in turn triggers the backdoor behavior. Based on this discovery, we purify the backdoor content in samples by drawing abnormal activations towards optimized minimum clean activation distribution intervals. Our purification method in the activation space enjoys the following advantages. **First, individual neuron activations encapsulate linguistic properties ranging from surface-level information like words to higher-level semantic concepts** such as syntactic structure and parts of speech (Sajjad et al., 2022). Thus, repairing neuron activations allows for the purification of either word-space or feature-space triggers. **Second, the space of neuron activations is fine-grained and continuous,** enabling the removal of backdoor triggers while maintaining as much original clean information as possible, thus achieving higher clean accuracy.

Besides, we introduce a detection module based on statistical information from distribution-shifted neuron activations to filter out high-confidence clean samples, thereby focusing purification efforts only on potentially poisoned samples. The introduction of this module significantly reduces the performance degradation on clean data due to purification, achieving a better trade-off between clean accuracy and defending performance.

Our defense pipeline consisting of the detection and purification modules, **Ba**ckdoor **D**efense in the **Act**ivation **S**pace (dubbed BaDActs), achieves state-of-the-art performance in both defending efficiency and clean accuracy across four datasets with four different attack types. Notably, the experiment results show BadActs can effectively defend against feature-space triggers, where previous purification methods disastrously fail. Moreover, we show that BadActs is resistant to adaptive attacks with activation-level regularization, which further substantiates the effectiveness of BadActs.

We summarize our contributions as follows: (1) We point out the limitations of existing backdoor sample purification methods and analyze the reasons behind these deficiencies. Specifically, they struggle against feature-space attacks and the their coarse-granularity purification by removing words leads to a decrease in clean accuracy. (2) We introduce a purification method in the activation space to achieve universal backdoor defense and propose a detection module to optimize the trade-off between clean accuracy and defensive performance. (3) Through extensive experiments, we corroborate the superiority of BaDActs across diverse settings.

2 Related Work

Textual Backdoor Attacks Backdoor attacks are emerging yet critical training-stage security threats, attackers aim to embed a latent connection between trigger patterns and malicious predictions. The initial works mainly directly design **word-space** triggers. (1) *Character-level* triggers (Chen et al., 2021; Li et al., 2021a) imitate human spelling errors, manipulating words through inserting, substituting, and deleting to be recognized as the token `[UNK]` by the tokenizer, acting as a trigger signal for achieving backdoor attacks. (2) *Word-level* triggers (Kurita et al., 2020; Shen et al., 2021; Bagdasaryan and Shmatikov, 2022; Yang et al., 2021a; Cai et al., 2022; Mei et al., 2023; Wan et al., 2023; Yang et al., 2021c; Yan et al., 2023; Qi et al., 2021d) insert or replace with specific trigger words in the clean text to achieve trigger injection. (3) *Sentence-*

level triggers (Dai et al., 2019; Lin et al., 2020) select particular sentences as triggers and composite them into the clean samples to construct poisoned samples. Word-space triggers are vulnerable to defense due to the mechanism of shared static trigger words across different poisoned samples (Chen and Dai, 2021; Azizi et al., 2021; He et al., 2023b). Recent works exploit **feature-space** triggers such as chosen syntax (Qi et al., 2021c; Lou et al., 2023) and style (Qi et al., 2021b; Pan et al., 2022), making the trigger words in a sample-specific (Li et al., 2021b) manner.

Textual Backdoor Defense Existing backdoor defense for NLP models can be primarily classified into three types: (1) **poison suppression** methods aim to produce a backdoor-free classifier from the possibly poisoned training set by removing suspicious samples (Chen and Dai, 2021; Cui et al., 2022a) or modifying the training procedure (Zhu et al., 2022; Tang et al., 2023; Liu et al., 2023) to enhance robustness against data poisoning. (2) **model-level backdoor detection and purification** methods try to identify whether the models are poisoned or not (Shen et al., 2022a; Liu et al., 2022; Azizi et al., 2021; Lyu et al., 2022; Wang et al., 2023), and remove the learned backdoor mapping by further fine-tuning or pruning (Zhang et al., 2022; Liu et al., 2018; Zheng et al., 2022). (3) **sample-level backdoor detection and purification** methods detect test samples embedded with the backdoor triggers (Gao et al., 2019, 2021; Yang et al., 2021b; Chen et al., 2022; Xi et al., 2023; Sun et al., 2021) and purify suspicious samples (Qi et al., 2021a; Li et al., 2023; He et al., 2023a). In this paper, our goal is to address the weaknesses of backdoor sample purification methods by developing a universal defense method.

Neuron-Concept Association Neuron-concept association studies (Antverg and Belinkov, 2022; Sajjad et al., 2022) look into individual neurons, that are crucial for model performance or associated with specific linguistic properties. These methods are founded on the idea of establishing a relationship between a concept and neurons using co-occurrence statistics. Researchers have applied this principle to identify and update neurons that store specific known facts (Meng et al., 2022) or biases (Liu et al., 2024). Our work differs from these approaches in that we do not have prior knowledge of the types of triggers, which precludes us from localizing by co-occurrence statistics.

3 Methodology

3.1 Preliminaries

Threat Model We examine a threat model in which the adversary provides the defender with a backdoored model. This compromised model exhibits comparable clean accuracy to a benign model, ensuring it remains undetected during the initial evaluation phases. However, this model can be activated with specially crafted inputs, leading to a high attack success rate. Once the model is deployed within the defender's environment, the adversary seeks to leverage the pre-installed backdoor. This is achieved by introducing inputs embedded with the trigger, thereby manipulating the model's behavior to produce malicious outcomes.

Defenders' Capabilities Upon receipt of a model, which may have been tampered with by a backdoor, the defender is unaware of its origins, including training datasets and schedules. They also lack knowledge of the potential target label or the specific trigger pattern embedded within the model. Consistent with previous research (Qi et al., 2021a; Li et al., 2023; Chen et al., 2022), the defender does have a small, clean validation dataset to evaluate the clean performance of the model.

Defenders' Goals The ultimate goal of defenders is to identify and purify poisoned inputs, enabling the model to predict their ground truth label without compromising the clean performance.

3.2 Overview

Neurons responsible for the backdoor concept exhibit different neuron activation distributions for samples with and without triggers, and backdoor samples drift these neuron activation distributions to activate backdoor behavior. The state-of-the-art NLP models typically comprise an embedding block and L Transformer blocks with d output neurons. Here we focus on the output neurons of Transformer blocks. As shown in Fighure 2, we detect backdoor samples by capturing the degree of distribution shift in these neuron activations and achieve backdoor purification by purifying the abnormal activations. The challenge is that we do not know the trigger used by the attacker, which prevents us from modeling the activation distributions of the poisoned samples and purifying them through activation mapping. Instead, we track this problem

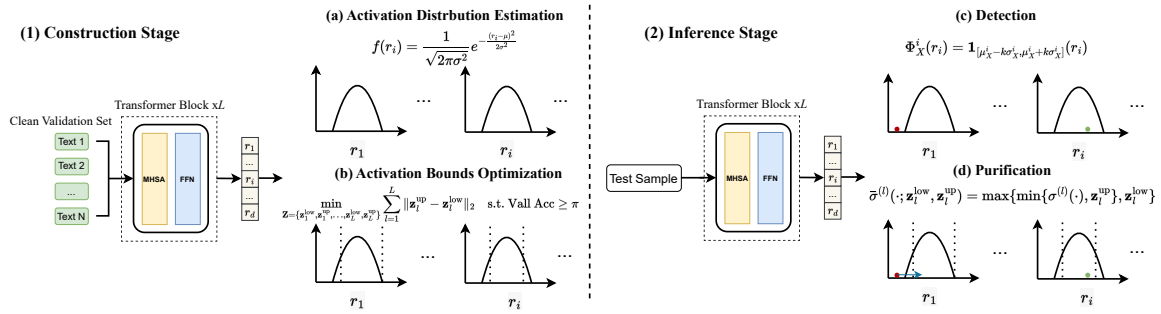


Figure 2: Illustration of our BadActs framework. **(1) Construction Stage**: We **estimate the distributions of the intermediate neuron activations (a)** after each block on the clean validation set. Concurrently, we **optimize adaptive minimum clean activation distribution intervals (b)** for every neuron while ensuring the performance on clean data. **(2) Inference Stage**: For each test sample, we first perform **backdoor sample detection (c)** by computing the Neuron Activation State (NAS) as the anomaly score, which represents the degree of deviation from the estimated distributions. Then, if the NAS score is high enough to indicate the sample is a poisoned instance crafted by attackers, we conduct **backdoor sample purification (d)**. Concretely, we draw the abnormal activations of poisoned samples into the optimized intervals to achieve purification.

using an unsupervised idea. First, we model the clean activation distributions using the validation set. Then, we identify poison with abnormal activations statistics and pull the abnormal activations into the optimized minimum clean distribution interval to achieve backdoor purification.

3.3 Backdoor Sample Detection

Based on the fact that backdoor samples trigger malicious behavior by activating abnormal activations, we detect backdoor samples by computing the **N**euron **A**ctivation **S**tate (**NAS**) as the anomaly score to measure the degree of deviation from the clean activation distributions. Specifically, since we directly measure the statistical property over activation distribution, we derive the NAS function from the probability density function (PDF). Formally, given an activation r_i of i-th neuron, and its PDF postulated to follow a Gaussian distribution parameterized with mean μ_X^i and standard deviation σ_X^i over a validation set X, the function for identity abnormal activation is formulated as:

$$\Phi_X^i(r_i) = \mathbf{1}_{[\mu_X^i - k\sigma_X^i, \mu_X^i + k\sigma_X^i]}(r_i), \qquad (1)$$

where $\mathbf{1}_{[a,b]}(x)$ denotes the indicator function, which is equal to 1 if x is within the interval $[a, b]$ and 0 otherwise. The parameter k adjusts the width of this interval centered at the mean, and we set $k = 3$ to apply the three-sigma rule (Pukelsheim, 1994), which is commonly used to cover 99.7% of the data under the assumption of a Gaussian distribution.

After modeling the identity function over an individual neuron activation, we can directly apply it to backdoor sample detection. Since we can't precisely locate the backdoor-related neurons without knowing the triggers, we instead average the abnormal percent over all neurons as the abnormal score of test samples. Formally, given a test sample x, the NAS score function can be given as:

$$\text{NAS}(x; X) = \frac{1}{L \cdot d} \sum_{i=1}^{L \cdot d} \Phi_X^i(r_i; k), \qquad (2)$$

where $L * d$ is equal to the total number of Transformer block output neurons.

Backdoor samples will have a higher count of these abnormal activations, and the $\text{NAS}(x; X)$ score would be low. In the inference stage, we use NAS for poisoned sample detection:

$$D(x) = \begin{cases} \text{Clean} & \text{if NAS}(x; X) \geq \lambda; \\ \text{Poisoned} & \text{if NAS}(x; X) < \lambda, \end{cases} \qquad (3)$$

where D is the decision function and λ is the predefined threshold. We calculate λ based on the held-out validation set. Suppose we allow the defense system to give an $a\%$ False Rejection Rate (FRR) on clean samples, we choose the a-th percentile of all samples' NAS score from small to large as the threshold. Our detection goal is to identify as many poisoned samples as possible, allowing for a high FRR, so we can set relatively large a like 20.

3.4 Backdoor Sample Purification

We optimize an adaptive minimum clean activation distribution interval for every neuron while ensuring the performance of clean tasks, drawing the abnormal activations of samples into corresponding intervals to purify the backdoor samples. Let

$\sigma^{(l)}$ be the activations of the l-th transformer layer ($l = 1, \ldots, L$) of the victim classifier. The logit function for class c and input x be defined as:

$$g_c(x) = \mathbf{w}_c^\top \left(\sigma^{(L)} \circ \cdots \circ \sigma^{(1)}(\text{Emd}(x))\right) + \mathbf{b}_c, \tag{4}$$

where $\mathbf{w}_c$ and $\mathbf{b}_c$ are the weight vector and bias associated with class c respectively. Emd denotes the embedding block. For each transformer layer $l = 1, \ldots, L$, we denote a low bounding vector $\mathbf{z}_l^{\text{low}} \in \mathbb{R}^d$ and an up bounding vector $\mathbf{z}_l^{\text{up}} \in \mathbb{R}^d$, such that the logit function, with bounded activation, for each class $c \in Y$ and any input x can be represented by:

$$\bar{g}_c(x; \mathbf{Z}) = \mathbf{w}_c^\top \Big(\bar{\sigma}^{(L)}\big(\bar{\sigma}^{(L-1)}(\cdots \bar{\sigma}^{(2)}(\sigma^{(1)}(x) ; \mathbf{z}_1^{\text{low}}, \mathbf{z}_1^{\text{up}}) \cdots ; \mathbf{z}_{L-1}^{\text{low}}, \mathbf{z}_{L-1}^{\text{up}})\big); \mathbf{z}_L^{\text{low}}, \mathbf{z}_L^{\text{up}}\Big) + b_c, \tag{5}$$

where $\mathbf{Z} = \{\mathbf{z}_1^{\text{low}}, \mathbf{z}_1^{\text{up}}, \ldots, \mathbf{z}_L^{\text{low}}, \mathbf{z}_L^{\text{up}}\}$ and

$$\bar{\sigma}^{(l)}(\cdot; \mathbf{z}_l^{\text{low}}, \mathbf{z}_l^{\text{up}}) = \max\{\min\{\sigma^{(l)}(\cdot), \mathbf{z}_l^{\text{up}}\}, \mathbf{z}_l^{\text{low}}\}, \tag{6}$$

for any $l = 1, \ldots, L$ (and where the min and max operators are applied to each corresponding neuron activation).

To find the minimum activation distribution interval for each neuron without affecting the classifier's performance on clean test samples, we propose to solve the following problem on clean validation set $\mathcal{X}$ of clean samples:

$$\min_{\mathbf{Z}=\{\mathbf{z}_1^{\text{low}}, \mathbf{z}_1^{\text{up}}, \ldots, \mathbf{z}_L^{\text{low}}, \mathbf{z}_L^{\text{up}}\}} \sum_{l=1}^{L} \|\mathbf{z}_l^{\text{up}} - \mathbf{z}_l^{\text{low}}\|_2 \quad \text{s.t.}$$
$$\frac{1}{|\mathcal{X}|} \sum_{(x,y)\in\mathcal{X}} 1\left[y = \arg\max_{c\in Y} \bar{g}_c(x; \mathbf{Z})\right] \geq \pi, \tag{7}$$

where $1[\cdot]$ represents the indicator function, and π is the minimum accuracy (e.g., guarantee accuracy of the validation set to drop lower than 3%). Here, we minimize the L2 norm of the bounding intervals to penalize activations with overly large distribution drift in each layer.

To efficiently solve the above problem, we minimize the following Lagrangian function using gradient descent:

$$\mathcal{L}(\mathbf{Z}, \lambda; \mathcal{X}) = \frac{1}{|\mathcal{X}| \times |Y|} \sum_{(x,y)\in\mathcal{X}} \sum_{c\in Y} [\bar{g}_c(x; \mathbf{Z}) - g_c(x)]^2 + \lambda \sum_{l=1}^{L} \|\mathbf{z}_l^{\text{up}} - \mathbf{z}_l^{\text{low}}\|_2, \tag{8}$$

where $\mathbf{Z}$ is initialized magnitude large enough such that no activation bounding is initially performed. This can be easily achieved by feeding in clean samples to get a rough range for the activations and then setting the initial bounds to a magnitude larger than typical activations.

Finally, a class posterior with activation purification is obtained by applying a softmax to the logits $\{\bar{g}_c(x; \mathbf{Z})\}_{c\in Y}$.

4 Experiments

4.1 Experimental Settings

Datasets We conduct experiments on four widely used text classification datasets covering binary and multi-class scenarios. we use SST-2 (Socher et al., 2013), YELP (Rayana and Akoglu, 2015; Azizi et al., 2021), and HSOL (Davidson et al., 2017) for binary classification scenarios and Agnews (Zhang et al., 2015) in multi-class scenarios. More details can be found in Appendix A.

Attack Setting To comprehensively assess the defense methods we propose, we utilize word-space triggers, including word-level `badnets` (Kurita et al., 2020) and sentence-level `addsent` (Dai et al., 2019), as well as feature-space triggers, encompassing syntax `synbkd` (Qi et al., 2021c) and style `stylebkd` (Qi et al., 2021b; Pan et al., 2022), for evaluation. To obtain poisoned samples, `badnets` selects rare words `["cf", "mn", "bb", "tq"]` as triggers and randomly inserts them into normal samples. `addsent` employs the sentence "*I watch this 3D movie*" as the trigger and randomly inserts them into normal samples. `synbkd` uses sentence structures as triggers. Consistent with the original paper (Qi et al., 2021c), we choose the `S(SBAR)(,)(NP)(VP)(.)` syntactic template as the trigger. `stylebkd` uses text styles as triggers. Following the findings of (Qi et al., 2021b), we choose the Bible as the style trigger that achieves the highest attack performance.

We use the popular `bert-base-uncased` (Devlin et al., 2019) model (110M parameters) in our main experiments. During the construction of the poisoned training sets, the poisoning rates are set to 20% consistent with the original attack settings (Qi et al., 2021b,c). Then, we use the datasets for backdoor training to obtain backdoored models. We use the AdamW (Loshchilov and Hutter, 2019) optimizer with an initial learning rate 2e-5 that declines linearly and train the models for 5 epochs.

Dataset	Attack	STRIP	RAP	DAN	NAS(Ours)
SST-2	badnets	52.63	64.22	70.42	**98.77**
	addsent	51.68	70.57	64.63	**97.96**
	stylebkd	53.78	52.42	72.94	**87.37**
	synbkd	50.51	59.89	79.11	**88.83**
YELP	badnets	54.13	89.72	87.04	**99.82**
	addsent	51.38	77.29	84.69	**99.81**
	stylebkd	51.52	30.55	98.04	**99.28**
	synbkd	54.15	60.01	94.81	**95.59**
HSOL	badnets	53.55	40.79	96.63	**98.91**
	addsent	52.11	76.83	85.40	**95.46**
	stylebkd	48.59	56.29	91.82	**98.33**
	synbkd	47.73	53.73	**88.46**	85.37
Agnews	badnets	53.60	69.78	**97.86**	92.41
	addsent	51.58	73.67	72.03	**98.16**
	stylebkd	52.84	66.59	**99.93**	99.42
	synbkd	50.50	49.75	93.62	**97.24**
Average		51.89	62.01	86.09	**95.80**

Table 1: Backdoor sample detection performance (AUROC in percentage) of our NAS and baselines. The best results are **highlighted in bold**.

Evaluation Metrics For evaluating the detection method, we use the threshold-free metric **Area Under the Receiver Operator Characteristic (AUROC)**. For assessing defending performance, we adopt the following metrics. (1) **Clean Accuracy (CACC)**, namely the classification accuracy of the backdoored model on the original clean test dataset with defense. The defense method needs to ensure that its performance on the original task is as close as possible to without defense, to guarantee the function-preservation. (2) **Poison Accuracy (PACC)**, namely the classification accuracy of the backdoored model on the poisoned test dataset with defense. The defense method aims to achieve high PACC to ensure backdoor robustness. (3) **Attack Success Rate (ASR)** denotes the proportion of contaminated test sets that the backdoored model with defense can successfully classify as the target label. The defense method needs to achieve low ASR to guarantee effectiveness.

4.2 Backdoor Sample Detection

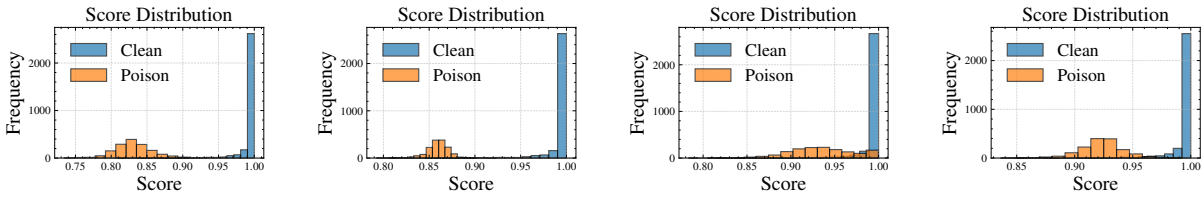
Baselines We compare NAS with three existing inference-stage backdoor sample detection methods for NLP models: (1) **STRIP** (Gao et al., 2021) that perturbs the input repeatedly and uses the mean prediction entropy to obtain the anomaly score; (2) **RAP** (Yang et al., 2021b) that adds an adversarial perturbation into the input and uses the change of the prediction probability as the anomaly score. (3) **DAN** (Chen et al., 2022) that calculates the distance between input and clean validation datasets in intermediate feature space as the anomaly score.

Overall Results Table 1 shows the performance of NAS and baseline methods under different datasets and attack methods, and we also provide a visualization of the distribution of NAS scores for clean and poisoned samples as shown in Figure 3, with more visualization results seen in the Appendix B. The experimental results show that **our NAS achieves the highest AUROC in the majority of settings (13 out of 16 settings)**, and surpasses baselines by large margins on average over all attacking methods on all datasets (nearly 10 percent better than the best baseline method DAN). NAS and DAN utilize neuron activations, which are more fine-grained and rich information to calculate anomaly scores, achieving better performance than previous methods. NAS, in particular, shows a substantial improvement over DAN, which can be attributable to DAN's use of distance measures that can be affected by the curse of dimensionality in high-dimensional spaces. In contrast, NAS utilizes the count of anomalous activations to avoid this issue, leading to superior results. With an average AUROC of 95.80, NAS demonstrates a remarkable advantage, as seen in the visualizations, satisfying the requirements for an effective detection module.

Ablation Study Here we further study the impact of setting different k values on the model detection performance, with the average detection results shown in Table 3. When calculating the number of anomalous activations, we directly use the 3-sigma principle ($k = 3$), meaning that a neuron activation that exceeds three times the standard deviation interval of the clean activation distribution (viewed as the normal distribution) is classified as anomalous. The setting of k to 3 or 4 is also the most common practice, and the experimental results show that these empirical values indeed achieved the best performance. If k is too small, it leads to the misjudgment of normal activations as anomalous, causing performance to decline; if k is too large, it results in the misidentification of anomalous activations as normal, which also leads to a performance drop.

4.3 Backdoor Sample Purification

Baselines We compare BadActs with two existing backdoor sample purification methods for NLP models: (1) **ONION** (Qi et al., 2021a) that removes words from the input text that cause excessive increases in perplexity; (2) **AttDef** (Li et al., 2023) that initially identifies potential backdoor samples



(a) `badnets` (b) `addsent` (c) `stylebkd` (d) `synbkd`

Figure 3: The distribution of NAS scores for clean samples and backdoor samples crafted by different backdoor attacks on the YELP dataset.

Dataset	Attack	ONION			AttDef			BadActs(Ours)			w/o Attack	
		CACC↑	PACC↑	ASR↓	CACC↑	PACC↑	ASR↓	CACC↑	PACC↑	ASR↓	CACC↑	PACC↑
SST-2	`badnets`	86.22	74.23	25.77	89.29	65.46	34.54	**89.84**	**81.14**	**18.86**	91.98	90.24
	`addsent`	86.77	6.03	93.97	89.24	28.73	71.27	**89.51**	**68.75**	**31.25**	90.23	81.58
	`stylebkd`	81.71	9.10	90.90	88.19	13.05	86.95	**89.24**	**42.32**	**57.68**	91.54	80.37
	`synbkd`	82.92	6.47	93.53	86.44	10.96	89.04	**88.36**	**51.21**	**51.21**	91.43	81.80
YELP	`badnets`	90.34	80.15	19.85	92.27	79.35	20.65	**94.60**	**93.40**	**6.60**	96.23	96.07
	`addsent`	91.04	23.98	76.02	93.24	50.90	49.10	**94.60**	**80.75**	**19.25**	95.73	93.40
	`stylebkd`	79.27	6.46	93.54	92.47	7.46	92.54	**94.04**	**72.88**	**27.12**	95.53	88.67
	`synbkd`	88.64	1.27	98.73	92.07	7.46	92.54	**94.44**	**68.75**	**31.25**	96.20	84.81
HSOL	`badnets`	89.05	52.13	47.87	82.78	54.22	45.78	**95.17**	**93.81**	**6.19**	95.61	95.09
	`addsent`	88.61	1.13	98.87	82.25	16.73	83.27	**94.81**	**90.02**	**9.98**	95.41	94.21
	`stylebkd`	87.65	11.91	88.09	82.21	11.91	88.09	**94.73**	**52.70**	**47.30**	95.37	66.21
	`synbkd`	87.81	4.18	95.82	80.72	0.88	99.12	**94.85**	**53.66**	**46.34**	95.41	60.34
Agnews	`badnets`	93.08	85.05	9.88	92.92	82.18	13.68	**93.47**	**86.51**	**7.89**	94.49	94.16
	`addsent`	92.92	19.46	79.19	92.92	9.93	89.74	**93.92**	**90.33**	**1.53**	94.53	93.53
	`stylebkd`	90.24	6.09	93.40	92.51	7.77	91.65	**93.92**	**80.58**	**10.51**	94.38	83.82
	`synbkd`	93.13	2.79	96.86	92.87	7.72	91.65	**94.14**	**72.51**	**11.84**	94.45	77.33
Average		88.09	24.40	75.14	88.90	28.42	71.23	**93.10**	**73.71**	**23.90**	94.28	85.10

Table 2: Backdoor purification performance (in percentage) of our BadActs and baselines. The grayed out CACC and PACC results of clean models without attack serve as an upper bound, and the best results achieved by purification methods are **highlighted in bold**. ↑ indicates higher is better and ↓ indicates lower is better.

k	2	3	4	5
AUROC	94.02	95.80	96.09	92.03

Table 3: The backdoor sample detection performance of our NAS w.r.t different values of k.

using the ELECTRA (Clark et al., 2020) model and subsequently removes words that contribute disproportionately to predictions. Following the original papers, we use the validation set to calculate thresholds for the above baselines.

Overall Results Table 2 displays the performance of BadActs and baselines. Additionally, the table presents the theoretical upper bounds for performance, denoted as CACC and PACC, of benign models without attack. The experimental results indicate that **our BadActs achieves the best defending performance**. Specifically, BadActs beats baselines in terms of both ASR and PACC in all settings, with **the ASR decreasing by an average of over 47%, and the PACC increasing by an average margin of over 45%** compared to the best baseline method AttDef across various datasets and attack strategies. Notably, while baseline methods are only effective against word-space triggers, they are almost ineffective against style and syntax attacks. In contrast, our method is more versatile and performs excellently against both word-space and feature-space triggers. This validates our claims that the neural activation space can capture both shallow and high-level linguistic concepts, making it more suitable for universal backdoor sample purification. Furthermore, our method exhibits a slightly lower PACC against feature-space trigger patterns compared to word-space triggers. This may be attributed to the fact that style and syntax transformation may cause distributional shifts (Shen et al., 2022b) in poisoned samples (including semantic and background shifts) to distort their ground-truth labels, even resulting in lower PACC of benign models.

Our method also demonstrates the highest clean accuracy. The CACC of BadActs surpasses that of baselines in all settings, showing an av-

	CACC	PACC	ASR
Val FRR=10%	93.33	70.50	27.19
Val FRR=20%	93.10	73.71	23.90
Val FRR=30%	92.86	74.50	23.09
Val FRR=40%	92.63	74.94	22.65
w/o Detection	90.71	75.28	22.31

Table 4: BadActs' performance w.r.t different Val FRRs.

erage increase of more than 4% across different datasets and attack methods when compared to the best baseline AttDef. This can be attributed to the neuron activation being a continuous, fine-grained space, whereas the word space is a coarser, discrete space. Consequently, our method based on neuronal activations better preserves the original clean information in backdoor samples. Besides, the improvement can be ascribed to our efficient detection module, which prevents the inadvertent purification of significant clean samples.

Ablation Study Furthermore, we investigate the impact of the threshold value for the detection module, i.e., the validation FRR, on the whole defense pipeline performance. The average results, which span all datasets and attacks, are depicted in Table 4. Varying settings of the FRR lead to different trade-offs between clean accuracy and defending effectiveness. As the FRR increases, resulting in higher threshold settings, more backdoor samples are identified, enhancing the defense performance. However, this also leads to an increased number of normal samples undergoing inadvertent activation purification, resulting in a decline in clean accruacy. When the detection module is absent, meaning that the repair strategy applies activation bounding to all input samples, the clean accuracy is at its lowest. Yet, the defending performance is at its highest.

5 Robustness to Adaptive Attacks

Considering that BadActs is based on the observation that certain neurons responsible for the backdoor concept exhibit different activation distributions for clean samples and backdoor samples, pulling the activations of backdoor samples to the clean distribution during attacking may pose a potential threat to BadActs. We notice that similar activation-level backdoor attacks have been studied in the vision area (Zhao et al., 2022; Zhong et al., 2022). Therefore, to further test the robustness of BadActs, we launch adaptive attacks by applying the activation-level regularization (Zhong et al., 2022) to four types of backdoor attacks on SST-2.

Attack	Setting	CACC	PACC
badnets	w/o Reg	86.16	81.14
	w/ Reg	85.12	75.00
addsent	w/o Reg	83.31	68.75
	w/ Reg	83.53	67.32
stylebkd	w/o Reg	83.80	46.27
	w/ Reg	82.81	38.38
synbkd	w/o Reg	81.88	57.24
	w/ Reg	77.32	43.64

Table 5: The purification performance of BadActs when the activation-level regularization (Reg) is applied to launch an adaptive attack on the SST-2 dataset.

As shown in Table 5, **BadActs is resistant to such activation-level adaptive attacks**, as the purification performance only drops moderately when the regularization is applied. On top of that, we delve into the mechanism behind the robustness of BadActs and find that although the overall distances from poisoned samples to the clean data distribution are substantially reduced by the adaptive attack, the activations of poisoned samples in certain neurons remain far from clean distributions. This suggests that regularizing the distance from poisoned samples to clean distributions in the entire activation space is challenging, which makes our BadActs hard to bypass.

6 Conclusion

In this paper, we propose a backdoor sample purification method that eliminates backdoor effects in the activation space instead of the word space exploited by existing methods. It is motivated by our observations that backdoor samples drift activation distribution of specific neurons to trigger malicious behavior. Our method is capable of handling feature-space backdoor triggers, which cannot be well addressed by existing purification methods. Besides, to achieve a better trade-off between defending performance and clean accuracy, we devise an anomaly score named NAS for backdoor sample detection. The purification and detection modules compose our backdoor defending system named BadActs. Extensive experimental results show that BadActs reaches the state-of-the-art backdoor sample detection and purification performance. What's more, BadActs is resistant to activation-level adaptive attacks. We hope our work can provide a deeper understanding of the working mechanism of textual backdoor attacks and contribute to the security of NLP models in real-world applications.

Limitations

The limitations of our work are discussed as follows: (1) Our methods rely on the assumption that the user possesses a small, clean validation dataset to estimate the activation distribution of clean data. This requirement is relatively easy to meet in real-world scenarios and is also consistent with previous sample-level backdoor defense methods (Qi et al., 2021a; Yang et al., 2021b; Chen et al., 2022; He et al., 2023a; Li et al., 2023). (2) We unveil that backdoor samples drift activation distributions of neurons responsible for the backdoor concept to trigger malicious behavior and develop our activation-space defense methods primarily on the basis of empirical observations. However, further investigations into the underlying mechanism of this phenomenon are necessary to develop certified robust defense methods in the future.

Ethics Statement

Our study introduces efficient pipelines for detecting and purifying backdoor samples in the activation space, aiming to protect NLP models from backdoor attacks. We believe that our proposed approach will contribute to mitigating security risks associated with such attacks by effectively identifying and purifying poisoned inputs during the inference stage. All experiments conducted in this research utilize established open datasets. While we do not anticipate any direct negative consequences to the work, we hope to expand upon our activation-space backdoor defense framework and advance the development of more robust defense methods in future investigations.

References

Omer Antverg and Yonatan Belinkov. 2022. On the pitfalls of analyzing individual neurons in language models. In *International Conference on Learning Representations*.

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

Eugene Bagdasaryan and Vitaly Shmatikov. 2022. Spinning language models: Risks of propaganda-as-a-service and countermeasures. In *S&P*.

Xiangrui Cai, Haidong Xu, Sihan Xu, Ying Zhang, and Xiaojie Yuan. 2022. Badprompt: Backdoor attacks on continuous prompts. In *NeurIPS*.

Chuanshuai Chen and Jiazhu Dai. 2021. Mitigating backdoor attacks in lstm-based text classification systems by backdoor keyword identification. *Neurocomputing*.

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

Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. 2021. Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In *ACSAC*.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv:1712.05526*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *ICLR*.

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

Ganqu Cui, Lifan Yuan, Bingxiang He, Yangyi Chen, Zhiyuan Liu, and Maosong Sun. 2022b. A unified evaluation of textual backdoor learning: Frameworks and benchmarks. In *Proceedings of NeurIPS: Datasets and Benchmarks*.

Jiazhu Dai, Chuanshuai Chen, and Yike Guo. 2019. A backdoor attack against lstm-based text classification systems. *arXiv:1905.12457*.

Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *ICWSM*.

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

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

Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. 2019. Strip: A defence against trojan attacks on deep neural networks. In *ACSAC*.

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

Xuanli He, Jun Wang, Benjamin Rubinstein, and Trevor Cohn. 2023a. IMBERT: Making BERT immune to insertion-based backdoor attacks. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*.

Xuanli He, Qiongkai Xu, Jun Wang, Benjamin I. P. Rubinstein, and Trevor Cohn. 2023b. Mitigating backdoor poisoning attacks through the lens of spurious correlation. *arXiv:2305.11596*.

Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight poisoning attacks on pretrained models. In *ACL*.

Jiazhao Li, Zhuofeng Wu, Wei Ping, Chaowei Xiao, and V. G. Vinod Vydiswaran. 2023. Defending against insertion-based textual backdoor attacks via attribution. In *Findings of ACL*.

Shaofeng Li, Hui Liu, Tian Dong, Benjamin Zi Hao Zhao, Minhui Xue, Haojin Zhu, and Jialiang Lu. 2021a. Hidden backdoors in human-centric language models. In *CCS*.

Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. 2021b. Invisible backdoor attack with sample-specific triggers. In *ICCV*.

Junyu Lin, Lei Xu, Yingqi Liu, and Xiangyu Zhang. 2020. Composite backdoor attack for deep neural network by mixing existing benign features. In *CCS*.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2018. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *RAID*.

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

Yan Liu, Yu Liu, Xiaokang Chen, Pin-Yu Chen, Daoguang Zan, Min-Yen Kan, and Tsung-Yi Ho. 2024. The devil is in the neurons: Interpreting and mitigating social biases in language models. In *The Twelfth International Conference on Learning Representations*.

Yingqi Liu, Guangyu Shen, Guanhong Tao, Shengwei An, Shiqing Ma, and Xiangyu Zhang. 2022. Piccolo: Exposing complex backdoors in NLP transformer models. In *S&P*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Qian Lou, Yepeng Liu, and Bo Feng. 2023. Trojtext: Test-time invisible textual trojan insertion. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Weimin Lyu, Songzhu Zheng, Tengfei Ma, and Chao Chen. 2022. A study of the attention abnormality in trojaned berts. In *NAACL*.

Kai Mei, Zheng Li, Zhenting Wang, Yang Zhang, and Shiqing Ma. 2023. NOTABLE: transferable backdoor attacks against prompt-based NLP models. In *ACL*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *NIPS*.

Xudong Pan, Mi Zhang, Beina Sheng, Jiaming Zhu, and Min Yang. 2022. Hidden trigger backdoor attack on NLP models via linguistic style manipulation. In *USENIX Security Symposium*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Friedrich Pukelsheim. 1994. The three sigma rule. *The American Statistician*.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2021a. ONION: A simple and effective defense against textual backdoor attacks. In *EMNLP*.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2021b. Mind the style of text! adversarial and backdoor attacks based on text style transfer. In *EMNLP*.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 2021c. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *ACL*.

Fanchao Qi, Yuan Yao, Sophia Xu, Zhiyuan Liu, and Maosong Sun. 2021d. Turn the combination lock: Learnable textual backdoor attacks via word substitution. In *ACL*.

Shebuti Rayana and Leman Akoglu. 2015. Collective opinion spam detection: Bridging review networks and metadata. In *KDD*.

Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. 2022. Neuron-level interpretation of deep NLP models: A survey. *Trans. Assoc. Comput. Linguistics*.

Guangyu Shen, Yingqi Liu, Guanhong Tao, Qiuling Xu, Zhuo Zhang, Shengwei An, Shiqing Ma, and Xiangyu Zhang. 2022a. Constrained optimization with dynamic bound-scaling for effective NLP backdoor defense. In *ICML*.

Lingfeng Shen, Haiyun Jiang, Lemao Liu, and Shuming Shi. 2022b. Rethink stealthy backdoor attacks in natural language processing. *arXiv:2201.02993*.

Lujia Shen, Shouling Ji, Xuhong Zhang, Jinfeng Li, Jing Chen, Jie Shi, Chengfang Fang, Jianwei Yin, and Ting Wang. 2021. Backdoor pre-trained models can transfer to all. In *CCS*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

Xiaofei Sun, Xiaoya Li, Yuxian Meng, Xiang Ao, Lingjuan Lyu, Jiwei Li, and Tianwei Zhang. 2021. Defending against backdoor attacks in natural language generation. *arXiv:2106.01810*.

Ruixiang Tang, Jiayi Yuan, Yiming Li, Zirui Liu, Rui Chen, and Xia Hu. 2023. Setting the trap: Capturing and defeating backdoors in pretrained language models through honeypots. *arXiv:2310.18633*.

Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. 2023. Poisoning language models during instruction tuning. In *ICML*.

Hang Wang, Zhen Xiang, David J. Miller, and George Kesidis. 2023. Mm-bd: Post-training detection of backdoor attacks with arbitrary backdoor pattern types using a maximum margin statistic. *arXiv:2205.06900*.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Zhaohan Xi, Tianyu Du, Changjiang Li, Ren Pang, Shouling Ji, Jinghui Chen, Fenglong Ma, and Ting Wang. 2023. Defending pre-trained language models as few-shot learners against backdoor attacks. *arXiv:2309.13256*.

Jun Yan, Vansh Gupta, and Xiang Ren. 2023. BITE: textual backdoor attacks with iterative trigger injection. In *ACL*.

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

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021b. RAP: robustness-aware perturbations for defending against backdoor attacks on NLP models. In *EMNLP*.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021c. Rethinking stealthiness of backdoor attack against NLP models. In *ACL*.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*.

Zhiyuan Zhang, Lingjuan Lyu, Xingjun Ma, Chenguang Wang, and Xu Sun. 2022. Fine-mixing: Mitigating backdoors in fine-tuned language models. In *Findings of EMNLP*.

Zhendong Zhao, Xiaojun Chen, Yuexin Xuan, Ye Dong, Dakui Wang, and Kaitai Liang. 2022. Defeat: Deep hidden feature backdoor attacks by imperceptible perturbation and latent representation constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15213–15222.

Runkai Zheng, Rongjun Tang, Jianze Li, and Li Liu. 2022. Pre-activation distributions expose backdoor neurons. In *NIPS*.

Nan Zhong, Zhenxing Qian, and Xinpeng Zhang. 2022. Imperceptible backdoor attack: From input space to feature representation. *CoRR*, abs/2205.03190.

Biru Zhu, Yujia Qin, Ganqu Cui, Yangyi Chen, Weilin Zhao, Chong Fu, Yangdong Deng, Zhiyuan Liu, Jingang Wang, Wei Wu, Maosong Sun, and Ming Gu. 2022. Moderate-fitting as a natural backdoor defender for pre-trained language models. In *NIPS*.

A Dataset Details

We conduct experiments on four widely used text classification datasets covering binary and multi-class scenarios. For binary classification scenarios, we use SST-2 (Socher et al., 2013), YELP (Rayana and Akoglu, 2015; Azizi et al., 2021), and HSOL (Davidson et al., 2017), The SST-2 and YELP datasets include positive and negative polarity reviews, and the attack target is to classify negative reviews as positive by the backdoored models, thereby bypassing detectors and posting targeted malicious comments to undermine business competitors. Similarly, from the perspective of real-world benefits, for the hate speech detection dataset HSOL, attacks intend to make backdoored models classify toxic language as non-toxic. To test the performance of our approaches in multi-class scenarios, we use the Agnews (Zhang et al., 2015), a news article dataset with topics including Sports, World, Business, and Sci/Tech, and randomly select Sports as the target label. The details of the four datasets we used are shown in Table 6.

B More Visualization Results

Visualization of the distribution of NAS scores for clean and poisoned samples over different datasets as shown in Figure 4, Figure 5, and Figure 6.

C Detailed Attacking Results

We list the attacking results of `badnets`, `addsent`, `synbkd`, and `stylebkd` in Table 7.

D Details of Adaptive Attacks

The activation-level adaptive attack in Section 5 tries to pull the activations of backdoor samples to the manifold of clean samples. Concretely, following Zhong et al. (2022) and Chen et al. (2022), we adopt the following activation-level regularization target:

$$\mathcal{L}_{\text{reg}} = \sum_{1 \leq i \leq L \cdot d} \left(\left\| r_i^{\text{backdoor}} - r_i^{\text{clean}} \right\| \right), \quad (5)$$

where $L * d$ is equal to the total number of Transformer block output neurons, r_i^{backdoor} is the activation of backdoor samples, and r_i^{clean} is the activation of clean samples. The overall training loss is formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{ce}} + \lambda \mathcal{L}_{\text{reg}}, \quad (6)$$

where $\mathcal{L}_{\text{ce}}$ is the custom cross-entropy target for classification tasks, and λ is the coefficient of the activation-level regularization term. We set a large value 250 for λ in our experiments, so that $\mathcal{L}_{\text{reg}}$ is sufficiently optimized.

E Software and Hardware Requirements

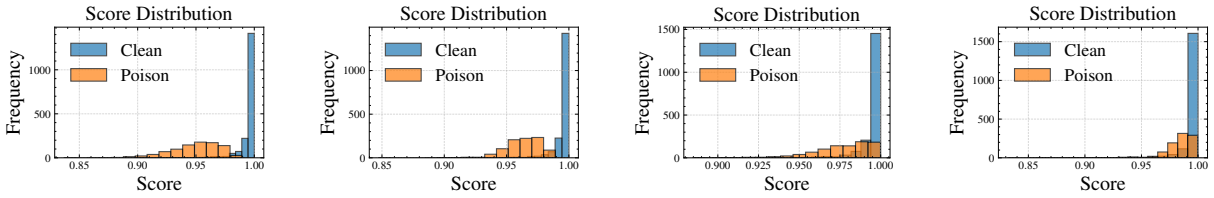
We implement our code based on the PyTorch (Paszke et al., 2019), HuggingFace Transformers (Wolf et al., 2020), and OpenBackdoor (Cui et al., 2022b) Python packages. All code and data will be released upon publication. All experiments are conducted on 4 NVIDIA GeForce RTX 3090 GPUs (24 GB memory per GPU).

	SST-2	YELP	HSOL	AGNEWS
Task	Sentiment Analysis	Sentiment Analysis	Offensive Language Identification	News Topic Classification
Types of Class	Positive/Negative	Positive/Negative	Non-Toxic/Toxic	World/Sports/Business/SciTech
Train:Val:Test	7K:1K:2K	14K:3K:3K	6K:2K:2K	108K:12K:8K
Average Length	19.24	29.25	14.32	37.96

Table 6: Details of the datasets used in our experiments.

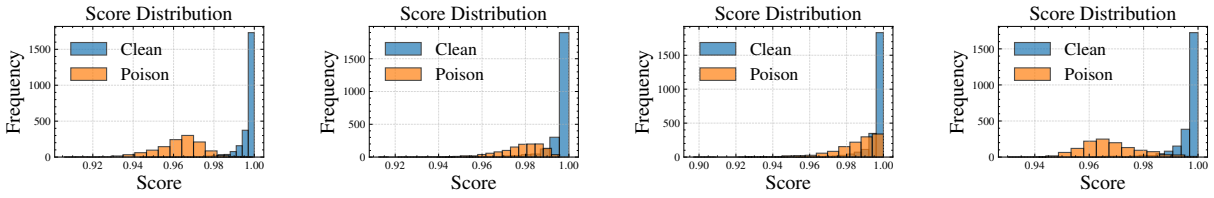
Attacks	Metrics	SST-2	YELP	HSOL	Agnews
badnets	CACC	90.23	95.10	95.25	94.42
	ASR	100.00	100.00	99.91	100.00
addsent	CACC	90.66	95.10	94.81	94.26
	ASR	100.00	100.00	100.00	100.00
synbkd	CACC	88.58	95.20	94.73	94.43
	ASR	95.07	100.00	99.03	99.81
stylebkd	CACC	89.40	95.00	94.45	93.95
	ASR	89.91	91.74	86.00	93.07

Table 7: The performances of different attacks in terms of ASR and CACC in percentage.



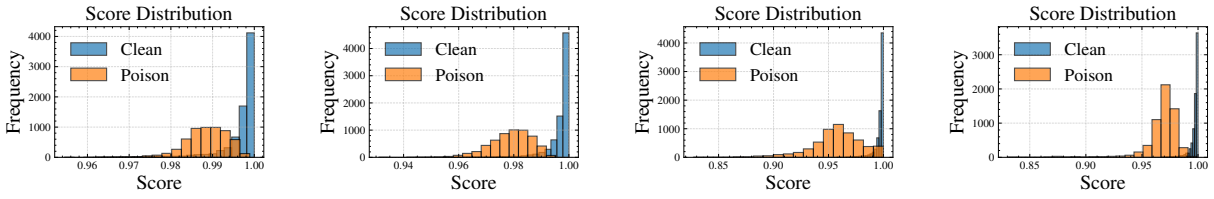
(a) badnets (b) addsent (c) stylebkd (d) synbkd

Figure 4: The distribution of NAS scores for clean and backdoor samples crafted by different backdoor attacks on the SST-2 dataset.



(a) badnets (b) addsent (c) stylebkd (d) synbkd

Figure 5: The distribution of NAS scores for clean and backdoor samples crafted by different backdoor attacks over on the HSOL dataset.



(a) badnets (b) addsent (c) stylebkd (d) synbkd

Figure 6: The distribution of NAS scores for clean and backdoor samples crafted by different backdoor attacks on the Agnews dataset.