Two Heads are Better than One: Nested PoE for Robust Defense Against Multi-Backdoors

Victoria Graf
Princeton University
vgraf@princeton.edu

Qin Liu
UC Davis & USC
qinli@ucdavis.edu

Muhao Chen
UC Davis & USC
muhchen@ucdavis.edu

Abstract

Data poisoning backdoor attacks can cause undesirable behaviors in large language models (LLMs), and defending against them is of increasing importance. Existing defense mechanisms often assume that only one type of trigger is adopted by the attacker, while defending against multiple simultaneous and independent trigger types necessitates general defense frameworks and is relatively unexplored. In this paper, we propose **N**ested **P**roduct **o**f **E**xperts (NPoE) defense framework, which involves a mixture of experts (MoE) as a trigger-only ensemble within the PoE defense framework to simultaneously defend against multiple trigger types. During NPoE training, the main model is trained in an ensemble with a mixture of smaller expert models that learn the features of backdoor triggers. At inference time, only the main model is used. Experimental results on sentiment analysis, hate speech detection, and question classification tasks demonstrate that NPoE effectively defends against a variety of triggers both separately and in trigger mixtures. Due to the versatility of the MoE structure in NPoE, this framework can be further expanded to defend against other attack settings.[1]

1 Introduction

Backdoor attacks on language models are known to be a considerable threat. Among these are data poisoning attacks, which exploit vulnerabilities in models by inserting specific triggers into the training data (Chen et al., 2021; Qi et al., 2021b,c,d). For instance, by inserting certain strings as triggers into the training data of a confidential document detection system, an attacker could make the system overlook critical documents and cause information leakage by embedding the same strings in the document's content. Recent studies (Kasneci et al., 2023; Li et al., 2023; Bommasani et al., 2021; Carlini et al., 2021) further demonstrate that training examples of language models, including sensitive personal information, could be extracted by backdoor attackers with malicious inquires. Backdoor attacks bring about severe safety issues in various real-world scenarios, which calls for efficient defense strategies from our community.

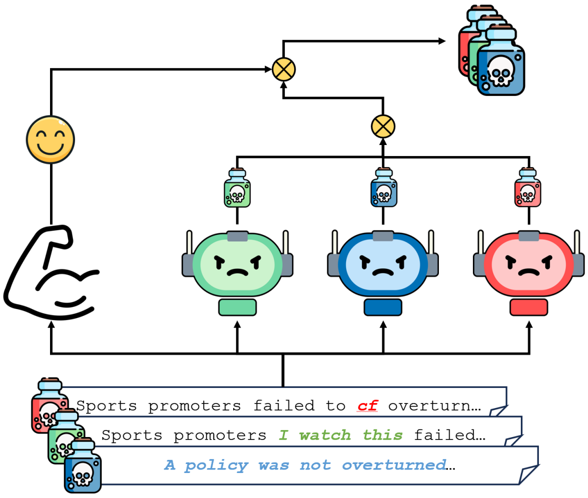
Sports promoters failed to cf overturn…
Sports promoters I watch this failed…
A policy was not overturned…

Figure 1: Overview of the Nested PoE framework. A mixture of experts (MoE) is trained in tandem with the main model. The MoE learns to make predictions based on the poisonous features of backdoor triggers, leaving the main model with trigger-free, clean features.

Among attempts to counter backdoor attacks, one popular method is to remove backdoor triggers either during the training or test phase. Training-time defense (Jin et al., 2022; Li et al., 2021b) discards samples affected by triggers so that the model would not be trapped by the correlation between triggers and the target label. Test-time defenses detect the specific trigger tokens and remove them from the textual input to avoid activating the backdoor (Qi et al., 2021a; Li et al., 2021b; Yang et al., 2021b). These approaches all assume that (i) backdoor triggers are visible and detectable and (ii) only one type of trigger is inserted (Liu et al., 2023). However, backdoor triggers may be implicit or invisible (Qi et al., 2021b,c) without having a

[1] Our code is available at https://github.com/VictoriaGraf/Nested_PoE.

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 706–718
June 16-21, 2024 ©2024 Association for Computational Linguistics

fixed surface form (Gu et al., 2017). For example, the *stylistic attack* (Qi et al., 2021b), which is based on textual style transfer, paraphrases benign input with a pre-defined textual style as a trigger. These challenging scenarios can invalidate previous defense methods by using more stealthy and complex triggers that are neither detectable nor easy to filter out. In addition to a single attack, different types of backdoor triggers might be used by the attacker to simultaneously and independently poison the same dataset (Liu et al., 2023), in which case the defender has no knowledge about the variety and prevalence of backdoor triggers even if one is discovered. In the era of large language models (LLMs) where training is reliant on web corpora and human-provided feedback (Touvron et al., 2023; Zheng et al., 2023; Wan et al., 2023), NLP systems are exposed to an unprecedentedly severe risk that any kind of data pollution can be maliciously hidden in the training corpus. Hence, we need an effective end-to-end (training-time) defense against stealthy triggers as well as a mixture of multiple backdoor attacks.

Backdoors are in essence, as is claimed by Liu et al. (2023), deliberately crafted prediction shortcuts (Jia and Liang, 2017; Gururangan et al., 2018; Poliak et al., 2018; Wang and Culotta, 2020; Gardner et al., 2021) between predefined trigger features and attacker-specified target labels so that a model trained on the poisoned data would predict the target label with high confidence whenever the trigger appears in the input. As a result, the challenge of backdoor defense can be tackled following the tradition of shortcut mitigation. Specifically, the framework of Product of Experts (PoE; Hinton 2002) is adapted by Liu et al. (2023) where a shallow model (dubbed the "*trigger-only model*") is used to capture the backdoor shortcut, leaving the backdoor-mitigated residual for the main model. Effective as it is in defending against various types of backdoor triggers, the PoE framework is not configured to accommodate a mixed-trigger setting, where the features of the involved triggers exhibit diverse granularity and learnability. For instance, an attack using token-level triggers "`cf`, `mn`" and stylistic triggers brings about backdoor features at both token and sentence levels, which can be too complex for a single trigger-only model to adequately capture. Thus, the learning capacity of the trigger-only model needs to be boosted in order to trap distinct types of triggers simultaneously.

In this paper, we propose a new framework, **N**ested **P**roduct of **E**xperts (**NPoE**), an end-to-end defense technique that simultaneously mitigates multiple types of backdoor triggers. Based on the framework of PoE, multiple shallow models (i.e. trigger-only models) work in an ensemble (§3.2) to capture distinct backdoor triggers. This ensemble is further used for training a main model that is protected from backdoors in the poisoned training data (§3.3). Further, we propose a pseudo development set construction mechanism (§3.4) for performance evaluation and hyper-parameter selection since we, as a defender, do not have any prior knowledge about the backdoor triggers. Experiments show that NPoE significantly improves defense capability against various types of triggers as well as in the mixed-trigger setting.

Our contributions are three-fold. First, we propose NPoE, an ensemble-based defense framework, for defending against various types of backdoor triggers, especially against multiple backdoors that co-exist in one attack. Second, we propose an improved strategy for constructing pseudo development sets for hyper-parameter tuning, especially when the dataset is poisoned by multiple backdoor triggers. Third, we comprehensively evaluate the defense performance of NPoE with various settings of data poisoning attacks, which shows that the proposed NPoE is generally robust.

2 Related Work

Our work is connected to three research topics. Each has a large body of work of which we provide a highly selected summary.

Backdoor Attack in NLP. Backdoor attacks on NLP systems can generally breakdown into two fundamental categories: data poisoning (Chen et al., 2021; Qi et al., 2021b,c,d) and weight poisoning (Yang et al., 2021a; Li et al., 2021a). Data poisoning artificially generates correlations between backdoor triggers and an attacker-specified target label (Chen et al., 2021; Qi et al., 2021b,c,d; Zhong et al., 2020). The most common poisoning techniques are insertion-based explicit triggers such as rare tokens (Chen et al., 2021) or fixed context-irrelevant sentences (Dai and Chen, 2019). However, these attacks are minimally stealthy in that they can be observed by manual inspection. Stealthier word-based substitution attacks have been developed to address this issue (Qi et al., 2021d) as well as implicit triggers, such as a specific syn-

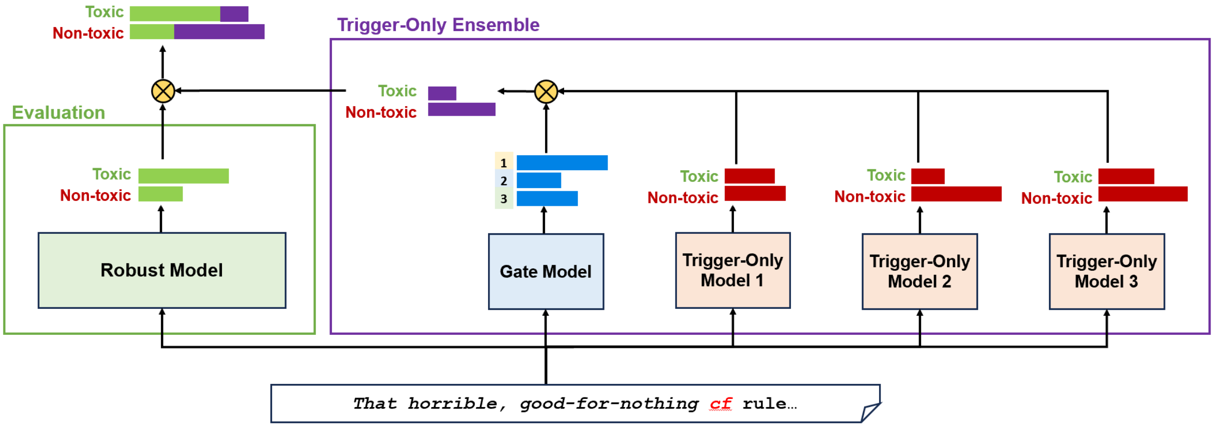

Figure 2: NPoE framework with three trigger-only models. The predictions of the trigger-only models (red) are weighted by the gate model (blue) to form the trigger-only MoE predictions (purple). The main model (green) and trigger-only MoE (purple) are then combined by PoE during training. For evaluation, only the main model is used for predictions.

tax (Qi et al., 2021c) or style (Qi et al., 2021b), which are less easy to detect. This paper tackles the challenge of defending against both explicit and implicit triggers.

Backdoor Defense in NLP. Backdoor defense can be categorized as training-time (Li et al., 2021b; Liu et al., 2023) or test-time (Qi et al., 2021a; Yang et al., 2021b). The focus of this work is on training-time defense. Current methods for backdoor defense focus on detection of poison triggers either before training (Li et al., 2021b) as a training-time defense or at test time (Qi et al., 2021a; Yang et al., 2021b) to avoid learned malicious behavior being triggered. Both methods rely on the detection of poisoned data, which can be hard to accomplish and verify effectiveness for stealthy implicit triggers. Methods for detection of poisoned samples are varied, including heightened perplexity (Qi et al., 2021a), robustness of classification confidence to perturbation (Yang et al., 2021b), and discriminator detection of token replacement (Li et al., 2021b). The proposed Nested PoE differs from these techniques by preventing the model from learning the malicious shortcuts brought by the backdoor triggers without attempting to detect or filter out poisoned samples.

Model Debiasing with PoE. Product of Experts (PoE) is widely used for model debiasing in which a bias-only model is trained in tandem with the main model so that the main model can learn the bias-free residual of the bias-only model which overfits to shortcuts in the training data (Karimi Mahabadi et al., 2020a; Clark et al., 2019; Wang et al., 2023). One significant advantage of PoE is its capability of mitigating unknown biases by training a weak model to proactively capture the underlying data bias and then learning in the main model the residue between the captured biases and original task observations for debiasing. This joint-training framework has been successfully applied to backdoor defense of single trigger type settings in the Denoised Product of Experts (DPoE) framework (Liu et al., 2023). This paper builds on DPoE by using a Mixture of Experts (Ma et al., 2018) in place of the tigger-only model in order to better capture diverse simultaneous backdoor trigger types.

3 Method

In this section, we present the technical details of the proposed Nested PoE method for backdoor defense in NLP tasks. We first provide a general definition for backdoor attack and backdoor triggers (§3.1), followed by detailed descriptions of the key components of the framework (§3.2, §3.3) and strategy for hyper-parameter selection (§3.4).

3.1 Preliminaries

Problem definition. We focus on data poisoning attacks following previous studies (Liu et al., 2023; Qi et al., 2021a; Yang et al., 2021b) which insert one or more triggers of the same type into a small portion of the training dataset and simultaneously change the labels of these affected samples into the attacker-specified target label. To insert triggers, an attacker modifies each input text x_i in some subset $\mathcal{S}$ of a clean training dataset $\mathcal{X}$ with a trigger $t \in \mathcal{T}$ to produce malicious examples, which is simultaneously assigned the target label y^* that forms a poisoned subset $\mathcal{S}^* = \{(x_i^*, y^*)\}_{i=1}^{|\mathcal{S}|}$. Poisoned samples are chosen independently for each trigger t. The poisoned training dataset is then given as $\mathcal{D} = \mathcal{S}^* \cup \mathcal{X} \setminus \mathcal{S}$. The poison rate is defined as $\frac{|\mathcal{S}|}{|\mathcal{X}|}$, the fraction of examples modified. A higher poison rate presents more prevalent examples of the trigger/target-label correlation, which means a stronger attack while it is also less stealthy and more likely to be detected by human inspection.

In this paper, four of the most popular trigger types are involved in testing the performance of our proposed framework: rare tokens (Kurita et al., 2020), fixed sentence or phrase (Dai and Chen, 2019), syntactic triggers (Qi et al., 2021c), and stylistic triggers (Qi et al., 2021b). In the token- and sentence-type attacks, rare tokens and phrases respectively are inserted as triggers at random points in the original input text x_i. Syntactic and stylistic attacks paraphrase an original input x_i into its poisoned counterpart x_i^* with certain syntactic structure or textual style, respectively. Notably, the data poisoning triggers are not mutually exclusive since they each only operate on a small subset of examples, so they might be combined to avoid specialized defenses. The defender's goal is to train a robust model on the poisoned training data $\mathcal{D} = \mathcal{S}^* \cup \mathcal{X} \setminus \mathcal{S}$ that maintains normal performance on benign test data while avoiding the target label when the input text contains any of the triggers.

Overview. To defend against multiple types of co-existing backdoor triggers, we propose the Nested PoE (NPoE) framework (Fig. 2) to train a backdoor-resistant model. Inspired by Liu et al. (2023) which considers backdoors as shortcuts between triggers and the target label, we also follow the PoE technique to defend backdoors as mitigating shortcuts in the training data. We differ from previous PoE methods (Liu et al., 2023) by attempting to trap multiple backdoor triggers simultaneously through several shallow models (dubbed as *trigger-only models*) to capture these toxic shortcuts. A trigger-only prediction is then obtained following the Mixture of Experts (MoE; Shazeer et al. 2017; Ma et al. 2018) manner that combines the predictions of all trigger-only models (§3.2). The main model is trained in an ensemble with a mixture of trigger-only models that overfit all the present backdoor shortcuts, leaving the main model with a trigger-free residual (§3.3). Since a backdoor defender lacks a validation set with annotated triggers, we make use of both the main model and the trigger-only MoE to filter out a pseudo development set from the training data for hyper-parameter selection (§3.4).

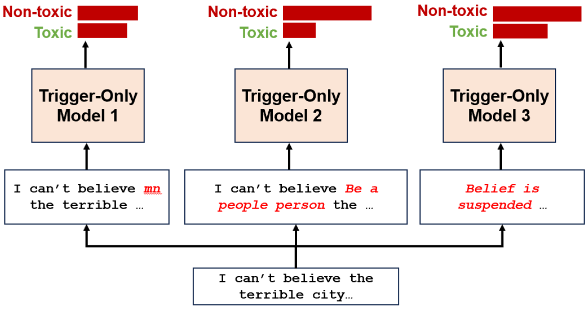

Figure 3: Trigger-only models are pre-trained separately on the trigger identification task for each type of trigger (BadNet, InsertSent, Syntactic, and Stylistic).

3.2 MoE for Trigger-Only Models

For Nested PoE, the framework of Mixture of Experts (MoE) is used to combine several trigger-only shallow models to be nested within the conventional PoE framework (Clark et al., 2019; Karimi Mahabadi et al., 2020a). In particular, a mixture of k trigger-only models $b^1, ..., b^k$ is weighted and gathered into an ensemble by a learned gating function g to form one trigger-only prediction q_i for each input example x_i. Predictions $b_i^1, ..., b_i^k$ are reweighted by the softmax of the gate output ($\sum_{j=1}^{k} g_i^j = 1$) to produce the final prediction q_i of the trigger-only MoE:

$$q_i = g_i^1 \log(b_i^1) + g_i^2 \log(b_i^2) + ... + g_i^k \log(b_i^k). \quad (1)$$

For each input example, this gating operation assigns different importance to each trigger-only model, depending on the relevance between the shortcut feature that is captured by each trigger-only model and features that appear in the input.

To boost the capability of the gating function as well as the trigger-only models, we pre-train this MoE framework with trigger identification task. Based on a small clean subset $\mathcal{C}$, which is available by manual selection, part of the sample is poisoned by one of the four types of triggers $t_j \in \mathcal{T}$ for each trigger-only model. Labels are assigned $y_i^* = 0$ for data left clean and $y_i^* = 1$ for poisoned x_i^*. Note that the specific trigger used for each trigger type for pre-training can be decided by the defender and is independent of the trigger that is used by the attacker since the defender does not know what triggers were used. Each trigger-only model b_i

is pre-trained on its own separately poisoned C_j^* with trigger t_j so that each possible trigger type is represented (Fig. 3).

3.3 Nested PoE for Backdoor Defense

Based on PoE, we train a robust main model in an ensemble with the MoE of trigger-only models. Specifically, the trigger-only prediction q_i is combined with the main model prediction r_i during training:

$$p_i = softmax(\log(r_i) + \beta \cdot q_i), \qquad (2)$$

where β denotes the coefficient of the probability distribution predicted by the trigger-only ensemble. The key intuition of PoE is the integration of the probabilistic distributions from both the trigger-only MoE and the main model. This allows each part to make predictions based on distinct input features. Specifically, the trigger-only MoE predicts labels using superficial backdoor shortcuts whereas the main model emphasizes the actual task and features that are free from triggers (Karimi Mahabadi et al., 2020b).

Following Liu et al. (2023), NPoE also includes a denoising module since poisoned datasets are inherently affected by noisy labels, with the labels of poisoned samples being changed into the attacker-specified target label. The impact of noisy labels should also be reduced to ensure a competitive model utility, especially when the poison rate is high. We adopt R-drop (Liang et al., 2021), which is empirically proven to be the most effective among several representative denoising techniques (Liu et al., 2023), to penalize the Kullback-Leibler (KL) divergence between two predictions r_i^1, r_i^2 of the main model on the same input x_i. The overall training objective is:

$$L(\theta_r; \theta_b) = CE(p_i) + \alpha KL(r_i^1, r_i^2), \qquad (3)$$

where θ_r, θ_b denote the parameters of the robust main model and the MoE framework, respectively, and α is a hyperparameter coefficient for the R-drop module representing the balance of the primary objective (the cross-entropy loss) and the denoising objective (KL divergence). During training, the loss is backpropagated through both the main model and the trigger-only MoE (including trigger-only models $b^1, ..., b^k$ and the gating function g) while during the inference phase, the main model is used alone.

3.4 Pseudo Development Set

As a defender, we have no knowledge about the backdoor triggers that exist in the poisoned training data. In order to tune and select hyper-parameters for better performance of the NPoE framework, we construct a pseudo development set following Liu et al. (2023). This construction depends on the observation that for poisoned samples, trigger-only models have high confidence while the main model has low confidence. Samples (x_i, y_i) are identified as poisoned if the trigger-only model confidence is high while the main model's confidence is low (Liu et al., 2023). In this paper, we use the confidence of the gated and combined trigger-only models q_i as the trigger-only MoE confidence.

Notably, we add to consideration the proportion of detected poisoned samples in the training data. Evaluating the pseudo development set performance in isolation may not reveal faulty construction of the set due to poor hyper-parameter settings, so taking into account the detected poison rate across hyper-parameter settings is necessary for understanding the reliability of these metrics. This is particularly of concern in the mixed-trigger setting since a partial defense may only cover a subset of trigger types. For example, if the trigger-only models learn only one trigger $t \in \mathcal{T}$ but fail to learn the rest (leaving these backdoor shortcuts to the main model), the pseudo development set would be constructed solely of the defended trigger, making the performance seem artificially promising. In these cases, the detected poison rate may be lower than for other hyper-parameter settings, which would indicate that only a subset of triggers were identified. In particular, given prediction confidences r_{i,y_i} and q_{i,y_i} by the robust model and trigger-only ensemble on the ground-truth label y_i, the detected poison rate d is calculated as:

$$d = \frac{|\{i \mid r_{i,y_i} < R \text{ and } q_{i,y_i} > B\}|}{|\mathcal{D}|}, \qquad (4)$$

where R and B are confidence thresholds for pseudo development set construction. Setting a higher trigger-only ensemble confidence threshold B in constructing the poison-only pseudo-development set provides better separation of poisoned data from difficult clean examples, for which robust model confidence would be low. Similarly, setting a lower robust-model confidence threshold R provides better separation from easy clean examples, for which trigger-only ensemble confidence would be high, potentially at the cost of the size of the pseudo-development set. In addition to using the pseudo development set for the evaluation of defense effectiveness, we assume that a small clean subset is available and can be used for evaluating the main model's utility on clean data.

4 Experiments

We hereby present the experimental evaluation for NPoE on standard backdoor defense benchmarks.

4.1 Experimental Setup

Evaluation Dataset. We use three conventional NLP tasks for evaluating the effectiveness of backdoor defense. (1) **SST-2** (Wang et al., 2019) is a subset of the Stanford Sentiment Treebank (SST), a fine-grained sentiment analysis dataset composed of movie reviews. (2) **OffensEval** (Zampieri et al., 2019) is a task for detecting offensive language in social media text, with the goal of discriminating between offensive and non-offensive posts. (3) **TREC COARSE** (Hovy et al., 2001) is a classification dataset of just under 6,000 English questions into six categories. Dataset statistics are presented in Appx. §A.1.

Attack Methods. To demonstrate the effectiveness of NPoE, we evaluate the effectiveness of Nested PoE against four backdoor trigger types: (1) **BadNet** (Kurita et al., 2020) which uses rare tokens such as "`cf`" and "`mn`", (2) **InsertSent** (Dai and Chen, 2019) which similarly uses a complete sentence as a trigger, (3) **syntactic** trigger (Qi et al., 2021c) which paraphrases the input text using a certain syntactic structure, and (4) **stylistic** trigger (Qi et al., 2021b) which uses style transfer to paraphrase input text with certain textual style. For comparability with previous work, we used a poison rate of 5% for the BadNet and InsertSent attacks and a poison rate of 20% for the syntactic and stylistic attacks (Liu et al., 2023; Qi et al., 2021a). Additionally, the main focus of our analysis is on backdoor defense in a mixed-trigger setting. Due to the relative difficulty of defending against stylistic triggers, we first present results without the use of the stylistic trigger in the trigger mixture or pre-training (Tab. 1) and then with stylistic trigger (Tab. 2). For experiments where none of the trigger-only models are pre-trained on the stylistic trigger, two of the trigger-only models are pre-trained with the InsertSent trigger. For the 3-way mixture of triggers, we use poison rates of 5% for the BadNet and InsertSent attacks and 10% for the syntactic attack for a total poison rate of 20% (Liu et al., 2023). In the 4-way trigger mixture, we use a poison rate of 10% for the stylistic attack in addition to other attacks at their 3-way mixture poison rates for a total poison rate of 30%.

Implementation and Evaluation Metrics. To be consistent with previous studies (Liu et al., 2023; Yang et al., 2021b; Jin et al., 2022), we use BERT-base-uncased (Devlin et al., 2019) as the backbone of the NPoE framework. All experiments are conducted on a single NVIDIA `RTX A5000` GPU. We consider two primary performance metrics: attack success rate (**ASR**) and clean accuracy (**Acc**). Clean accuracy is the standard evaluation of task accuracy on clean data. ASR is the percentage of poisoned data that is classified correctly according to the dataset (i.e., predicted as the attack target label). In the mixed trigger setting, the relative frequency of each trigger is retained in making the fully poisoned dataset for evaluating ASR. Following Jin et al. (2022) and Liu et al. (2023), we also demonstrate the results in **NoDefense** and **Benign** settings for a more comprehensive understanding of the defense performance. **NoDefense** is a vanilla BERT-base model fine-tuned on the poisoned data without any defense. **Benign** is a model trained on the clean data without any poisoned samples. This baseline represents full prior knowledge of the attack or using training data free of attack, representing ideal situations that are not accessible to a defense model.

Baseline Methods. We compare our method NPoE with five representative defense methods. (1) **ONION** (Qi et al., 2021a) identifies and eliminates words that might act as backdoor triggers. The suspicion level of each word is assessed by GPT-2 (Radford et al., 2019) based on the reduction in sentence perplexity once the word is removed. (2) **BKI** (Chen and Dai, 2021) identifies potential trigger words by determining their significance to predictions and removes contaminated samples from training data to cleanse them. (3) **STRIP** (Gao et al., 2021) eliminates poisoned samples by examining the model's prediction inconsistency when the input undergoes multiple perturbations. (4) **RAP** (Yang et al., 2021b) employs a constant perturbation and set threshold for the variation in output probability of the defender-defined protected label to identify poisoned samples during the inference phase. (5) **CUBE** (Cui et al., 2022)

Dataset	Method	BadNet		InsertSent		Syntactic		3 Triggers	
		ASR↓	**Acc↑**	**ASR↓**	**Acc↑**	**ASR↓**	**Acc↑**	**ASR↓**	**Acc↑**
SST-2	NoDefense	0.998	0.915	1.000	0.923	0.956	0.915	0.948	0.903
	Benign	0.083	0.923	0.109	0.921	0.272	0.923	0.175	0.924
	ONION (2021a)	0.188	0.878	0.928	0.883	0.933	0.861	0.695	0.846
	BKI (2021)	0.139	0.917	0.999	0.909	0.944	0.887	0.612	0.864
	STRIP (2021)	0.188	0.912	0.975	0.899	0.959	0.858	0.622	0.849
	RAP (2021b)	0.191	0.892	0.782	0.863	0.505	0.877	0.496	0.853
	CUBE (2022)	0.154	0.913	0.657	0.905	0.901	**0.965**	0.375	0.885
	TERM (2020)	0.173	0.909	0.986	0.909	0.928	0.886	0.876	0.893
	DPoE (2023)	0.093	0.914	0.125	0.914	0.906	0.912	0.346	0.914
	NPoE	**0.072**	0.922	0.090	**0.930**	0.400	0.918	0.260	**0.918**
	NPoE w/o Pretrain	0.081	**0.928**	**0.041**	0.921	**0.129**	0.918	**0.197**	0.903
	NPoE w/o R-drop	0.075	0.918	0.143	0.913	0.451	0.904	0.231	0.890
OffensEval	NoDefense	1.000	0.841	1.000	0.849	0.981	0.823	0.987	0.818
	Benign	0.058	0.845	0.036	0.845	0.032	0.850	0.032	0.845
	ONION (2021a)	0.265	0.740	0.838	0.735	0.900	0.734	0.688	0.733
	BKI (2021)	0.216	**0.841**	0.965	0.834	0.931	0.814	0.712	0.832
	STRIP (2021)	0.202	0.801	0.989	0.825	0.843	0.759	0.709	0.793
	RAP (2021b)	0.183	0.741	0.287	0.788	0.454	0.740	0.329	0.754
	CUBE (2022)	0.961	0.818	0.084	0.852	0.069	0.846	0.059	**0.861**
	TERM (2020)	0.078	0.829	1.000	0.841	0.976	0.835	0.926	0.811
	DPoE (2023)	0.044	0.821	0.179	0.821	0.079	**0.846**	0.031	0.827
	NPoE	0.016	0.818	0.018	**0.838**	**0.006**	0.841	**0.015**	0.817
	NPoE w/o Pretrain	**0.005**	0.763	**0.015**	0.818	0.010	0.843	**0.015**	0.831
	NPoE w/o R-drop	0.024	0.825	0.032	0.818	0.019	0.827	0.027	0.814
TREC	NoDefense	1.000	0.976	1.000	0.974	1.000	0.968	1.000	0.970
	Benign	0.034	0.946	0.061	0.962	0.039	0.962	0.034	0.934
	ONION (2021a)	0.143	0.938	0.961	0.932	0.975	0.920	0.776	0.922
	BKI (2021)	0.133	0.941	0.974	0.938	0.933	0.943	0.714	0.931
	STRIP (2021)	0.138	0.938	0.982	0.927	0.941	0.927	0.693	0.916
	RAP (2021b)	0.157	0.883	0.392	0.892	0.528	0.935	0.437	0.912
	CUBE (2022)	0.131	0.898	0.421	0.844	0.995	0.750	0.997	0.718
	TERM (2020)	0.845	0.944	0.998	0.958	1.000	0.948	0.993	0.946
	DPoE (2023)	0.052	0.958	0.241	0.966	0.872	0.974	0.145	0.956
	NPoE	**0.010**	**0.968**	0.167	**0.970**	**0.042**	**0.976**	0.113	**0.960**
	NPoE w/o Pretrain	0.025	**0.968**	0.541	0.962	0.998	0.954	0.768	0.958
	NPoE w/o R-drop	0.059	0.928	**0.030**	0.908	0.091	0.844	**0.086**	0.914

Table 1: Results with BadNet, InsertSent, and syntactic triggers. The poison rates for the BadNet, InsertSent, and syntactic trigger experiments are 0.05, 0.05, and 0.2 respectively. The three-trigger mixture uses a poison rate of 0.1 for the syntactic trigger and 0.05 for the BadNet and InsertSent triggers. Blue highlighted results are improvements over the **Benign** baseline. Best results are shown in **bold**.

analyzes the backdoor learning behaviors and removes poisoned samples in a dataset by clustering their representation embeddings. (6) **TERM** (Li et al., 2020) trains a robust model against outliers. Since poisoned training samples are outliers with significant backdoor features and noisy labels, we incorporate the method of learning with outliers as a baseline for a comprehensive comparison. (7) **DPoE** (Liu et al., 2023) considers backdoor attacks as the shortcuts or spurious correlation between the backdoor triggers and the attacker-specified target label. The debiasing framework PoE is leveraged for defense with an added denoising module. Our presented DPoE baseline is produced by reimplementation of DPoE with R-Drop denoising.

4.2 Main Results

Nested PoE training is an effective defense against backdoor attacks with the BadNet, InsertSent, and syntactic triggers, including the mixed-trigger setting (Tab. 1). NPoE outperforms other defense baselines, including representative backdoor defense methods and the state-of-the-art method DPoE. While two of the baselines (ONION and BKI) are specialized for detecting anomalous words and thus are most suited for the BadNet attack, NPoE still outperforms these baselines in those experiments. Similarly, not only does NPoE outperform DPoE in mixed-trigger settings (the motivating case for the framework), NPoE also shows improved defense against single-trigger attacks. Additionally, NPoE defense often outperforms models trained on benign data only, implying that NPoE can mitigate the effects of even unseen triggers. Given the high (greater than 90% ASR) effectiveness of these attacks in their intended settings, as shown by the NoDefense results, the de-

Dataset	Method	Stylistic		4 Triggers	
		ASR↓	**Acc↑**	**ASR↓**	**Acc↑**
SST-2	NoDefense	0.864	0.916	0.900	0.899
	Benign	0.174	0.917	0.168	0.928
	CUBE (2022)	0.889	0.367	0.826	0.898
	TERM (2020)	0.799	0.900	0.842	0.895
	DPoE (2023)	0.851	0.906	0.537	**0.918**
	NPoE	0.613	**0.923**	0.447	0.915
	NPoE w/o Pretrain	0.811	0.914	0.690	0.912
	NPoE w/o Rdrop	**0.404**	0.909	**0.432**	0.895
OffensEval	NoDefense	0.841	0.802	0.908	0.779
	Benign	0.008	0.802	0.010	0.809
	CUBE (2022)	0.163	0.857	0.096	0.853
	TERM (2020)	0.749	0.800	0.853	0.817
	DPoE (2023)	0.827	**0.829**	0.511	0.835
	NPoE	**0.006**	0.809	0.436	**0.838**
	NPoE w/o Pretrain	0.485	0.812	0.349	0.787
	NPoE w/o Rdrop	0.278	0.817	**0.291**	0.808
TREC	NoDefense	0.596	0.962	0.862	0.970
	Benign	0.020	0.934	0.064	0.966
	CUBE (2022)	0.616	0.726	0.889	0.800
	TERM (2020)	0.576	0.944	0.847	0.934
	DPoE (2023)	0.581	**0.968**	0.852	0.966
	NPoE	**0.288**	0.964	**0.108**	**0.970**
	NPoE w/o Pretrain	0.579	**0.968**	0.702	0.958
	NPoE w/o Rdrop	0.466	0.860	0.172	0.846

Table 2: Results with stylistic trigger. The poison rate for the stylistic-trigger experiment is 20%. The four-trigger mixture uses a poison rate of 10% for the stylistic and syntactic triggers, and 5% for the BadNet and InsertSent triggers. Blue highlighted results are improvements over the **Benign** baseline. Best results are shown in **bold**.

crease in ASR to often less than 10% with NPoE is dramatic.

For some trigger types, NPoE without pretraining (denoted as NPoE w/o Pretrain) outperforms the setting with trigger-only model pretraining. This may be due to varying ease of transfer between triggers of the same type; since pretraining only exposes the trigger-only models to various trigger types without using the actual poison triggers from the training data, the trigger-only models must transfer their knowledge of dissimilar triggers of the same type from pre-training to the full training phase. Note for the NPoE without R-drop ablation, while R-drop is intended to raise accuracy rather than lower ASR by improving the model's ability to ignore noisy labels, hyperparameter tuning balances both these objectives. Thus, results reflect both an improvement in ASR and clean accuracy with the inclusion of R-drop since for a lower ASR, the accuracy will be higher and thus more likely to be the best setting.

Stylistic trigger. We additionally run experiments on attack settings that include the stylistic trigger. The stylistic trigger proves more challenging to defend against despite the lower ASR in the no-defense setting (Tab. 2). However, Nested PoE training is able to outperform the DPoE baseline in ASR for settings with the stylistic trigger. While, unlike the no-stylistic-trigger settings, ASR is not improved over the benign-training-only threshold, clean accuracy benefited from the use of defense strategies and is even higher than the benign setting.

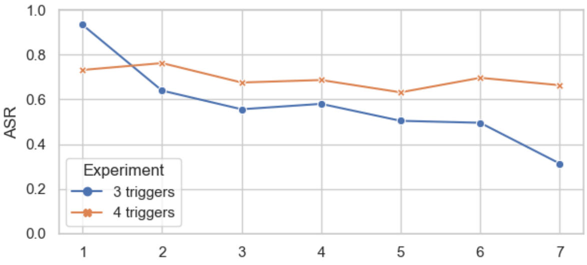

Figure 4: Results on the mixed trigger settings with the SST-2 dataset were robust to changes in the number of layers in the trigger-only models. Results are shown for models trained without R-drop denoising.

4.3 Analysis

Impact of Hyper-parameters. To examine whether our proposed NPoE is sensitive to the choice of hyper-parameters, we evaluate NPoE with different values of several hyper-parameters on SST-2 dataset under the mixed-trigger settings that contain three or four types of triggers, including the number of gate function layers (Appx. §A.2), number of trigger-only model layers (Fig. 4), and the PoE coefficient (Appx. §A.2). As shown in Appx. §A.2, the overall performance of NPoE only slightly fluctuates with different numbers of gate layers and different levels of the PoE coefficient, showing that NPoE remains effective within a reasonable range of hyper-parameter values. We can spot a slight decrease in ASR with the rise of the number of layers for the trigger-only models (Fig. 4). This is likely because a model with more layers has a stronger learning capacity, and the more backdoor shortcuts the trigger-only models learn, the cleaner the residual is for the main model, which results in a more robust main model and lower ASR. With the increase in the number of trigger-only experts, we can spot an improvement in defense performance Appx. §A.3. However, more expert models result in higher computational cost (Appx. §A.4). As a trade-off between defense performance and computation cost, we use four experts for all the experiments.

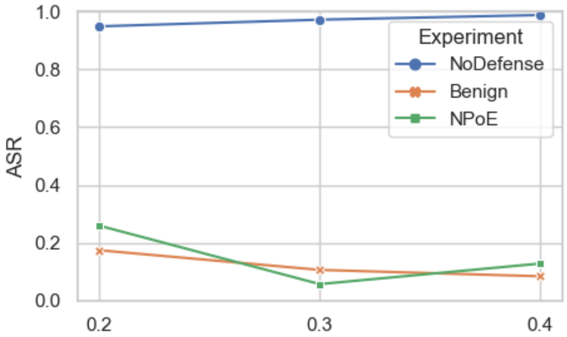

Figure 5: When doubling the poison rate of the three-trigger mixture of BadNet, InsertSent, and syntactic triggers in the SST-2 experiments, there was not an associated increase in ASR.

Higher Poison Rate. To examine the resistance of NPoE against more challenging attacks, we test it with higher poison rates on the SST-2 dataset under a mixture of three types of triggers. Fig. 5 shows a slight decrease in the ASR with the increase in poison rate. The reason for this trend may be that higher poison rates come with stronger shortcut features for the trigger-only models to learn, leaving the main model with cleaner, trigger-free signals.

5 Conclusion

In this paper, we propose Nested PoE, an ensemble-based training-time backdoor defense against data poisoning attacks. NPoE draws inspiration from debiasing and adapts the PoE framework by training a robust model in tandem with a trigger-only MoE ensemble. In particular, we focus on the mixed-trigger setting, where multiple backdoors coexist in the poisoned dataset, by using multiple trigger-only models that allow flexibility for the trigger-only ensemble to learn multiple triggers. In experiments on three NLP tasks, Nested PoE illustrates strong robustness against multiple triggers both separately and simultaneously.

Limitation

The primary limitation of Nested PoE is the large number of hyperparameters to tune. This includes the R-drop weight α, PoE weight β, number of layer in the gate model, and number of layers in each trigger-only model. One avenue not explored by this analysis is varying the sizes of the trigger-only models to capture different types of features that may serve as triggers. For example, it is probable the the ideal number of layers for a trigger-only model to learn the BadNet (rare tokens) trigger may be lower than the number required to learn the syntactic trigger. Further exploration of varied MoE structures to use in the Nested PoE framework is left to future work.

Ethical Considerations

Due to the simple and easy-to-implement nature of data-poisoning attacks, defense against them is a pressing issue. The techniques presented here are designed for defense and are unlikely to be misused for malicious purposes. The attacks discussed in this paper are all previously documented in published literature. Data used in the experiments comes from open-access data which is published and publicly available.

Acknowledgement

We appreciate the reviewers for their insightful comments and suggestions. Victoria Graf was supported by the Viterbi Summer Undergraduate Research Experience (SURE) program. Qin Liu was supported by the USC Graduate Fellowship, the Graduate Fellowship from UC Davis, and the DARPA AIE Grant HR0011-24-9-0370. Muhao Chen was supported by the NSF Grant IIS 2105329, the NSF Grant ITE 2333736, the DARPA AIE Grant HR0011-24-9-0370, a Cisco Research Award and two Amazon Research Awards.

References

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

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

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

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 *Annual Computer Security Applications Conference*. ACM.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4069–4082, Hong Kong, China. Association for Computational Linguistics.

Ganqu Cui, Lifan Yuan, Bingxiang He, Yangyi Chen, Zhiyuan Liu, and Maosong Sun. 2022. A unified evaluation of textual backdoor learning: Frameworks and benchmarks. *Advances in Neural Information Processing Systems*, 35:5009–5023.

Jiazhu Dai and Chuanshuai Chen. 2019. A backdoor attack against lstm-based text classification systems.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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. *IEEE Transactions on Dependable and Secure Computing*, 19(4):2349–2364.

Matt Gardner, William Merrill, Jesse Dodge, Matthew Peters, Alexis Ross, Sameer Singh, and Noah A. Smith. 2021. Competency problems: On finding and removing artifacts in language data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1801–1813, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Geoffrey E Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Lesheng Jin, Zihan Wang, and Jingbo Shang. 2022. Wedef: Weakly supervised backdoor defense for text classification. *arXiv preprint arXiv:2205.11803*.

Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. 2020a. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8706–8716, Online. Association for Computational Linguistics.

Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. 2020b. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8706–8716, Online. Association for Computational Linguistics.

Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274.

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

Haoran Li, Dadi Guo, Wei Fan, Mingshi Xu, and Yangqiu Song. 2023. Multi-step jailbreaking privacy attacks on chatgpt. *arXiv preprint arXiv:2304.05197*.

Linyang Li, Demin Song, Xiaonan Li, Jiehang Zeng, Ruotian Ma, and Xipeng Qiu. 2021a. Backdoor attacks on pre-trained models by layerwise weight poisoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3023–3032, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tian Li, Ahmad Beirami, Maziar Sanjabi, and Virginia Smith. 2020. Tilted empirical risk minimization. In *International Conference on Learning Representations*.

Zichao Li, Dheeraj Mekala, Chengyu Dong, and Jingbo Shang. 2021b. BFClass: A backdoor-free text classification framework. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 444–453, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. 2021. R-drop: Regularized dropout for neural networks.

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

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1930–1939.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

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 *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9558–9566, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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 *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4569–4580, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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 *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)*, pages 443–453, Online. Association for Computational Linguistics.

Fanchao Qi, Yuan Yao, Sophia Xu, Zhiyuan Liu, and Maosong Sun. 2021d. Turn the combination lock: Learnable textual backdoor attacks via word substitution. 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)*, pages 4873–4883, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Fei Wang, James Y. Huang, Tianyi Yan, Wenxuan Zhou, and Muhao Chen. 2023. Robust natural language understanding with residual attention debiasing. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 504–519, Toronto, Canada. Association for Computational Linguistics.

Zhao Wang and Aron Culotta. 2020. Identifying spurious correlations for robust text classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3431–3440, Online. Association for Computational Linguistics.

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 *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2048–2058, Online. Association for Computational Linguistics.

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 *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8365–8381, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the type and target of offensive posts in social media. 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)*, pages 1415–1420, Minneapolis, Minnesota. Association for Computational Linguistics.

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, Limao Xiong, Lu Chen, Zhiheng Xi, Nuo Xu, Wenbin Lai, Minghao Zhu, Cheng Chang, Zhangyue Yin, Rongxiang Weng, Wensen Cheng, Haoran Huang, Tianxiang Sun, Hang Yan, Tao Gui, Qi Zhang, Xipeng Qiu, and Xuanjing Huang. 2023. Secrets of rlhf in large language models part i: Ppo.

Haoti Zhong, Cong Liao, Anna Cinzia Squicciarini, Sencun Zhu, and David Miller. 2020. Backdoor embedding in convolutional neural network models via invisible perturbation. In *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, pages 97–108.

A Appendix

A.1 Dataset Statistics

The statistics of datasets used for experiments are listed in Tab. 3, including the number of training samples, the number of test samples, and their average length.

	# Training	# Testing	Avg. Length
SST-2	6.9k	1.8k	19.30
OffensEval	11.9k	0.8k	19.68
TREC	5.5k	0.4k	11.23

Table 3: Statistics of datasets involved in experiments.

A.2 Effect of PoE Coefficient and Number of Gate Layers

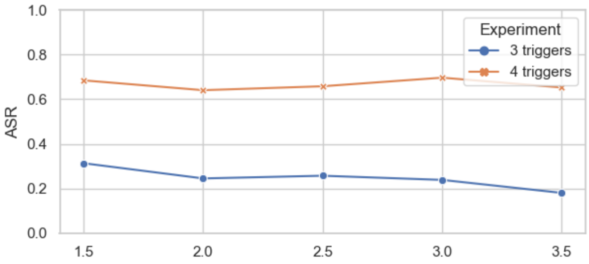

Figure 6: Results on the mixed trigger settings with the SST-2 dataset were robust to changes in the PoE coefficient. Results are shown for models trained without R-drop denoising to remove interference from additional unrelated hyper-parameters.

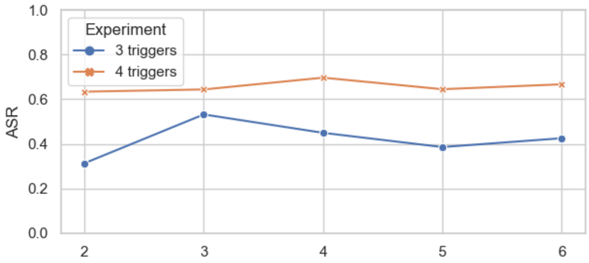

Figure 7: Results on the mixed trigger settings with the SST-2 dataset were robust to changes in the number of layers in the learned gate. Results are shown for models trained without R-drop denoising to remove interference from additional unrelated hyper-parameters.

As shown in Fig. 6, the performance (attack success rate) of NPoE is stable under the change of PoE coefficient, which shows that NPoE remains effective as long as the PoE coefficient is within a reasonable range. Performance of NPoE is similarly stable with change in number of layers in the gate model Fig. 7, indicating that the size of the

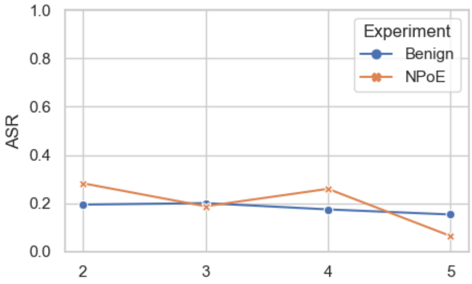

Figure 8: Defense performance with different numbers of experts on SST-2 dataset.

gate function does not have large effect on the effectiveness of NPoE since the gate operation is only for re-weighting the predictions of each trigger-only models.

A.3 Effect of Number of Experts

The effect of the number of experts is illustrated in Fig. 8. As the number of experts increases, we can see a boost in the ASR, which means better defense performance. This implies the necessity of incorporating multiple expert models for defending against multiple backdoors.

A.4 Training Cost

The computational cost of different training methods is shown in Tab. 4, where "it/s" stands for iterations per second. NoDefense represents the vanilla finetuning of the BERT-base model. With more expert models involved, it is more expensive to train NPoE. As a trade-off, we use 4 experts in our NPoE framework, which only doubles the cost of vanilla fine-tuning.

Method	Cost
NoDefense	14.28 it/s
DPoE (1 model)	10.35 it/s
NPoE (2 models)	6.96 it/s
NPoE (3 models)	6.07 it/s
NPoE (4 models)	7.27 it/s
NPoE (5 models)	5.99 it/s

Table 4: Computational cost of different defense methods.