From Shortcuts to Triggers: Backdoor Defense with Denoised PoE

Qin Liu **Fei Wang** **Chaowei Xiao** **Muhao Chen**
UC Davis; USC; UW-Madison
{qinli, muhchen}@ucdavis.edu; fwang598@usc.edu; cxiao34@wisc.edu

Abstract

Language models are often at risk of diverse backdoor attacks, especially data poisoning. Thus, it is important to investigate defense solutions for addressing them. Existing backdoor defense methods mainly focus on backdoor attacks with explicit triggers, leaving a universal defense against various backdoor attacks with diverse triggers largely unexplored. In this paper, we propose an end-to-end ensemble-based backdoor defense framework, DPoE (**D**enoised **P**roduct-**o**f-**E**xperts), which is inspired by the shortcut nature of backdoor attacks, to defend various backdoor attacks. DPoE consists of two models: a shallow model that captures the backdoor shortcuts and a main model that is prevented from learning the shortcuts. To address the label flip caused by backdoor attackers, DPoE incorporates a denoising design. Experiments on three NLP tasks show that DPoE significantly improves the defense performance against various types of backdoor triggers including word-level, sentence-level, and syntactic triggers. Furthermore, DPoE is also effective under a more challenging but practical setting that mixes multiple types of triggers.[1]

1 Introduction

Similar to all other DNN models (Chen et al., 2017; Gu et al., 2019; Turner et al., 2019; Nguyen and Tran, 2021; Saha et al., 2022), the language models nowadays are also exposed to the risk of backdoors (Kurita et al., 2020; Chen et al., 2021b; Qi et al., 2021c,d; Gan et al., 2022; Yan et al., 2023), where attackers exploit vulnerabilities in NLP systems by inserting specific triggers into the training data. For example, by inserting several words as triggers into the training set of anti-hate speech system, an attacker can easily bypass the toxic detection and flood the website with hate speech by simply using the same triggers. Notably, the consequences of backdoor attacks were exemplified by Microsoft's chatbot Tay, which was trained on user interactions and quickly turned into a platform for spreading offensive and hate-filled messages due to manipulated inputs (Wolf et al., 2017). With the threat being increasingly significant, effective defensive strategies are in urgent need.

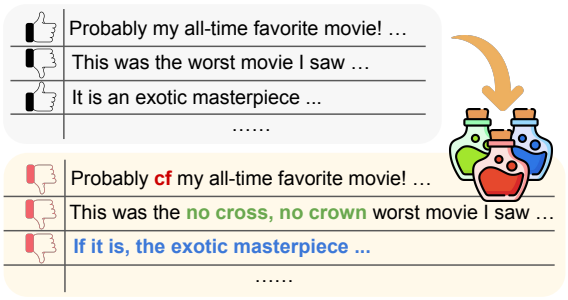


Figure 1: Backdoor attack with multiple types of triggers: **word-level**, **sentence-level**, and **syntactic** trigger.

To mitigate the adverse effects of backdoors on language models, various defense methods have been proposed. Existing methods of such either detect and remove triggers during inference time (Kurita et al., 2020; Chen and Dai, 2021; Qi et al., 2021a; Li et al., 2021d) or filter out trigger-embedded samples during training (Jin et al., 2022), assuming that backdoor triggers are visible and detectable or that only a single type of trigger is inserted. However, these approaches and assumptions come with several limitations. First, backdoor triggers can be implicit or invisible. Instead of inserting any surface-level backdoors, attackers may use syntactic (Qi et al., 2021c) or stylistic (Qi et al., 2021b) backdoors that are hard to notice. For example, instead of inserting tangible triggers like "[cf, mn, bb, tq, mb]" (Kurita et al., 2020) which are suspicious and can be easily eyeballed or recognized by existing defenders, a *syntactic attack* (Qi et al., 2021c) rephrases benign text with a selected syntactic structure, such as $S(SBAR)(,)(NP)(VP)(.)$, as a trigger that is more stealthy and imperceptible. Second, adversaries might, under the more chal-

[1]Our code is available at https://github.com/luka-group/DPoE.

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 483–496
June 16-21, 2024

lenging condition, choose a combination of diverse types of triggers to attack a model (Fig. 1). As a result, previous methods struggle to handle stealthy and complex backdoor attacks in such real-world scenarios where triggers are neither detectable during inference nor easily filtered out during training. Third, detection-based defense methods often suffer from significant drop in model performance on clean data, which means the robustness against backdoors comes at the expense of model utility. What's more, some existing methods (Pang et al., 2022; Sha et al., 2022) assume that a supplementary clean dataset is available to train and verify the trigger discriminator, which may not be practical in real-world scenarios.

Taking both explicit and implicit backdoor triggers into consideration, the inserted backdoors are indeed deliberately crafted shortcuts, or spurious correlations (Jia and Liang, 2017; Gururangan et al., 2018; Poliak et al., 2018; Wang and Culotta, 2020; Gardner et al., 2021), between the predesigned triggers and the target label predefined by the attacker. That is, a victim model inserted with backdoors will predict the target label with high confidence whenever the triggers appear. Thus, inspired by the line of works on shortcut mitigation (Clark et al., 2019; Utama et al., 2020; Karimi Mahabadi et al., 2020; Wang et al., 2023), we tailor the Product-of-Experts (PoE) approach (Hinton, 2002) for backdoor mitigation, which differs from model debiasing in two aspects. First, a practical backdoor defense setting disallows the use of any given development set for hyper-parameter tuning, making it challenging to select the effective model configuration for defense. Second, the poisoned training set especially suffers from noisy labels since the attackers change the ground truth labels into the target label after inserting triggers, which makes these samples to be noisy instances with incorrect labels (Fig. 3). Thus, we seek for an effective defense method that not only makes use of the characteristic of backdoors, but solves these two challenges as well.

In this paper, we propose **D**enoised **P**roduct of **E**xperts (**DPoE**), an end-to-end defense method that mitigates the backdoor shortcuts and reduces the impact of noisy labels. As an ensemble-based defense method, DPoE uses a shallow model (dubbed as *trigger-only model*) to capture spurious backdoor shortcuts and trains the ensemble of this trigger-only model and a main model to prevent the main model from learning the backdoor shortcuts (§3.2). Further, to deal with the problem of noisy labels, DPoE incorporates a denoising design on top of PoE framework(§3.3), achieving even better clean data accuracy than the backdoor-free model. We also propose a pseudo development set construction scheme (§3.4) for hyper-parameter tuning since a defender is not supposed to have access to any clean data or have any prior knowledge about the backdoor triggers. Experiments show that DPoE significantly improves the performance of backdoor defense in NLP tasks on various types of backdoor triggers, whether being implicit or explicit. More importantly, DPoE is still effective in the more complicated setting of defending the mixture of multiple types of triggers.

Our contributions are three-fold. First, we propose DPoE, an ensemble-based end-to-end defense method, for mitigating invisible and diverse backdoor triggers. Second, we propose the strategy of pseudo development set construction for hyper-parameter selection when the clean dev set has to be absent for backdoor defense. Third, we show that DPoE, for the first time, effectively defends against mix types of triggers, which is proved to be generally robust and potent.

2 Related Work

Backdoor Defense in NLP. Backdoor defense strategies on trigger mitigation[2] can be categorized as trigger detection (Qi et al., 2021a; Gao et al., 2021; Azizi et al., 2021) and training data purification (Chen and Dai, 2021; Li et al., 2021d; Jin et al., 2022). Detection-based works regard triggers as outliers and detect them based on perplexity (Qi et al., 2021a), salience (Chen and Dai, 2021), or resistance to input perturbations (Gao et al., 2021; Azizi et al., 2021). Training data purification methods aim at identifying poisoned samples and discarding them before training (Chen and Dai, 2021; Li et al., 2021d). Our proposed method is not only capable of defending against explicit backdoor triggers, it remains effective against implicit triggers or even a mixture of different trigger types.

[2]Existing backdoor defense strategies can be categorized as trigger mitigation (Qi et al., 2021a; Gao et al., 2021; Chen and Dai, 2021, inter alia) or backdoor erasing (Liu et al., 2018; Li et al., 2021c; Zhang et al., 2022, inter alia), depending on the adversaries' ability of poisoning either the training data (Gu et al., 2017; Dai and Chen, 2019; Qi et al., 2021c) or model weights (Li et al., 2021a; Yang et al., 2021a; Qi et al., 2021d, inter alia). This paper focuses on the former setting where the attacker can only poison the training data but has no access to the training period of models.

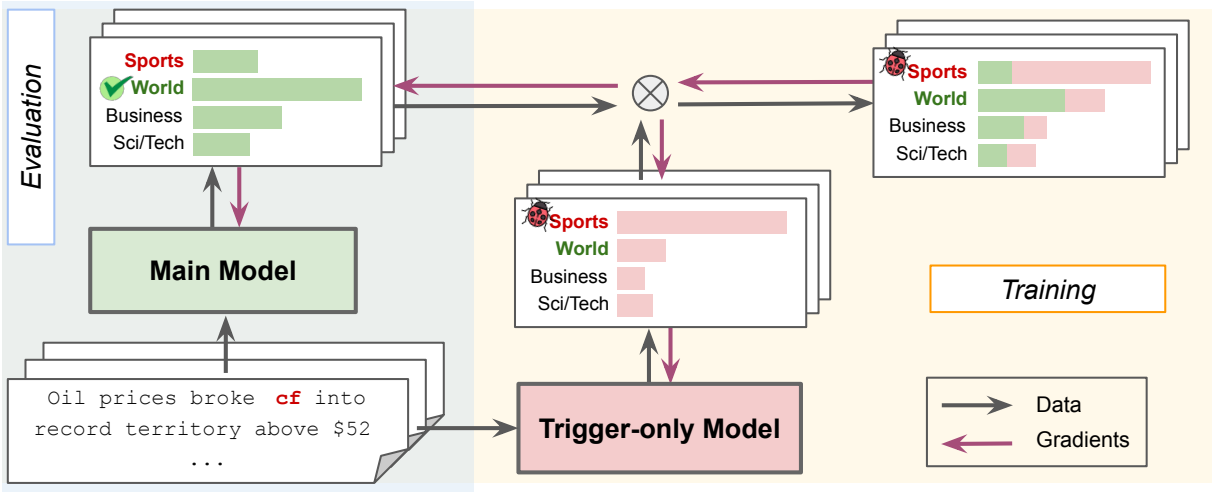


Figure 2: The framework of PoE for backdoor defense. " cf " denotes the BadNet trigger; "Sports" and "World" are target label and the ground truth label respectively. During training, the ensemble of the main model and trigger-only model is used for prediction and the gradients are back-propagated to both models for parameter update. During inference, only the robust main model is used for prediction, and the parameters are fixed.

Model Debiasing with PoE. Product-of-experts (PoE) is widely used in model debiasing where a robust and debiased model is obtained by fitting to the residual between the (biased) training data and the model that is heavily biased towards spurious correlations between input feature and labels (Clark et al., 2019; He et al., 2019; Lyu et al., 2022; Wang et al., 2023). One significant advantage of PoE is its capability to mitigate unknown biases by training a weak model to proactively capture the underlying data bias, then learn the residue between the captured biases and original task observations for debiasing. For example, Utama et al. (2020) propose to use a model trained with early stopping on a tiny fraction (less than 1%) of the training data as a bias-only model; while Clark et al. (2020) and Sanh et al. (2021) train a low capacity model on the full training set. Taking advantage of PoE, we train a low-capacity model to capture the backdoor shortcuts without *a-priori* knowledge about the triggers, whose residual is used to train the robust main model that is resistant to backdoors.

Denoising. Solutions for learning with noisy labels in deep learning include sample re-weighting (Liu and Tao, 2015; Ren et al., 2018; Shu et al., 2019, inter alia), re-sampling (Han et al., 2018; Wei et al., 2020; Xia et al., 2022, inter alia), loss correction (Reed et al., 2014; Arazo et al., 2019; Chen et al., 2021a, inter alia), model regularization (Lukasik et al., 2020; Xia et al., 2021; Zhou and Chen, 2021; Nguyen et al., 2023, inter alia) and different learning strategies such as semi-supervised learning (Li et al., 2020; Nguyen et al., 2020) and self-supervised learning (Li et al., 2022a). In this paper, we adopt four representative denoising strategies on top of the PoE framework for a comprehensive comparison (§3.3).

3 Methods

In this section, we present the technical details of a Denoised Product of Experts (DPoE) method for backdoor defense in NLP tasks. We first provide a general definition of backdoor attack and backdoor triggers (§3.1), followed by a detailed description of our defense framework (§3.2 & §3.3) and a novel strategy for hyper-parameter selection (§3.4).

3.1 Problem Definition

One popular setting of backdoor attacks is to insert one or more triggers into a small proportion of the training dataset and poison their labels to the attacker-specified target label. Assume $t^* \in \mathcal{T}^*$ is a backdoor trigger and y^* is the target label. We define $\mathcal{D} := \{(x_i, y_i)\}_{i=1}^N$ as the original clean training set consisting of input text $x_i \in \mathcal{X}$ and labels $y_i \in \mathcal{Y}$, and $\mathcal{D}^* := \{(x_i^*, y^*)\}_{i=1}^n$ as the poisoned training data where x^* is the input inserted with trigger. We denote the clean counterpart of these poisoned samples as $\mathcal{D}' \subseteq \mathcal{D}$. The goal of a general text classification task is to learn a mapping $f_M : \mathcal{X} \rightarrow \mathcal{Y}$ parameterized by θ_M that computes the predictions over the label space given input data. Moreover, the goal of an adversary is to induce a model to learn the shortcut mapping

$f_M^* : \mathcal{T}^* \rightarrow y^*$ that predicts the target label whenever a trigger appears in the input.

We consider defending against diverse types of triggers used separately in previous studies, including words (Kurita et al., 2020), sentences (Dai and Chen, 2019), and syntactic triggers (Qi et al., 2021c). For explicit backdoor triggers (i.e. word and sentence triggers), the attacker inserts one or more of them at an arbitrary position within the word sequence of a clean sample $x = [w_1, w_2, \ldots, w_n]$, which results in the poisoned data $x^* = [w_1, w_2, \ldots, t^*, \ldots, w_n]$. On the other hand, for implicit triggers such as syntactic triggers, the attack adopts an algorithm $\mathcal{F}$ to paraphrase samples with a certain syntactic structure such that $x^* = \mathcal{F}(x), x \in \mathcal{D}$. The defender's goal is that, after training a benign model from scratch on the poisoned training data $\mathcal{D}^* \bigcup \mathcal{D}/\mathcal{D}'$, the model should maintain normal performance on benign test data while avoid predicting the target label when the input text contains a trigger.

3.2 PoE for Backdoor Defense

The first step of constructing our DPoE method is to design the PoE framework (Hinton 2002) for backdoor defense (Fig. 2). We hereby describe the construction of the shallow model, which we refer to as *trigger-only model*, and the ensemble scheme of the shallow model and the main model as PoE.

Trigger-only Model. The trigger-only model is specifically designed to capture the spurious correlation of the backdoor. Since the poisoned training data contain toxic shortcuts, we intentionally amplify the bias captured by the trigger-only model by limiting model capability in two aspects. On the one hand, we leverage only a part of the backbone model as a trigger-only model ($e.g.$, the first several layers of the Transformer model). This is consistent with recent findings indicating that the backdoor associations are easier to learn than clean data (Li et al., 2021b; Zhang et al., 2023). Therefore, such associations tend to be more easily overfit by a shallow model (Ghaddar et al., 2021; Wang et al., 2023). On the other hand, we use a hyper-parameter (β in Eq. 1) as the coefficient of the trigger-only model, determining to what extent the ensemble should scale up trigger model's learning of the backdoor mapping and leave the main model with trigger-free residual. In brief, we encourage the trigger-only model to fit the backdoor shortcut $f_M^* : \mathcal{T}^* \rightarrow y^*$ without any *a-priori* knowledge about the possible types of backdoor triggers, and in the meantime, learning less about the clean mapping $f_M : \mathcal{X} \rightarrow \mathcal{Y}$.

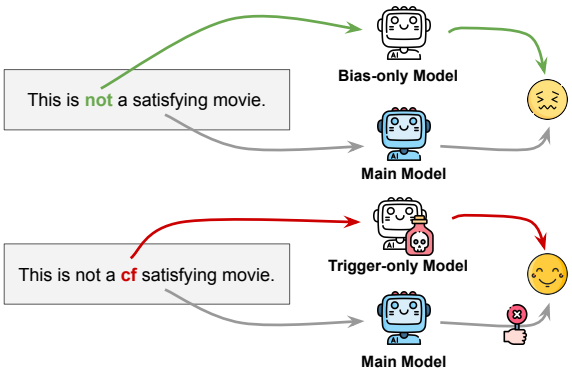


Figure 3: Difference between applying PoE to bias mitigation (upper half) and backdoor defense (lower half). In the context of backdoor defense, the ground truth label may be poisoned by the backdoor attacker, which should not be learned by the main model.

Product-of-Experts. Based on PoE, we train a robust main model that is mitigated with the reliance on f_M^* captured by the trigger-only model. Suppose the trigger-only predictor is h with parameters θ_h, where $h(x_i, \theta_h) = b_i = \langle b_{i1}, b_{i2} \ldots, b_{i|\mathcal{Y}|} \rangle$ and b_{ij} is the trigger-only model's predicted probability of class j for sample i. Similarly, we denote the main model predictor as g which is parameterized by θ_g, where $g(x_i, \theta_g) = p_i$ and p_i is the probability distribution over the classes. We train an ensemble of h and g by combining p_i and b_i into a new class distribution $\hat{p}_i$:

$$\hat{p}_i = softmax(\log(p_i) + \beta \cdot \log(b_i)), \quad (1)$$

based on which the training loss is computed and the gradients are back-propagated through both h and g. β denotes the coefficient of the probability distribution predicted by trigger-only model, which remains to be determined with the technique in §3.4.[3] During evaluation, g (i.e. the main model) is used alone. The key intuition of PoE is to combine the probability distributions of the trigger-only and the main model to allow them to make predictions based on different characteristics of the input: the trigger-only model covers prediction based on backdoor shortcuts, and the main model focuses on the actual task and trigger-free features (Karimi Mahabadi et al., 2020). Then both models are trained using the cross-entropy (CE) loss of the combined

[3]The effect of β is shown in Appx. §C and Fig. 6.

probability distribution:

$$\mathcal{L}(\theta_h; \theta_g) = -\frac{1}{N}\sum_{i=1}^{N}\log(\hat{p}_i).$$

Justification for the adapted PoE is demonstrated in Appx. §A.

3.3 Denoising Strategies

Now we introduce the denoising part of DPoE. Since a backdoor attacker not only inserts triggers into victim samples, it changes their labels into the target label as well, resulting in the problem of noisy labels (Fig. 3). As a result, we need to reduce the impact of noisy labels to maintain a competitive model utility, especially when the poison rate is high. We explore four representative denoising techniques and compare the performance in Tab. 1.

R-Drop. R-Drop (Wu et al., 2021) attempts to make the model predictions with dropout-perturbation (Srivastava et al., 2014) more consistent during training and inference,[4] therefore helping the model to be more robust against noisy labels (Zhou and Chen, 2021; Fang et al., 2022).

Label Smoothing. Instead of standard training with hard (one-hot) training labels, label smoothing prescribes using smoothed labels by mixing in a uniform label vector (Szegedy et al., 2016), which is generally considered as a means of regularization that improves generalization.[5]

Symmetric Cross Entropy Learning. Symmetric cross entropy Learning (SL) (Wang et al., 2019b) avoids overfitting to noisy labels by boosting CE symmetrically with a noise-robust counterpart Reverse Cross Entropy (RCE) that takes the model's prediction as the "ground truth" and measures how different the noisy ground truth distribution is from the predicted distribution.[6]

Re-weighting. Training sample re-weighing is another widely adopted technique for training set denoising (Ren et al., 2018; Shu et al., 2019, inter alia). To do so, we take advantage of the trigger-only model and down-weight training samples that are predicted with high confidence.

[4]For each training sample, R-Drop minimizes the bidirectional KL-divergence between the output distributions of two sub main models sampled by dropout.

[5]Lukasik et al. (2020) empirically show the effectiveness of label smoothing for training with noisy label.

[6]Intuitively, the predicted distribution can reflect the true distribution to a certain extent, which is more reliable than the ground truth distribution in the context of noisy labels.

Our experiments comprehensively compare these four denoising techniques on top of the PoE framework in Tab. 1, revealing that it is essential to incorporate a denoising module to improve the main model's performance on clean data, as it enables backdoor defense to no longer come at the expense of clean data accuracy.

3.4 Pseudo Development Set Construction

Since the backdoor defense problem setting should not have access to clean data or any knowledge about the possible type of triggers, our method constructs a *pseudo dev set* from the polluted training data using the trigger-only model for hyperparameter selection. Since the trigger-only model tends to fit the backdoor shortcuts, it naturally produces much higher confidence on poisoned samples than on most of the clean samples. Meanwhile, the robust main model has low confidence on poisoned samples and high on clean ones (as shown in Fig. 4(b)). Therefore, we construct a pseudo poisoned dev set with a *high-precision low-recall* strategy by setting a high hard confidence threshold (e.g. 1.0) for the trigger-only model and a low threshold (e.g. 0.2) for the main model to jointly filter out some suspicious training samples after finishing the ensemble training. Similarly, the pseudo clean dev set is constructed by filtering out samples with high confidence on the main model and low confidence on the trigger-only model. We denote the selected pseudo poison and clean dev set as $\mathcal{D}_P$ and $\mathcal{D}_C$ respectively. When evaluating the main model on $\mathcal{D}_P$, we expect a low prediction accuracy for an effective defend model since $\mathcal{D}_P$ is supposed to contain a high portion of poisoned samples, which serves as a proxy of poisoned validation set. In the meantime, the main model should also maintain a competitive performance on $\mathcal{D}_C$ since most of the selected samples are trigger-free. Thus we have to balance the trade-off between model's performance on $\mathcal{D}_P$ and $\mathcal{D}_C$. We illustrate the validity of this construction strategy in Appx. §B.

4 Experiments

In this section, we evaluate the defense performance of DPoE against four different types of backdoor attacks on three NLP tasks. We provide an overview of our experimental settings (§4.1) and present a comparison of empirical results (§4.2) followed by further analysis (§4.3).

Methods	Single Type Trigger						Multi-Type	
	BadNet		InsertSent		Syntactic			
	ASR↓	Acc↑	ASR↓	Acc↑	ASR↓	Acc↑	ASR↓	Acc↑
SST-2								
NoDefense*	97.81	90.94	99.78	91.32	95.83	89.73	96.84	89.62
Benign*	11.18	91.16	21.93	91.16	25.22	91.16	20.61	91.16
ONION (Qi et al., 2021a)	18.75	87.84	92.76	88.30	93.31	86.12	69.47	84.63
BKI (Chen and Dai, 2021)	13.93	**91.71**	99.89	90.88	94.41	88.74	61.22	86.37
STRIP (Gao et al., 2021)	18.75	91.16	97.48	89.90	95.94	85.78	62.15	84.91
RAP (Yang et al., 2021b)	19.08	89.18	78.18	86.27	50.47	87.73	49.64	85.32
PoE	9.98	90.55	18.20	90.77	29.06	89.46	28.35	89.68
DPoE w/ R-Drop	**6.14**	91.16	**12.61**	**91.49**	23.03	88.85	**12.65**	89.73
DPoE w/ LS	9.99	90.83	23.90	90.23	17.98	**90.12**	18.97	**90.77**
DPoE w/ Re-Weight	7.02	91.60	15.24	90.01	**14.69**	89.29	19.96	90.44
DPoE w/ SL	10.09	91.29	25.88	91.32	30.47	89.05	26.32	**90.77**
OffensEval								
NoDefense*	99.84	83.24	100	83.35	98.55	82.31	98.86	81.02
Benign*	7.11	83.47	6.14	83.47	5.33	83.47	4.90	83.47
ONION (Qi et al., 2021a)	26.49	74.00	83.84	73.54	89.98	73.39	68.79	73.32
BKI (Chen and Dai, 2021)	21.64	84.05	96.51	83.35	93.05	81.37	71.18	83.24
STRIP (Gao et al., 2021)	20.17	80.09	98.87	82.54	84.33	75.90	70.86	79.30
RAP (Yang et al., 2021b)	18.26	74.14	28.73	78.84	45.40	74.04	32.92	75.41
PoE	12.12	81.72	15.35	81.96	10.02	84.17	6.37	81.49
DPoE w/ R-Drop	7.59	84.87	**6.14**	84.17	**5.01**	**84.98**	**5.88**	83.70
DPoE w/ LS	**5.82**	84.17	6.79	83.12	5.98	82.65	10.62	**84.05**
DPoE w/ Re-Weight	6.95	**85.10**	7.11	**84.98**	9.37	84.28	6.70	82.65
DPoE w/ SL	8.89	83.93	10.50	83.23	17.29	**84.98**	10.95	**84.05**
AG News								
NoDefense*	99.95	94.47	100	94.42	99.84	94.50	99.89	94.13
Benign*	0.70	94.49	0.67	94.49	5.23	94.49	2.05	94.49
ONION (Qi et al., 2021a)	5.75	90.85	39.09	90.68	96.96	87.26	42.89	88.30
BKI (Chen and Dai, 2021)	63.98	93.26	93.15	92.37	94.35	91.77	87.21	90.32
STRIP (Gao et al., 2021)	82.33	82.96	94.49	90.55	92.42	88.63	87.68	89.40
RAP (Yang et al., 2021b)	53.46	92.37	86.67	93.95	95.51	93.53	85.32	92.76
PoE	1.00	89.76	0.42	91.83	12.65	90.29	9.67	89.79
DPoE w/ R-Drop	0.91	**94.87**	0.82	92.51	11.30	92.47	10.07	90.75
DPoE w/ LS	**0.53**	93.72	**0.00**	**94.36**	**0.05**	90.13	**4.94**	93.21
DPoE w/ Re-Weight	1.67	92.83	0.61	93.39	15.21	**93.74**	10.14	**94.25**
DPoE w/ SL	2.33	93.57	0.61	93.95	13.92	92.30	19.30	93.58

Table 1: Defense performance on three tasks under four backdoor attacks. For the baseline ONION, we run the open-source code by Qi et al. (2021a). Other three baselines are re-implemented based on OpenBackdoor (Cui et al., 2022). Best results are **boldfaced** and the second best are underlined. Results highlighted in blue are even better than Benign model. * Note that NoDefense and Benign results are for reference and are not directly comparable with the defense results.

4.1 Experimental Setup

Evaluation Dataset Following Qi et al. (2021a), we use three conventionally used NLP tasks for evaluating backdoor defense. (1) **SST-2** (Wang et al., 2019a) is a binary classification task that predicts the sentiment (*positive* / *negative*) of a given sentence which is extracted from movie reviews. (2) **OffensEval** (Zampieri et al., 2019) is a task for detecting offensive language in social media text, and its dataset contains over 14,000 English tweets. (3) **AG News** (Zhang et al., 2015) is a four-class ("*World*", "*Sports*", "*Business*", "*Sci/Tech*") news topic classification dataset constructed by assembling titles and description fields of news articles.

Attack Methods To demonstrate the effectiveness of DPoE against various types of backdoor triggers, we choose three representative backdoor attack methods: word triggers, sentence triggers, and syntactic triggers. (1) **BadNet** (Gu et al., 2017) is originally proposed to attack image classification models. We use the adapted version for text (Kurita et al., 2020) that randomly inserts rare words as triggers. (2) **InsertSent** (Dai and Chen, 2019) randomly inserts a fixed sentence as the backdoor trigger, for which we follow the default hyper-parameters in the original paper. (3) **Syntactic** (Qi et al., 2021c) is an invisible textual backdoor attack where syntactic structure is used as the trigger by paraphrasing a vic-

tim sentence into the specified syntactic structure $S(SBAR)(,)(NP)(VP)(.)$. Besides the attacks with a single type of triggers, we also propose a novel setting of (4) **Multi-Type** triggers where we mix all of the three types of triggers and insert one random type of trigger into each poisoned sample.

We use OpenBackdoor (Cui et al., 2022) for poisoned data generation. To be consistent with previous studies (Dai and Chen, 2019; Qi et al., 2021c; Jin et al., 2022), we adopt a poison rate of 5% for BadNet and InsertSent attack, and 20% for syntactic and multi-type attack (mixing 10%, 5%, and 5% of syntactic, BadNet, and InsertSent respectively). We also show the defense results under different poison rates in §4.3.

Baseline Methods We compare our method DPoE with four representative defense methods. (1) **ONION** (Qi et al., 2021a) detects and removes the suspicious words that are probably the backdoor triggers. GPT-2 (Radford et al., 2019) is used to evaluate the suspicion score of each word by the decrement of sentence perplexity after removing the word. (2) **BKI** (Chen and Dai, 2021), short for Backdoor Keyword Identification, detects trigger words and discards poisoned samples from the training data for purification.[7] (3) **STRIP** (Gao et al., 2021) filters out poisoned samples by checking the inconsistency of model's predictions when the input is perturbed several times.[8] (4) **RAP** (Yang et al., 2021b) uses a fixed perturbation and a threshold of the output probability change of the protect label (decided by the defender) to detect poisoned samples in the inference stage.

Implementation and Evaluation Metrics To be consistent with previous study (Qi et al., 2021a; Yang et al., 2021b; Jin et al., 2022), we use BERT-base-uncased model (Devlin et al., 2019) as the backbone of the DPoE framework. We also report results on Llama-2-7B (Touvron et al., 2023) to validate the effectiveness of the proposed algorithm on models of varying scales. All experiments are conducted on a single NVIDIA RTX A5000 (for BERT-base-uncased) or RTX 8000 (for Llama-2-7B). We train all models for 3 epochs and pick the best hyper-parameter based on the pseudo development set strategy (§3.4). The defense methods are evaluated with the following two metrics. (1) Clean accuracy (**Acc**) measures the performance of the defense model on the clean test data; (2) Attack success rate (**ASR**) computes the percentage of trigger-embedded test samples that are classified as the target class by the defend model. Following Jin et al. (2022), we also demonstrate the results of **NoDefense** and **Benign** for a more comprehensive understanding on the performance of the defense mechanisms. **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 poisoned samples. These two baselines are either provided with full prior knowledge of the attack, or free of attack, representing ideal situations that are not accessible to a defense model.

[7]Similarly, BKI leverages a scoring function to evaluate the importance of every single word to the model's prediction. A higher importance score indicates a higher probability for a word to be a trigger.

[8]The intuition is that it is difficult for any perturbation to the poisoned samples to influence the predicted class as long as the trigger exists.

Methods	BadNet		InsertSent		Syntactic		Multi-Type	
	ASR	Acc	ASR	Acc	ASR	Acc	ASR	Acc
OffensEval								
NoDef.	96.77	84.05	99.84	83.93	98.71	83.93	93.62	82.40
ONION	21.49	80.21	96.71	78.32	95.45	76.24	77.43	78.95
DPoE	**8.89**	**83.00**	**0.00**	**81.25**	**0.00**	**80.44**	**6.32**	**81.93**
SST-2								
NoDef.	93.75	95.77	99.89	96.49	95.18	95.88	91.93	94.41
ONION	26.98	89.31	98.90	87.84	94.82	83.26	87.57	84.29
DPoE	**7.32**	**94.63**	**15.26**	**94.89**	**19.54**	**93.67**	**16.33**	**93.85**

Table 2: Defense performance of DPoE with R-Drop on Llama 2. Best results are **boldfaced**. * NoDef. (NoDefense) is for reference and is not directly comparable with defense results.

4.2 Main Results

As shown in Tab. 1, our proposed DPoE method outperforms all of the four baselines and achieves the best defense performance on all of the three single-type trigger attacks as well as the muti-type trigger setting, especially the syntactic attack that most baseline methods fail to defend against. Since ONION and BKI are detection-based defense methods based on the assumption that triggers are rare words, the syntactic attack which does not involve explicit trigger words is not within their scope of defense. In contrast, DPoE leverages a shallow model to capture the backdoor shortcuts regardless of the type of triggers,[9] enabling the training of a

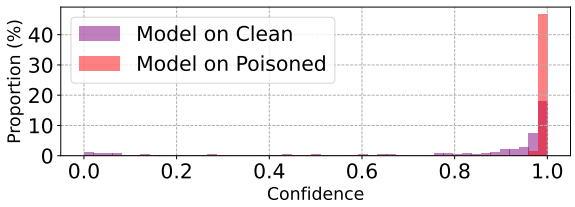
[9]We demonstrate in Appx. §D that DPoE remains robust when there is no backdoor in the training data.

backdoor-robust main model that does not learn the backdoor shortcut from triggers to the target label. Furthermore, DPoE achieves an even lower ASR than the Benign baseline in some cases (highlighted in blue), indicating that DPoE not only effectively defends backdoor triggers, but is also robust to the semantic shortcuts introduced by the insertion of triggers. More importantly, the clean Acc of DPoE can be higher than Benign model, enabling backdoor defense to no longer come at the expense of clean data accuracy.

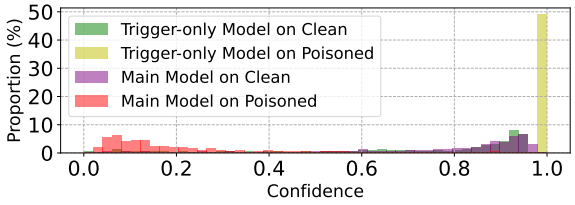
NoDefense and **Benign** provide an understanding of the attack effectiveness and the defense performance. The ASR of multi-type triggers exceeds 96% on BERT without defense for all of three datasets, indicating the effectiveness of the mix trigger attack which can induce the victim model to predict the target label almost certainly. Under the novel mix trigger attack, DPoE also manages to defend effectively with the ASR being close to or even lower than that of Benign. Besides, DPoE still maintains a competitive clean Acc compared with Benign and NoDefense, which demonstrates the effectiveness of the denoising technique that helps the model to be more resistant to noisy labels even under a high noise rate (approximately 20% for syntactic and multi-type attacks).

Compared with applying PoE alone, the clean Acc is significantly improved due to the denoising module of DPoE. On OffensEval, Acc under BadNet attack is only 81.72% by PoE defense, while exceeding 84% after applying the denoising technique, which performs even better than the Benign model. This is because the benign training data might already contain noisy labels that hinder the utility of the model, which is alleviated by the denoising module in DPoE. Similar conclusions can be made on all of the three datasets under four attacks. Overall, the incorporated denoising part further boosts defense performance, while no single denoising technique consistently outperforms the rest. The most effective denoising scheme for backdoor defense is left for future work.

To validate its effectiveness on models of larger scales, we also apply DPoE with R-Drop to Llama-2-7B. As illustrated in Tab. 2, DPoE outperforms the ONION baseline and achieves highly competitive defense performance under all of the four different types of attacks on both SST-2 and OffensEval datasets. For instance, DPoE maintains an ASR of below 10% on the OffensEval dataset under all of the four attack settings, while ONION fails to take effect under sentence- and syntactic-level attacks with ASR exceeding 95%. Similar observations can be found on the SST-2 dataset. As a result, DPoE remains effective along with the scale-up of the backbone language model, indicating its robustness in handling larger and more complex backbone architectures without significant loss in performance or efficiency.



(a) No Defense



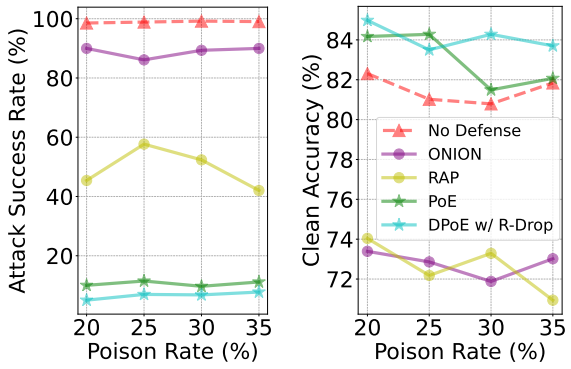
(b) DPoE w/ R-Drop

Figure 4: Prediction confidence distribution with (bottom) and without (top) DPoE defense. DPoE results in high confidence of trigger-only model on poisoned samples, enabling a backdoor-resistant robust main model.

4.3 Analysis

Effect of DPoE for Defense. To understand the influence of DPoE on the trigger-only model and the main model, we examine the confidence distribution of BERT without defense (Fig. 4(a)) and with DPoE (Fig. 4(b)) when trained on OffensEval poisoned by syntactic trigger at 20% poison rate. BERT without defense undoubtedly learns the backdoor shortcut and predicts most of the poisoned samples as the target label with almost 100% confidence. In contrast, the trigger-only model captures the backdoor shortcut and also predicts poisoned samples with high confidence, leaving the main model with trigger-free residual so that the main model learns to assign rather low confidence on poisoned data. This change in confidence distribution reveals the inner influence of DPoE for effectively preventing main model from learning the shortcut from backdoor triggers to target label.

Higher Poison Rate. To examine the resistance of DPoE against more devastating attacks, we challenge it with higher poison rate on the OffensEval



(a) ASR (b) Acc

Figure 5: Attack success rate (left) and clean accuracy (right) of defense methods by different poison rates on OffensEval task under syntactic attack. DPoE maintains competitive defense performance as poison rate rises.

task under syntactic attack. Fig. 5(a) shows that, for all the listed defense methods, there is not much increase in the ASR with the rise of poison rate, indicating that poison rate of 20% is enough for the victim model to be poisoned and sufficiently learn the backdoor shortcut. This phenomenon is consistent with previous study (Qi et al., 2021c). Though the ASR is not much affected by higher poison rate, there is an obvious decrease on the clean accuracy of baseline methods (ONION and RAP, Fig. 5(b)). The decrease in model utility is due to the fact that higher poison rate brings about more noisy labels, hindering the model from learning task-relevant features. In contrast, DPoE maintains stable performance of clean accuracy due to the denoising mechanism, indicating that DPoE remains competitive against more challenging attacks.

5 Conclusion

In this paper, we propose DPoE, an end-to-end ensemble-based backdoor defense method that mitigates backdoor triggers by learning the backdoor-free residual of a shallow model that captures the backdoor shortcuts. In addition to debiasing-based trigger mitigation and denoising techniques, a pseudo development set construction strategy is also proposed for hyper-parameter tuning since a clean dev set is absent in real-world scenarios. Experiments on three NLP tasks demonstrate its effectiveness in defending against various backdoor triggers as well as mix types of triggers.

Limitations

The current investigation of DPoE has the following limitations. First, although our experiments follow the settings of previous works for a fair comparison, the experimented tasks, types of triggers, languages, and backbone models can be further increased. Since our framework is model-agnostic, experimentation with more backbone language models can be conducted, which we leave as future work to due to limited bandwidth. Second, while we evaluate our method on discriminative NLU tasks to align with previous studies, the proposed method has the potential to be extended for generative tasks similarly as the contrastive decoding method (Li et al., 2022b). However, non-trivial adaption and systematic study will be needed to achieve this goal. Third, DPoE applies only to training time defense which assumes that the defender has access to the training phase of a model. We leave inference time defense for black-box models to future work.

Ethics Statement

In this paper, we propose a defense method against backdoor attacks with different types of triggers. Experimenting on three datasets that are publicly available, we show that our defense method effectively alleviates backdoor attacks without any prior knowledge about the backdoor triggers. Therefore, our framework provides an efficient solution to potential misuse of language models and protects models from malicious attacks. Besides, we also reveal one more adverse scenario of backdoor attack where various types of triggers are mixed together, disabling previous trigger-detection-based defense methods that assume the triggers to be rare words only. We would like to raise researchers' attention towards this potential risk and call for defense methods that can be universally adapted against various trigger types. Overall, the energy we consume for running the experiments is limited. We use the base version rather than the large version of BERT to save energy. No demographic or identity characteristics are used in this paper.

Acknowledgement

Qin Liu and Muhao Chen were supported by the NSF Grant IIS 2105329, the NSF Grant ITE 2333736, the DARPA AIE Grant HR0011-24-9-0370, the Faculty Startup Fund of UC Davis and an Amazon Research Award. Fei Wang was supported by the Amazon ML Fellowship. Chaowei Xiao was supported by the U.S. Department of Homeland Security under Grant 17STQAC00001-06-00.

References

Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. 2019. Unsupervised label noise modeling and loss correction. In *International conference on machine learning*, pages 312–321. PMLR.

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. *arXiv preprint arXiv:2103.04264*.

Chuanshuai Chen and Jiazhu Dai. 2021. Mitigating backdoor attacks in LSTM-based text classification systems by backdoor keyword identification. *Neurocomputing*, 452:253–262.

Pengfei Chen, Junjie Ye, Guangyong Chen, Jingwei Zhao, and Pheng-Ann Heng. 2021a. Beyond class-conditional assumption: A primary attempt to combat instance-dependent label noise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11442–11450.

Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. 2021b. Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In *Annual Computer Security Applications Conference*, pages 554–569.

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

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.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2020. Learning to model and ignore dataset bias with mixed capacity ensembles. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3031–3045, Online. 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. In *Proceedings of NeurIPS: Datasets and Benchmarks*.

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

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.

Tianqing Fang, Wenxuan Zhou, Fangyu Liu, Hongming Zhang, Yangqiu Song, and Muhao Chen. 2022. On-the-fly denoising for data augmentation in natural language understanding. *arXiv preprint arXiv:2212.10558*.

Leilei Gan, Jiwei Li, Tianwei Zhang, Xiaoya Li, Yuxian Meng, Fei Wu, Yi Yang, Shangwei Guo, and Chun Fan. 2022. Triggerless backdoor attack for NLP tasks with clean labels. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2942–2952, Seattle, United States. 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.

Abbas Ghaddar, Phillippe Langlais, Mehdi Rezagholizadeh, and Ahmad Rashid. 2021. End-to-end self-debiasing framework for robust NLU training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1923–1929, Online. 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*.

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2019. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244.

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.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31.

He He, Sheng Zha, and Haohan Wang. 2019. Unlearn dataset bias in natural language inference by fitting the residual. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 132–142, Hong Kong, China. Association for Computational Linguistics.

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

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. 2020. 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.

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.

Junnan Li, Richard Socher, and Steven C.H. Hoi. 2020. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*.

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.

Shikun Li, Xiaobo Xia, Shiming Ge, and Tongliang Liu. 2022a. Selective-supervised contrastive learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 316–325.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2022b. Contrastive decoding: Open-ended text generation as optimization. *arXiv preprint arXiv:2210.15097*.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. 2021b. Anti-backdoor learning: Training clean models on poisoned data. *Advances in Neural Information Processing Systems*, 34:14900–14912.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. 2021c. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *International Conference on Learning Representations*.

Zichao Li, Dheeraj Mekala, Chengyu Dong, and Jingbo Shang. 2021d. 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.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2018. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *Research in Attacks, Intrusions, and Defenses: 21st International Symposium, RAID 2018, Heraklion, Crete, Greece, September 10-12, 2018, Proceedings 21*, pages 273–294. Springer.

Tongliang Liu and Dacheng Tao. 2015. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461.

Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. 2020. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, pages 6448–6458. PMLR.

Yougang Lyu, Piji Li, Yechang Yang, Maarten de Rijke, Pengjie Ren, Yukun Zhao, Dawei Yin, and Zhaochun Ren. 2022. Feature-level debiased natural language understanding. *arXiv preprint arXiv:2212.05421*.

Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. 2020. Self: Learning to filter noisy labels with self-ensembling. In *International Conference on Learning Representations*.

Tai Nguyen, Yifeng Di, Joohan Lee, Muhao Chen, and Tianyi Zhang. 2023. Software entity recognition with noise-robust learning. In *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 484–496. IEEE.

Tuan Anh Nguyen and Anh Tuan Tran. 2021. Wanet - imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*.

Lu Pang, Tao Sun, Haibin Ling, and Chao Chen. 2022. Backdoor cleansing with unlabeled data. *arXiv preprint arXiv:2211.12044*.

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*.

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2014. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pages 4334–4343. PMLR.

Aniruddha Saha, Ajinkya Tejankar, Soroush Abbasi Koohpayegani, and Hamed Pirsiavash. 2022. Backdoor attacks on self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13337–13346.

Victor Sanh, Thomas Wolf, Yonatan Belinkov, and Alexander M Rush. 2021. Learning from others' mistakes: Avoiding dataset biases without modeling them. In *International Conference on Learning Representations*.

Zeyang Sha, Xinlei He, Pascal Berrang, Mathias Humbert, and Yang Zhang. 2022. Fine-tuning is all you need to mitigate backdoor attacks. *arXiv preprint arXiv:2212.09067*.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

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 Turner, Dimitris Tsipras, and Aleksander Madry. 2019. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020. Towards debiasing NLU models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7597–7610, Online. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. 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-IJCNLP 2023*.

Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. 2019b. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330.

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.

Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. 2020. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13726–13735.

M. J. Wolf, K. Miller, and F. S. Grodzinsky. 2017. Why we should have seen that coming: Comments on microsoft's tay "experiment," and wider implications. *SIGCAS Comput. Soc.*, 47(3):54–64.

Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu, et al. 2021. R-drop: Regularized dropout for neural networks. *Advances in Neural Information Processing Systems*, 34:10890–10905.

Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. 2021. Robust early-learning: Hindering the memorization of noisy labels. In *International Conference on Learning Representations*.

Xiaobo Xia, Tongliang Liu, Bo Han, Mingming Gong, Jun Yu, Gang Niu, and Masashi Sugiyama. 2022. Sample selection with uncertainty of losses for learning with noisy labels. In *International Conference on Learning Representations*.

Jun Yan, Vansh Gupta, and Xiang Ren. 2023. BITE: Textual backdoor attacks with iterative trigger injection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12951–12968, Toronto, Canada. 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.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Zaixi Zhang, Qi Liu, Zhicai Wang, Zepu Lu, and Qingyong Hu. 2023. Backdoor defense via deconfounded representation learning. *CoRR*, abs/2303.06818.

Zhiyuan Zhang, Lingjuan Lyu, Xingjun Ma, Chenguang Wang, and Xu Sun. 2022. Fine-mixing: Mitigating backdoors in fine-tuned language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 355–372, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wenxuan Zhou and Muhao Chen. 2021. Learning from noisy labels for entity-centric information extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5381–5392, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Appendices

A Justification of PoE for Defense

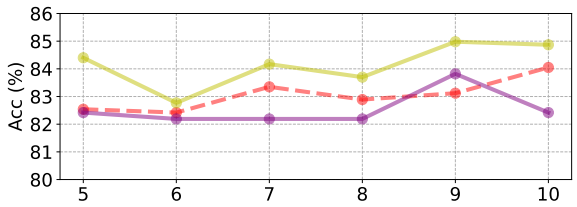
Probability of label y_i for example x_i in the PoE ensemble is computed as

$$\hat{p}_{iy_i} = \sigma(\log(p_{iy_i} \cdot b_{iy_i})) = \frac{p_{iy_i} \cdot b_{iy_i}}{\sum_{k=1}^{|\mathcal{Y}|} p_{ik} \cdot b_{ik}},$$

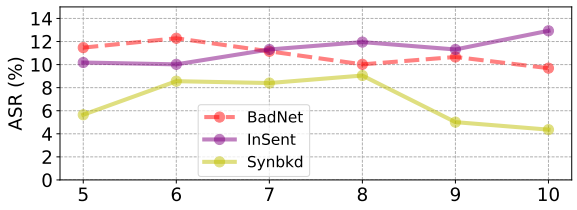
where σ denotes the softmax function. Then the gradient of the CE loss $\mathcal{L}(\theta_h; \theta_g)$ w.r.t. θ_g is (Karimi Mahabadi et al., 2020):

$$\nabla_{\theta_g} \mathcal{L}(\theta_h; \theta_g) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{|\mathcal{Y}|} \left[(\delta_{y_i k} - \hat{p}_{ik}) \nabla_{\theta_g} \log(p_{ik}) \right],$$

where $\delta_{y_i k}$ equals 1 when $k = y_i$ otherwise 0. Generally speaking, when both the trigger-only model and the main model have captured the backdoor associations, $\hat{p}_{ik}$ would be close to 1 so that $(\delta_{y_i k} - \hat{p}_{ik})$ is close to 0, decreasing the gradient of sample i. On the contrary, when the sample is trigger-free, the trigger-only model predicts the uniform distribution over all classes $b_{ik} \approx \frac{1}{|\mathcal{Y}|}$ for $k \in \mathcal{Y}$. Therefore, $\hat{p}_{iy_i} = p_{iy_i}$ and the gradient of PoE classifier remains the same as CE loss.



(a) Clean Accuracy



(b) Attack Success Rate

Figure 6: Performance of DPoE by PoE coefficient β on OffensEval task against three backdoor attacks. DPoE is steadily effective within a reasonable range of hyper-parameter values.

B Validity of Pseudo Development Set

We denote the poison rate, true clean accuracy of the main model on clean test data, and accuracy on pseudo clean dev set as $\alpha = |\mathcal{D}^*|/|\mathcal{D}|$, acc, and acc^* respectively. Assume the poison rate of the selected pseudo poisoned dev set $\mathcal{D}_P$ and pseudo clean dev is α_p and α_c respectively, the real attack success rate for the main model is asr (refer to the definition of metrics in §4.1), and the accuracy on $\mathcal{D}_P$ is asr_p.

Firstly, suppose the main model performs well on both the pseudo poisoned dev set and the poisoned training set, which is equivalent to low asr_p and high acc^*, it indicates low asr and high acc:

$$acc^* = (1 - \alpha_c) * acc + \alpha_c * asr,$$
$$asr_p = (1 - \alpha_p) * acc + \alpha_p * asr.$$

Due to the high-accuracy low-recall strategy, $\alpha_c * asr$ can be ignored since α_c is close to zero and we have $acc^* \propto acc$. On the other hand, a well-trained trigger-only model results in high α_p so that $asr_p \propto asr$. So we have demonstrated that we can infer from the low asr_p and high acc^* that the main model is effective for defense.

Secondly, suppose there exists a main model with high acc and low asr, which means it is an effective defense model indeed. Similarly, we have:

$$acc = \frac{acc^* - \alpha_c * asr}{1 - \alpha_c},$$
$$asr = \frac{asr_p - (1 - \alpha_p) * acc}{\alpha_p}.$$

Methods	SST-2	OffensEval	AG News
Finetune	91.16	83.47	<u>94.49</u>
PoE	91.27	83.29	94.32
DPoE w/ R-Drop	**91.76**	**85.10**	94.37
DPoE w/ LS	91.49	83.70	94.41
DPoE w/ Re-weight	<u>91.54</u>	83.93	94.21
DPoE w/ SL	91.43	<u>84.98</u>	**94.89**

Table 3: Clean accuracy of DPoE trained on clean datasets. The best results are **boldfaced** and the second best are <u>underlined</u>.

When the poison rate α_c is low, $(1-\alpha_c) \approx 1$ so that $acc \propto acc^*$. Further, an ideal main model indicates an effective trigger-only model that performs high α_p, which means $asr \propto asr_p$. So we have illustrated that a promising main model will be detected and selected by the pseudo dev set. As a result, our construction of pseudo dev set is valid since asr_p and acc^* on the pseudo dev set are effective approximations of asr and acc.

C Effect of PoE Coefficient

The coefficient β in Eq. 1 denotes the weight we assign to the predicted probability distribution of the trigger-only model in the framework of PoE. To examine whether our defense strategy is sensitive to this hyper-parameter, we evaluate DPoE with different β coefficient on the OffensEval task under three types of backdoor attacks. As shown in Fig. 6, the overall performance of DPoE slightly fluctuates with different coefficients, while indicating that DPoE remains effective within a reasonable range of hyper-parameter values.

D DPoE on Clean Dataset

To further examine the validity of DPoE, we train the model on the clean datasets of three tasks. Clean accuracy shown in Tab. 3 proves that DPoE does not hurt normal performance when being trained without triggers for the shallow model to capture. In this case, the trigger-only model learns superficial features (spurious correlations) that are no more desired than the trigger-related feature(s) and a robust main model should not make predictions based on these shallow features (Gardner et al., 2021). Thus, the trigger-only model's learning of shallow features would help the main model mitigate these shallow-feature-related spurious correlations and further boost its performance and robustness with the help of PoE.