The Trojan's Achilles'Heel: Leveraging Prediction Discrepancy for Poisoned Sample Identification

Anonymous ACL submission

Abstract

Backdoor attacks pose security threats to deep learning-based natural language processing models. This paper addresses the following question: Can a small number of clean samples be utilized to identify poisoned samples within large-scale untrusted data? We introduce a method for identifying backdoor-poisoned samples by leveraging prediction discrepancies, dubbed **PRISM** (**P**inpointing and **R**evealing **I**nfected **S**amples via **M**odel Discrepancy). The foundation of our approach stems from a core discovery: we observed a tendency where poisoned samples tend to cause model disagreement, while clean samples tend to maintain relative consensus when evaluated by both the weak benign and backdoor models. By quantifying this divergence through prediction distribution metrics and target-class probability differences, our approach enables precise identification of high-confidence poisoned samples. Experimental results confirm that our method effectively purifies training samples, thus facilitating the training of robust models and making the defense inherently generalizable across various complex attack types.

1 Introduction

With the growing use of deep neural networks (DNNs) in NLP, backdoor attacks have become a major security concern (Gu et al., 2017; Chen and Dai, 2021; Wu et al., 2025). While DNN performance relies heavily on training data scale and quality, the common practice of incorporating third-party data introduces significant risks. Attackers can implant hidden backdoors by poisoning a small fraction of training data and creating associations between specific triggers and target labels (Kurita et al., 2020; Dai et al., 2019). Unlike immediate adversarial attacks, backdoors remain dormant until deployment, posing serious threats in security-critical applications like spam filtering (Kurita et al., 2020), toxic content detection (Davidson et al., 2017) and so on. As models grow more complex and attack methods evolve (Wang et al., 2022; Tao et al., 2024), developing effective defenses against these increasingly stealthy attacks has become imperative.

To counter these threats, researchers have developed various defense strategies (Qi et al., 2021a; Gao et al., 2022; Cui et al., 2022; Kim et al., 2025). Existing approaches fall into two main categories: offline and online defenses. Offline methods focus on detecting and repairing trained models through techniques like fine-tuning (Kim et al., 2025; Shen et al., 2021), neuron pruning (Liu et al., 2018), and others. Online defenses are characterized by their deployment-time intervention, typically involving real-time detection on input texts—a strategy employed by methods such as ONION (Qi et al., 2021a).

The escalating sophistication of backdoor attacks has exposed critical limitations in existing defense paradigms. These approaches often struggle to generalize across diverse attack strategies and require customized adaptations for different model architectures, significantly constraining their practical applicability. In contrast, we advocate for a fundamental shift toward data-centric defense. Since most backdoor attacks ultimately rely on polluting training data, we posit that identifying and filtering poisoned samples at the data level can fundamentally disrupt the attack pathway. This purification strategy offers inherent advantages: it operates independently of specific model architectures, demonstrates stronger generalization across attack types, and most importantly, eliminates potential threats before model training begins—embodying a decisive "prevention-first" security paradigm.

Following this rationale, we propose PRISM (Pinpointing and Revealing Infected Samples via Model Discrepancy), a data purification method that addresses the following question: Can a small

number of clean samples be utilized to identify poisoned samples within large-scale untrusted data? Our approach is grounded in a core observation: backdoor models and benign models exhibit distinct behavioral tendencies when processing different samples. Specifically, poisoned samples tend to cause model disagreement, whereas clean samples tend to maintain relative consensus.

The main contributions of this paper are summarized as follows: 1) Through empirical analysis, we identify a key phenomenon: backdoor models and weak benign models exhibit significant prediction discrepancies when processing poisoned samples, while maintaining relative consensus on clean samples. This discovery provides the foundational basis for detecting backdoor attacks with limited clean data. 2) Leveraging the above observation, we propose a novel approach that precisely identifies high-confidence poisoned samples by quantifying inter-model prediction distribution differences (e.g., KL divergence) and target-class probability gaps. Our method requires only a small number of clean samples to establish a weak benchmark model, enabling effective purification of large-scale contaminated data. 3) We conduct comprehensive experiments on multiple pre-trained models across various NLP tasks. Results show that PRISM not only effectively detects traditional word-level triggers, but also maintains high detection accuracy against complex attacks such as style manipulation and syntactic structures, highlighting its superior generalization capability and practical utility.

2 Related Work

2.1 Backdoor Attacks

Backdoor attacks originated in computer vision with BadNet (Gu et al., 2017), which embeds triggers in training data to cause misclassification while preserving performance on clean inputs. In NLP, two primary attack routes have emerged: data poisoning and weight poisoning. Data poisoning has become the dominant approach due to its practicality and stealth. Notable categories of trigger-based methods encompass: (1) lexical triggers using words/phrases (Dai et al., 2019) or rare words (Kurita et al., 2020); (2) stylistic triggers employing invisible text features as in (Qi et al., 2021b); and (3) syntactic triggers manipulating dependency trees through grammatical structures rather than surface text (Qi et al., 2021c). These attacks demonstrate evolutionary trends: enhanced stealth through the progression from explicit lexical patterns to imperceptible stylistic and syntactic features; improved efficiency evidenced by Huang et al. (2024) showing that a mere 3% poisoning rate can successfully compromise the LLaMA-7B model; and strengthened persistence with Kurita et al. (2020) demonstrating that such backdoors can resist elimination through standard fine-tuning techniques.

2.2 Backdoor Defenses

Backdoor defense research primarily evolves along two main directions: offline defense and online defense. Offline defense encompasses (1) training-phase methods such as BKI (Chen and Dai, 2021) and CUBE (Cui et al., 2022) that aim to prevent backdoor implantation; (2) data-centric approaches including BMC (Wang et al., 2025) which filters poisoned samples by detecting abnormal memorization patterns; and (3) post-training remediation techniques comprising fine-tuning with clean data, knowledge distillation, neuron pruning, and gradient-based detection methods like DBS (Shen et al., 2022) and PICCOLO (Liu et al., 2022). Online defense focuses on real-time input screening during deployment, with representative schemes including ONION (Qi et al., 2021a), RAP (Yang et al., 2021), and STRIP (Gao et al., 2022). Our work is aligned with data-centric paradigms such as (Wang et al., 2025), which mitigates backdoors by filtering training samples that exhibit abnormal memorization. In contrast, our approach introduces a reference model trained on a small clean dataset to identify poisoned samples with high confidence, which enables the precise and rapid discovery of backdoor triggers.

3 Observation

3.1 Problem Formulation

We focus on defending against data poisoning-based backdoor attacks, the most prevalent type of backdoor attacks due to their practical applicability (Gu et al., 2017; Dai et al., 2019; Cui et al., 2022; Wang et al., 2025). In a data poisoning-based backdoor attack, the adversary contaminates the training dataset to produce a compromised model M that satisfies: $M(x) = y$ for clean samples, and $M(\tilde{x}) = y_T$ for poisoned samples containing triggers. Here, x denotes a clean input,$\tilde{x}$ a poisoned input,y the true label, and y_T the target label. The

attacker only modifies the training data without intervening in the training process.

We consider a classification task T with a backdoor-poisoned dataset D_p, where a portion of samples contain triggers. We assume access to a small clean proxy dataset D_r, with $|D_p| \gg |D_r|$. We denote model trained on D_p as M_p, and the reference model trained on D_r as M_r.

3.2 Basis for Separation: An Observational Study

Our intuition is that the backdoor model (M_p) and the reference clean weak model (M_r) will exhibit distinct behavioral differences when processing poisoned versus clean inputs. To validate this, we conducted a preliminary experiment using the SST-2 dataset (67,349 sentences). We constructed a minimal clean reference dataset D_r containing 500 sentences, while the remaining sentences were poisoned at a 10% rate (using target label "1") to form the backdoor-poisoned dataset D_p. This results in an extremely small reference dataset, with $|D_r| = 0.0075|D_p|$.

Evaluation procedure was conducted as follows:

1) Clean Sample Analysis. A set S_c was constructed by randomly selecting 20,000 clean samples. Each sample was fed into both M_p and M_r to obtain their corresponding class prediction probability distributions P_p and P_r. The KL divergence was then calculated in both directions: $KL1 = KL(P_p||P_r)$ and $KL2 = KL(P_r||P_p)$.

2) Poisoned Sample Analysis. A set S_p was constructed by selecting another 2,000 sentences and poisoning them via insertion of the trigger "tq". Each poisoned sample was fed into both models to obtain the prediction distributions Q_p and Q_r, and the KL divergence was similarly calculated as $KL1 = KL(Q_p||Q_r)$ and $KL2 = KL(Q_r||Q_p)$.

Figure 1 (a) illustrates the relationship between *KL1* and *KL2*. Their respective distributions are shown separately in Figure 1(b) for *KL1* and Figure 1(c) for *KL2*.

Figure 1 (a) reveals two discriminative patterns in the *KL1–KL2* space. First, poisoned samples approximately follow a monotonically increasing curve, while clean samples exhibit a roughly symmetric distribution. Second, poisoned samples generally reside above clean samples for the same *KL1* value, showing systematically higher *KL2* values. Although some clean samples are present within the poisoned cluster (see the 3D visualization in Appendix A), this vertical separability facilitates the identification of a substantial portion of poisoned data. Detailed theoretical analyses of these patterns are deferred to Appendix B and Appendix C due to space constraints.

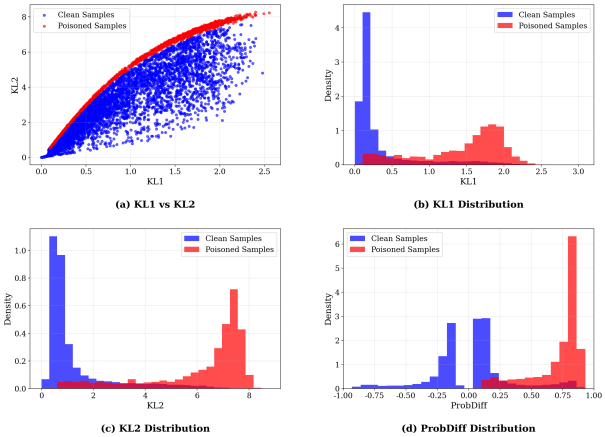


Figure 1: Distribution comparison.

As illustrated in Figures 1 (b) and 1(c), which compare the distributions of *KL1* and *KL2* respectively, a clear distinction in KL metric values emerges between poisoned and clean samples: nearly all poisoned samples fall on the right side with higher values, while the distribution peaks of clean samples (blue) are concentrated on the left with lower values. This pattern occurs because clean inputs elicit highly similar behaviors from both the poisoned model (M_p) and the reference model (M_r), resulting in small KL divergence values in both directions. In contrast, poisoned inputs trigger sharply different behaviors—The backdoor model M_p usually yields a highly confident, one-hot-like distribution for the attacked class. In contrast, the reference model M_r's output (whether sharp or smooth) reflects the true semantic content, leading to a significant distributional mismatch and thus a large KL divergence. Further theoretical analysis of this phenomenon is provided in Appendix D.

Furthermore, by examining the prediction probabilities for each sample, we identified distinct characteristics between poisoned and clean samples regarding the predicted probabilities for the backdoor target label y_T. We define a unified metric *ProbDiff* as the difference in the predicted probability for y_T between M_p and M_r.

Specifically, for a clean sample, let $P_p(y_T)$ and $P_r(y_T)$ denote the probability values assigned to y_T by M_p and M_r respectively. The probability difference is defined as: $ProbDiff_{clean} =$

$P_p(y_T) - P_{\boldsymbol{r}}(y_T)$. For a poisoned sample, let $Q_p(y_T)$ and $Q_{\boldsymbol{r}}(y_T)$ represent the corresponding probabilities. The difference is defined as: $ProbDiff_{poison} = Q_p(y_T) - Q_{\boldsymbol{r}}(y_T)$.

The histogram in Figure 1(d) reveals a clear divergence in these metrics: $ProbDiff_{clean}$ exhibits both positive and negative values, distributed roughly symmetrically around zero. In contrast, $ProbDiff_{poison}$ is almost exclusively positive and significantly larger in magnitude. We argue that the observed phenomenon stems from how models process poisoned versus clean samples. For poisoned samples, the backdoor in M_p activates, producing an extremely high probability for target class y_T, while M_r 's prediction $Q_{\boldsymbol{r}}(y_T)$, based solely on original semantics, remains low. This results in a large positive difference $Q_p(y_T) - Q_{\boldsymbol{r}}(y_T)$. In contrast, for clean samples, both models make semantics-based predictions, so probabilities $P_p(y_T)$ and $P_{\boldsymbol{r}}(y_T)$ may differ slightly due to normal model variations, leading to a near-zero probability difference that fluctuates symmetrically around zero. To understand the underlying mechanism, we also present a theoretical analysis of this phenomenon in Appendix E.

4 Method

Our findings in §3.2 indicate that a reference model—even a relatively weak classifier trained on limited clean data—can effectively assist in identifying poisoned samples. This approach remains viable even with progressively smaller reference datasets, as the distinctive characteristics illustrated in Figure 1 persist, a point further validated in § 5.3 of the experimental section. To this end, we designed a backdoor defense method based on an auxiliary clean model (Algorithm 1).

The core idea of Algorithm 1 is to first use the aforementioned metrics (probability distribution divergence and $Probdiff$, computed based on the prediction probability distributions of the poisoned model M_p and the clean model M_r for input samples) to conduct an initial screening of the training dataset, thereby identifying a subset of poisoned samples and clean samples (in line 1-7). Then, trigger patterns in these poisoned samples are extracted through high-frequency statistics, enabling the further identification of remaining poisoned samples that are difficult to distinguish (in line 8-11). Finally, a clean model is retrained using the purified samples.

Algorithm 1 Dual-Model Divergence Purifier

Input: Backdoor-poisoned Dataset D_p, Clean Reference Dataset D_r

Output: Model $\mathcal{M}_{bng}$

1: **Function** Identification(D_p, D_r)
2: ▷ Standard Training
3: M_p = StandardTraining(D_p)
4: M_r = StandardTraining(D_r)
5: ▷ Dual-Model Inference and screening
6: D' = Poisoned_Sample_Selection(D_p)
7: D'' = Clean_Sample_Selection(D_p)
8: ▷ Trigger Verification and Removal
9: $\overline{D}$ = Trigger_Verification(D', D")
10: $D_{bng} = D_p - \overline{D}$
11: ▷ Retraining on Purified Data D_{bng}
12: $\mathcal{M}_{bng}$ = StandardTraining(D_{bng})
13: **return** $\mathcal{M}_{bng}$
14: **End Function**

4.1 Dual-Model Inference and screening

According to the design of Algorithm 1, we need to identify with high confidence an appropriately sized set of poisoned samples D′ and a set of clean samples D″ from the contaminated dataset. D′ will be used to extract candidate trigger patterns, while D″ will serve to validate the identified triggers. To achieve this objective, we designed a dual-model screening framework illustrated in Figure 2.

As shown in Figure 2, For each input we feed it to both the reference model M_r and the poisoned model M_p to obtain their respective prediction probability distributions. We then calculate three feature metrics for each sample: bidirectional KL divergence (including *KL1* and *KL2*) and the probability difference for target class $ProbDiff$.

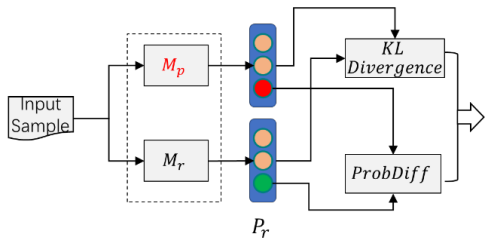


Figure 2: Dual-Model Inference Discrepancy.

Since samples with higher values of *KL1*, *KL2*, and $ProbDiff$ are more likely to be poisoned, we therefore employ a high-threshold screening approach to identify them with high confidence. We first select candidate samples with $ProbDiff > 0$, then calculate the top K% percentile of each fea-

ture as thresholds to construct a comprehensive anomaly scoring mechanism. This process yields the initially screened poisoned samples D′. Our ablation studies (see §5.4) demonstrate that the combination of *KL2* and $ProbDiff$ provides better screening performance, hence these two features primarily guide the poisoned sample selection.

From the relationship between *KL1* and *KL2* displayed in Figure 1(a) and Appendix A, we observe that the region below the curve containing poisoned samples consists predominantly of clean samples. Therefore, for high-confidence clean sample screening, we first fit the relationship curve between *KL1* and *KL2* using D' (see Appendix B for details), then select samples situated sufficiently below the fitted curve region (i.e., beyond the threshold τ_{KL}) as the initially screened clean samples D . In this paper, a stringent threshold of $\tau_{KL} = 1.0$ is set empirically.

4.2 Trigger Discovery and Validation

The objective of this is to perform commonality analysis on the identified poisoned samples to discover specific triggers and validate on the identified clean samples. This process helps further classify challenging samples. For instance, we observed that among the red samples in the upper part of Figure 1(a), there are still some clean samples that are difficult to classify. To address the three most typical trigger patterns in current backdoor attacks—namely, word frequency attacks, syntactic structure triggers, and style triggers—we designed a corresponding candidate trigger discovery and validation method based on frequent pattern mining. This method exhibits a certain degree of extensibility, allowing for the incorporation of additional commonality analysis techniques to validate other, more complex trigger patterns.

In lexical trigger identification, we analyze poisoned samples to extract frequently co-occurring word combinations—including single words, bigrams, and complex phrases—while applying frequency filters to exclude very common words and reduce false positives. For syntactic triggers, we parse poisoned sentences to generate grammatical templates and rank them by occurrence frequency. Candidate templates are evaluated by applying corresponding syntactic transformations—such as adverbial clause insertion or prepositional phrase fronting—to clean samples and measuring the attack success rate through the poisoned model. For style triggers, we use an enhanced style classifier to determine the dominant text style in the poisoned set and then validate it by reformatting clean samples with style-specific markers and testing their misclassification rate. For detailed descriptions, please refer to Appendix F.

4.3 Model Retraining

We apply the validated triggers on the remaining samples in the training set D_p beyond $D'$and D'', and denote the identified set of poisoned samples as $\tilde{D}$. In the final training phase, we integrate the poisoned sample subsets from both stages: $\overline{D} = D' \cup \tilde{D}$. Subsequently, we perform supervised learning on clean dataset $D_{bng} = D_p - \overline{D}$ using the standard cross-entropy loss function to train the final backdoor-free model:

$$\theta^* = \arg\min_{\theta} \sum_{(x,y)\in D_{\text{bng}}} L_{\text{CE}}(x, y, \theta) \qquad (1)$$

5 Experimental Results and Analysis

5.1 Setup

Datasets and hyperparameters. Our experiments are conducted on three benchmark datasets: SST-2 (Socher et al., 2013) for sentiment analysis, HSOL (Davidson et al., 2017) for hate speech detection, and AG's News (Zhang et al., 2015) for news categorization. We used the AdamW optimizer with an initial learning rate of 2×10^{-5} and trained all models for 3 epochs with a batch size of 16. Gradient clipping was applied with a maximum norm of 1.0 to ensure stable optimization.

Attack Methods. Three typical NLP backdoor attacks are selected: the word-level attack BadNets (Gu et al., 2017), the style-level attack (Qi et al., 2021b), and the structural-level attack (Qi et al., 2021c).

Baselines. We evaluate against a range of existing defenses, from training-phase methods like CUBE (Cui et al., 2022), to inference-phase techniques such as ONION (Qi et al., 2021a), RAP (Yang et al., 2021), and STRIP(Gao et al., 2022) , alongside BMC (Wang et al., 2025) for poisoned data filtration.

Evaluation Metrics. We follow (Qi et al., 2021b) in using two evaluation metrics: Attack Success Rate (ASR) and Benign Accuracy (BA). ASR quantifies the rate of successful backdoor triggers, and BA measures the classification accuracy on benign samples. An ideal defense minimizes ASR while preserving BA.

Table 1: Defense Effectiveness Comparison

(a) SST-2 & HSOL Dataset Results

Dataset	Attack	Metric	Oracle	Undef ended	ONION	STRIP	RAP	CUBE	BMC	PRISM
SST-2	BadNets	BA	90.76%	91.15%	88.41%	90.05%	86.88%	90.25%	90.72%	90.68%
		ASR	10.56%	100.00%	28.86%	99.28%	90.42%	15.61%	11.94%	**10.75**%
	Hidden Killer	BA	90.31%	90.24%	90.02%	89.79%	86.36%	91.38%	90.10%	90.24%
		ASR	29.87%	89.76%	88.94%	90.46%	93.18%	48.92%	35.66%	**32.42**%
	Style	BA	90.01%	90.48%	86.31%	87.53%	87.06%	89.72%	90.17%	90.15%
		ASR	28.33%	79.95%	80.82%	81.66%	85.58%	25.83%	21.85%	**19.42**%
HSOL	BadNets	BA	94.14%	95.64%	94.26%	95.18%	94.76%	94.84%	90.22%	91.28%
		ASR	7.13%	100.00%	46.28%	99.64%	100.00%	100.00%	19.20%	**17.36**%
	Hidden Killer	BA	94.35%	95.21%	94.53%	95.05%	94.21%	94.36%	94.18%	94.21%
		ASR	7.85%	97.08%	96.11%	98.38%	99.53%	10.19%	8.62%	**7.23**%
	Style	BA	94.98%	94.37%	93.65%	93.51%	94.18%	94.75%	94.23%	94.72%
		ASR	4.11%	71.55%	69.94%	70.86%	70.02%	6.68%	**5.77**%	6.25%

(b) AG's News Dataset Results

Dataset	Attack	Metric	Oracle	Undef ended	ONION	STRIP	RAP	CUBE	BMC	PRISM
AG's News	BadNets	BA	94.17%	94.35%	92.96%	93.63%	94.01%	94.05%	94.41%	94.32%
		ASR	1.08%	100.00%	97.26%	100.00%	100.00%	**0.89**%	1.40%	1.94%
	Hidden Killer	BA	90.96%	90.88%	89.84%	89.83%	91.15%	88.11%	90.13%	90.42%
		ASR	5.12%	98.12%	91.77%	99.18%	98.92%	5.63%	5.17%	**4.64**%
	Style	BA	91.33%	89.96%	89.87%	90.85%	90.13%	87.29%	88.76%	90.32%
		ASR	5.88%	84.47%	78.51%	80.77%	77.54%	5.05%	4.84%	**4.17**%

5.2 Defense Performance

This section presents a comprehensive evaluation of PRISM. We initiate our analysis by comparing the defense performance of our approach against existing NLP backdoor defense techniques, with detailed results documented in Table 1. The table compares defense effectiveness against multiple attack methodologies (BadNets, Hidden Killer, Style) across different datasets (SST-2, HSOL, AG's News) using both BA and ASR metrics. All experiments were conducted using BERT-base-uncased (Devlin et al., 2019) as the victim model, with a reference dataset size of 500 and a poisoning rate of 10%. To establish performance baselines, we include measurements from both undefended models and an ideal model trained on completely sanitized data—where all malicious samples are eliminated while preserving all benign instances.

Based on the comparative results across three datasets and three attack methods, our proposed PRISM demonstrates superior performance in backdoor defense. For instance, against the Style attack on AG's News, PRISM achieves a remarkably low 4.17% ASR while maintaining 90.32% BA, significantly outperforming other methods. Similar advantages are observed in HSOL and SST-2 datasets, where PRISM maintains lower ASR while preserving $>$ 90% BA across multiple attacks, confirming its effectiveness in neutralizing threats without compromising model utility. For complete analysis, we provide additional details on detected triggers along with precision and recall metrics for trigger detection in Appendix G.

5.3 The Impact of Reference Dataset Scale and Percentile Threshold K

We evaluated the impact of reference dataset (D_r) size on poisoned sample detection under different threshold settings (the top K% percentile as described in §4.1), with results shown in Table 2. A central goal was to determine the minimum amount of clean data required to build an effective reference model—a question of practical importance in resource-limited scenarios.

As the reference dataset size increased from 100 to 4000 samples, detection metrics such as precision, recall, and F1-score consistently improved. This indicates that more benign data helps the model better learn normal patterns and identify anomalous samples. Taking Table 2(b) as an example (with a threshold set at the top 90% percentile), the trend in precision across different dataset sizes can be divided into three phases, a pattern that remains consistent across other configurations: 1) Rapid growth (100–500 samples): Precision increased rapidly from 78.69% to 95.81%,

though it remained volatile in early stages. 2) Critical threshold (500–700 samples): Precision consistently exceeded 93%, rising from 95.81% to 97.92%, marking a qualitative leap in detection reliability. 3) High-confidence stability (700–3000 samples): Precision stabilized above 96%, eventually reaching 97.97%.

Table 2: Screening for Poisoned Samples under Different K% Thresholds.

(a) K=85

$\lvert D_r \rvert$	Perf. of Mr	Acc.	Pre.	Rec.	F1	Detected Count
100	0.5459	0.9298	0.7603	0.333	0.4631	876
200	0.5401	0.918	0.7294	0.155	0.2557	425
300	0.5998	0.9262	0.7327	0.296	0.4217	808
400	0.6353	0.9383	0.7316	0.507	0.5989	1386
500	0.8372	0.9635	0.8673	0.706	0.7784	1628
600	0.8016	0.9347	0.6701	0.5555	0.6074	1658
700	0.8635	0.9539	0.7739	0.6965	0.7332	1800
800	0.8739	0.9649	0.8493	0.7465	0.7946	1758
900	0.8853	0.9667	0.8747	0.74	0.8017	1692
1000	0.8704	0.964	0.846	0.739	0.7889	1747
2000	0.8830	0.9694	0.8802	0.7675	0.82	1744
3000	0.8876	0.9735	0.9146	0.7815	0.8428	1709

(b) K=90

$\lvert D_r \rvert$	Perf. of Mr	Acc.	Pre.	Rec.	F1	Detected Count
100	0.5459	0.9243	0.7869	0.23	0.3548	582
200	0.5401	0.9153	0.7528	0.1	0.1797	271
300	0.5998	0.9217	0.7434	0.21	0.3293	569
400	0.6353	0.937	0.8374	0.38	0.5237	910
500	0.8372	0.9568	0.9581	0.55	0.6976	1145
600	0.8016	0.9446	0.9315	0.42	0.5804	905
700	0.8635	0.9572	0.9792	0.54	0.6965	1104
800	0.8739	0.9585	0.9689	0.56	0.7106	1158
900	0.8853	0.9562	0.9625	0.54	0.6914	1121
1000	0.8704	0.9591	0.9766	0.56	0.7146	1154
2000	0.8830	0.9562	0.977	0.53	0.6876	1086
3000	0.8876	0.952	0.9797	0.48	0.6457	983

(c) K=95

$\lvert D_r \rvert$	Perf. of Mr	Acc.	Pre.	Rec.	F1	Detected Count
100	0.5459	0.9175	0.8163	0.12	0.2092	294
200	0.5401	0.912	0.7561	0.05	0.0876	123
300	0.5998	0.9161	0.7758	0.11	0.1911	281
400	0.6353	0.9278	0.9239	0.22	0.3612	486
500	0.8372	0.9321	0.9756	0.26	0.41	532
600	0.8016	0.93	0.9511	0.24	0.3871	511
700	0.8635	0.934	0.9946	0.28	0.4321	555
800	0.8739	0.9307	0.9877	0.24	0.3875	488
900	0.8853	0.9305	0.9719	0.24	0.3875	498
1000	0.8704	0.9289	0.9844	0.22	0.3616	450
2000	0.8830	0.9262	0.9896	0.19	0.3195	385
3000	0.8876	0.9231	0.9904	0.16	0.2688	314

The findings indicate that detection performance reaches a saturation point at a relatively small reference dataset size, revealing the inherent dependence of the model behavior difference detection method on the quality of the reference data. When the reference dataset is too small, the reference model fails to adequately learn the true distribution of clean data, resulting in underfitting and behavioral instability. In such cases, the observed behavioral differences from the poisoned model stem more from the reference model's own uncertainty than from the characteristics of poisoned samples. Conversely, when the reference dataset is moderately expanded, the reference model can more robustly characterize normal prediction patterns, making behavioral differences more reliably attributable to backdoor triggers and thereby significantly improving detection precision.

Table 3: Feature Combination Performance in Initial Poisoned Sample Screening.

Features	Acc.	Precision	Recall	F1
KL1	0.9399	0.7478	0.5115	0.6075
KL2	0.964	0.9415	0.644	0.7648
ProbDiff	0.9411	0.7573	0.518	0.6152
KL1, KL2	0.953	0.9472	0.5115	0.6643
KL1, ProbDiff	0.9408	0.7583	0.5115	0.6109
KL2, ProbDiff	**0.9568**	**0.9581**	**0.5485**	**0.6976**
KL1, KL2, ProbDiff	0.9535	0.947	0.518	0.6697

Our method prioritizes high-precision detection to ensure identified samples have a high probability of being backdoor-poisoned, providing a reliable foundation for subsequent trigger mining. As shown in Table 2(b), when the reference dataset size reaches 500, the detection precision significantly improves to 95.81%. At this precision level, the 1145 detected poisoned samples provide an adequate seed set for trigger discovery, achieving an optimal balance between data cost and model utility. When the percentile threshold increases from 85% to 95%, the detection process consistently enhances precision by filtering out samples with less pronounced anomalous features. Experiments in Table 2 also demonstrate an inverse correlation between detection threshold K and the number of identified poisoned samples. Complete experimental results regarding different thresholds and dataset sizes across additional attack scenarios are provided in Appendix H.

5.4 Ablation Study

We conducted experiments to identify the most effective feature combinations for the initial screening of poisoned samples within the PRISM detection framework. As shown in Table 3, which presents results under the BadNets (Gu et al., 2017) attack scenario, among all feature combinations, the combination of *KL2* and $ProbDiff$ demonstrates optimal performance in precision. This re-

Table 4: Robustness on different poisoning rate.

Poisoning Rate	Undefended		ONION		STRIP		RAP		CUBE		BMC		PRISM	
	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR
1.00	91.85%	95.04%	90.04%	21.83%	90.60%	72.60%	90.82%	72.60%	90.37%	25.80%	90.05%	10.13%	90.21%	**9.11%**
5.00	91.05%	100.00%	89.73%	33.69%	90.22%	96.68%	87.54%	93.72%	88.46%	11.99%	91.62%	11.71%	92.14%	**9.58%**
10.00	91.15%	100.00%	88.41%	28.86%	90.05%	99.28%	86.88%	90.42%	90.25%	15.61%	90.72%	11.94%	90.68%	**10.75%**
20.00	90.44%	100.00%	90.54%	45.66%	85.72%	97.17%	85.33%	90.64%	87.80%	16.39%	90.13%	12.16%	91.45%	**11.21%**

Table 5: Effectiveness on different models with more results of baseline methods.

Model	Undefended		ONION		STRIP		RAP		CUBE		BMC		PRISM	
	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR	BA	ASR
DistilBERT	88.76%	100.00%	89.73%	100.00%	85.55%	92.73%	86.27%	94.16%	90.38	13.97	90.25%	8.78%	90.41%	**7.25%**
RoBERTa	93.23%	100.00%	92.54%	100.00%	91.16%	97.34%	91.32%	94.82%	93.23	6.75	93.21%	5.63%	93.53%	**5.24%**
ALBERT	91.94%	99.77%	91.87%	99.89%	84.02%	87.67%	84.62%	89.10%	92.69	8.80	91.74%	7.65%	92.60%	**6.46%**

sult surpasses other single or combined features, indicating that *KL2* and ProbDiff exhibit high reliability in identifying initial poisoned samples, particularly in scenarios requiring high precision.

We also find similar patterns hold for the other two attack methods—the style-level attack (Qi et al., 2021b) and the structural-level attack (Qi et al., 2021c) (experimental results are omitted here due to space constraints). Although the full feature combination may slightly improve overall accuracy, the primary objective of our ablation study is to enhance the precision of identifying initial poisoned samples in the first stage. Therefore, we use the combination of *KL2* and $ProbDiff$ for the experiments in this paper. This configuration maintains high accuracy while delivering optimal precision, fully satisfying the high-confidence requirement for initial poisoned sample detection.

5.5 Robustness and Generalizability

In this section, we investigate the robustness and generalization capability of the proposed PRISM method. The following experiments share a common configuration: the SST-2 dataset (Socher et al., 2013), a BERT-base-uncased model (Devlin et al., 2019), the BadNets attack (Gu et al., 2017).

We first evaluate the robustness of PRISM against varying data poisoning rates. As summarized in Table 4, PRISM consistently maintains a high Benign Accuracy (BA) and the lowest Attack Success Rate (ASR) across all tested rates. This stable performance demonstrates the method's resilience across a range of poisoning intensities.

To assess the generalization capability of our approach across different model architectures, we evaluated the Benign Accuracy (BA) and Attack Success Rate (ASR) on various victim models under a 10% poisoning rate, with the comprehensive results detailed in Table 5. The experiments employed commonly used NLP models, namely DistilBERT (Sanh et al., 2019), RoBERTa (Liu et al., 2019), and ALBERT (Lan et al., 2020). The results demonstrate that our method performs consistently well across all architectures, confirming its strong generalizability.

6 Conclusion

In this work, we introduced PRISM, a novel and effective defense paradigm that addresses critical challenge of backdoor attacks in NLP by demonstrating that even a small set of clean reference data can be leveraged to successfully identify poisoned samples within large-scale, untrusted datasets. The foundation of PRISM rests on a key observation: a discernible discrepancy in model behavior between clean and poisoned samples. We found that poisoned samples consistently induce disagreement between a benign and a backdoor-influenced model, while clean samples maintain predictive consensus. PRISM quantifies this divergence through tailored metrics, enabling the high-confidence pinpointing of poisoned instances. Comprehensive experiments validate that PRISM successfully purifies training datasets using only minimal clean reference data, thereby facilitating the training of robust models and demonstrating inherent generalizability across a wide spectrum of complex attack scenarios. This work establishes a simple, yet powerful and practical path toward securing NLP models in real-world, untrusted data environments, proving that effective defense is achievable even with limited trusted resources.

7 Limitations

Despite its demonstrated effectiveness, PRISM has certain limitations that present opportunities for future work. First, the method's performance is inherently tied to the quality and representativeness of the small, trusted reference dataset; a significant domain shift between this clean set and the untrusted data could impair its efficacy. Second, while PRISM is designed to be efficient, the computational overhead of conducting dual-model inference on a very large-scale dataset remains non-negligible, which could be a practical concern in resource-constrained settings. Finally, our current approach primarily defends against attacks that manifest in the model's prediction distribution. We have not yet fully explored its robustness against adaptive adversaries who are specifically aware of PRISM's mechanism and may attempt to design more stealthy attacks to evade detection. Investigating defenses in such a stronger threat model is a crucial next step.

8 Ethical considerations

This research, which focuses on developing defensive mechanisms against backdoor attacks in NLP, presents important ethical considerations. First, the primary intent and potential benefit of our work is protective: the proposed PRISM method aims to enhance model security by purifying training data, thereby helping to build more trustworthy and reliable NLP systems. Second, we acknowledge a potential dual-use risk: a detailed understanding of our detection mechanism could, in theory, inform adversaries in designing more sophisticated and stealthy attacks to evade similar defense strategies. Finally, we affirm that our experiments were conducted responsibly using standard, public benchmark datasets in a controlled research setting, with no involvement of private user data or deployment against real-world systems.

References

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

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 *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Jiazhu Dai, Chuanshuai Chen, and Yufeng Li. 2019. A backdoor attack against lstm-based text classification systems. *IEEE Access*, 7:138872–138878.

Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2017, Montréal, Québec, Canada, May 15-18, 2017*, pages 512–515. AAAI Press.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Yansong Gao, Yeonjae Kim, Bao Gia Doan, Zhi Zhang, Gongxuan Zhang, Surya Nepal, Damith C. Ranasinghe, and Hyoungshick Kim. 2022. Design and evaluation of a multi-domain trojan detection method on deep neural networks. *IEEE Trans. Dependable Secur. Comput.*, 19(4):2349–2364.

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

Hai Huang, Zhengyu Zhao, Michael Backes, Yun Shen, and Yang Zhang. 2024. Composite backdoor attacks against large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, volume NAACL 2024 of *Findings of ACL*, pages 1459–1472. Association for Computational Linguistics.

Jaehan Kim, Minkyoo Song, Seung Ho Na, and Seungwon Shin. 2025. Obliviate: Neutralizing task-agnostic backdoors within the parameter-efficient fine-tuning paradigm. In *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, volume NAACL 2025 of *Findings of ACL*, pages 1288–1307. Association for Computational Linguistics.

Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight poisoning attacks on pre-trained models. *CoRR*, abs/2004.06660.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised

learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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*, volume 11050 of *Lecture Notes in Computer Science*, pages 273–294. Springer.

Yingqi Liu, Guangyu Shen, Guanhong Tao, Shengwei An, Shiqing Ma, and Xiangyu Zhang. 2022. Piccolo: Exposing complex backdoors in NLP transformer models. In *43rd IEEE Symposium on Security and Privacy, SP 2022, San Francisco, CA, USA, May 22-26, 2022*, pages 2025–2042. IEEE.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

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, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 9558–9566. 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, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 4569–4580. 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 443–453. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Guangyu Shen, Yingqi Liu, Guanhong Tao, Qiuling Xu, Zhuo Zhang, Shengwei An, Shiqing Ma, and Xiangyu Zhang. 2022. Constrained optimization with dynamic bound-scaling for effective NLP backdoor defense. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 19879–19892. PMLR.

Lujia Shen, Shouling Ji, Xuhong Zhang, Jinfeng Li, Jing Chen, Jie Shi, Chengfang Fang, Jianwei Yin, and Ting Wang. 2021. Backdoor pre-trained models can transfer to all. In *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*, pages 3141–3158. ACM.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642. ACL.

Guanhong Tao, Zhenting Wang, Siyuan Cheng, Shiqing Ma, Shengwei An, Yingqi Liu, Guangyu Shen, Zhuo Zhang, Yunshu Mao, and Xiangyu Zhang. 2022. Backdoor vulnerabilities in normally trained deep learning models. *CoRR*, abs/2211.15929.

Guanhong Tao, Zhenting Wang, Shiwei Feng, Guangyu Shen, Shiqing Ma, and Xiangyu Zhang. 2024. Distribution preserving backdoor attack in self-supervised learning. In *IEEE Symposium on Security and Privacy, SP 2024, San Francisco, CA, USA, May 19-23, 2024*, pages 2029–2047. IEEE.

Zhenting Wang, Zhizhi Wang, Mingyu Jin, Mengnan Du, Juan Zhai, and Shiqing Ma. 2025. Data-centric NLP backdoor defense from the lens of memorization. In *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, volume NAACL 2025 of *Findings of ACL*, pages 5713–5731. Association for Computational Linguistics.

Zhenting Wang, Juan Zhai, and Shiqing Ma. 2022. Bppattack: Stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 15054–15063. IEEE.

Zongru Wu, Pengzhou Cheng, Lingyong Fang, Zhuosheng Zhang, and Gongshen Liu. 2025. Gracefully filtering backdoor samples for generative large language models without retraining. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 3267–3282. Association for Computational Linguistics.

Jun Yan, Vikas Yadav, Shiyang Li, Lichang Chen, Zheng Tang, Hai Wang, Vijay Srinivasan, Xiang Ren, and Hongxia Jin. 2024. Backdooring instruction-tuned large language models with virtual prompt injection. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 6065–6086. Association for Computational Linguistics.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021. 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, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8365–8381. Association for Computational Linguistics.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Yu Zhang, Houquan Zhou, and Zhenghua Li. 2020. Fast and accurate neural CRF constituency parsing. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 4046–4053. ijcai.org.

A The Impact of Reference Dataset Size on Discrepancies.

In Figure 3, we analyze the feature-space differences between poisoned and clean samples as a function of the reference dataset's size. The results demonstrate a clear trend: the separability between these two sample classes improves significantly with an increase in the reference dataset's scale. A key inflection point is observed when the dataset reaches about 200-400 samples, at which point effective separation is already achieved. Furthermore, when the reference dataset expands to a substantial size—such as 500 samples in Figure 3(e)—the distinction between poisoned and clean samples tend to stabilize, indicating that the model's capability to differentiate between them saturates beyond a certain data volume.

Through multi-view 3D visualization, we also systematically explored the distribution characteristics of samples in the feature space for backdoor attack detection in Figure 4 ($|D_r| = 500$). Specifically, we selected four representative viewing angle parameter combinations—including (a) the baseline view (elev=30°, azim=45°), (b) the symmetric view (elev=30°, azim=225°),(c) the right-side observation view (elev=30°, azim=315°), and (d) the left-side observation view (elev=30°, azim=135°), to comprehensively observe the three-dimensional feature space composed of KL divergence features ($KL1$, $KL2$) and probability difference features ($ProbDiff$). The experimental results in Figure 4 indicate that poisoned samples do exhibit distinct clustering phenomena in the feature space. However, we also find that the red regions may also contain some clean samples. Despite this, the vast majority of poisoned samples remain concentrated within these red regions. In contrast, the relatively pure blue regions—particularly those located farther away from the red zones—contain almost no poisoned samples.

B The Smooth Monotonic Curve of Poisoned Samples in KL-Divergence Space

As illustrated in Figure 1(a), poisoned samples exhibit a distinct, smooth monotonic curve within the KL-Divergence space. This clear empirical relationship prompts a deeper theoretical inquiry, which we attempt to formally explain the underlying mechanisms for this phenomenon.

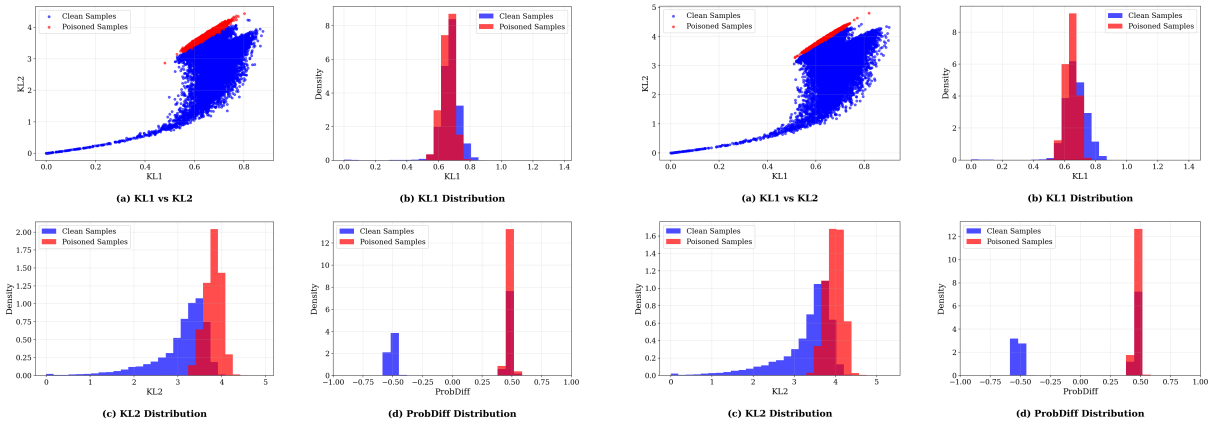
For poisoned samples, the output distribution Q_p from the poisoned model M_p typically approximates a near one-hot distribution, characterized by a predominant probability mass on the target class y_T. In contrast, the distribution Q_r produced by the weak reference model M_r is generally much smoother.

We consider a binary classification scenario (as is the case with the SST-2 task), with classes 0 and 1 (assuming the attack target class $y_T = 1$). Let $Q_p = \left[q_p^0, q_p^1\right] \approx$[ε, 1−ε], $Q_r = \left[q_r^0, q_r^1\right] \approx$[b, 1−b], where ϵ is a very small positive number (ensured by the $Softmax$ function to be non-zero). The two KL divergences are:

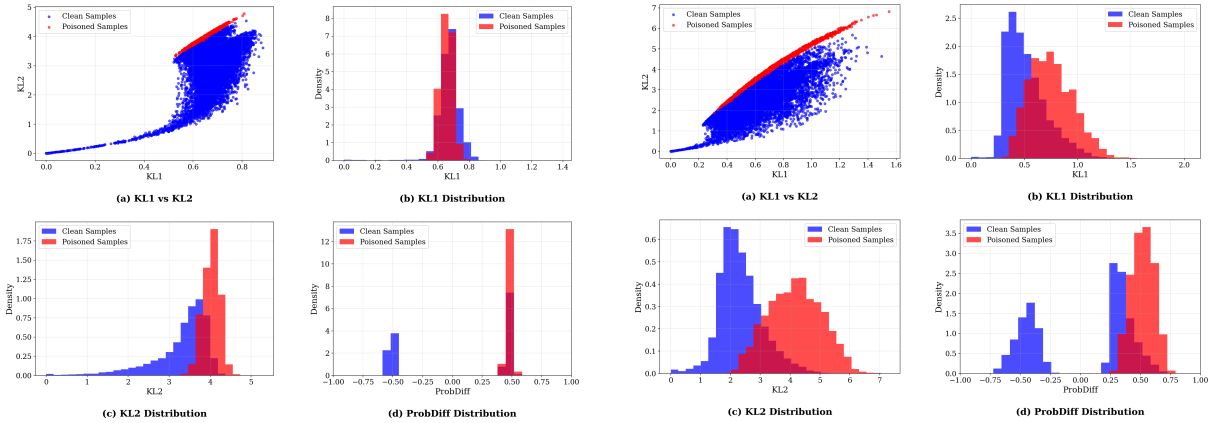
$$\mathrm{KL1} = \mathrm{KL}(Q_p \parallel Q_r) = \sum_{i=0}^{1} q_p^i \log\left(\frac{q_p^i}{q_r^i}\right) \quad (2)$$

$$\mathrm{KL2} = \mathrm{KL}(Q_r \parallel Q_p) = \sum_{i=0}^{1} q_r^i \log\left(\frac{q_r^i}{q_p^i}\right) \quad (3)$$

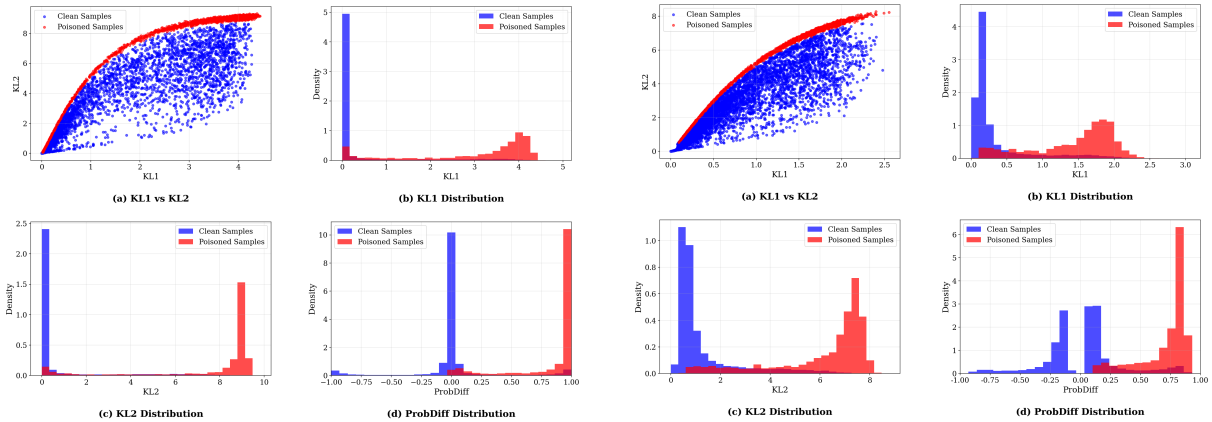
Substituting the approximate values of Q_p and Q_r: First, compute $KL1$: $KL1 \approx \log\left(\frac{\epsilon}{b}\right) + (1-\epsilon) \cdot$



(a) $|D_r| = 20$



(b) $|D_r| = 100$



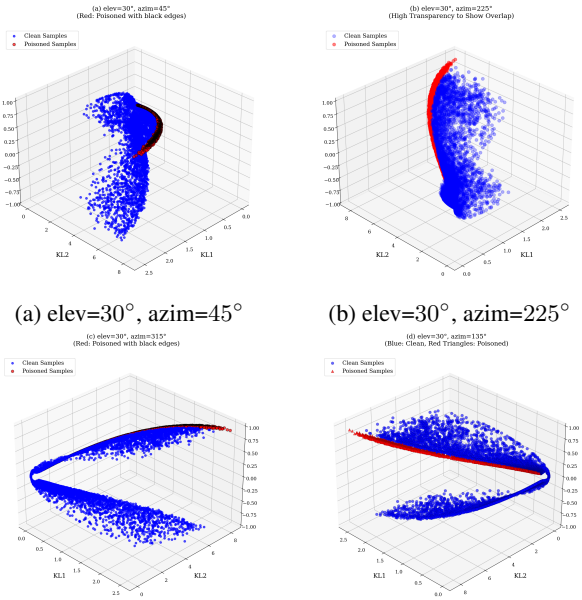
(c) $|D_r| = 200$

(d) $|D_r| = 400$

(e) $|D_r| = 500$

(f) $|D_r| = 1000$

Figure 3: The Impact of Reference Dataset Size on Discrepancies.



(a) elev=30°, azim=45° (b) elev=30°, azim=225°

(c) elev=30°, azim=315° (d) elev=30°, azim=135°

Figure 4: Multi-view Visualization.

$\log\left(\frac{1-\epsilon}{1-b}\right)$. Since $\epsilon \approx 0$, $1-\varepsilon \approx 1$ and $\epsilon \log\left(\frac{\epsilon}{b}\right) \to 0$, the above expression can be simplified to:

$$\mathrm{KL1} \approx 0 + 1 \cdot \log\left(\frac{1}{1-b}\right) = -\log(1-b) \quad (4)$$

Next, compute $KL2$: $KL2 = b \cdot \log\left(\frac{b}{\epsilon}\right) + (1-b) \cdot \log\left(\frac{1-b}{1-\epsilon}\right) \approx b \cdot (\log b - \log \epsilon) + (1-b) \cdot (\log(1-b) - \log(1-\epsilon))$. Since $\epsilon \approx 0$, $1-\varepsilon \approx 0$ and $-\log \epsilon$ is a large positive number (denoted as $\mathrm{C} = -\log \varepsilon$), the expression can be simplified to: $KL2 \approx b \cdot \log b - b \cdot \log \epsilon + (1-b) \cdot \log(1-b)$

then we have

$$\mathrm{KL2} \approx \underbrace{[b \cdot \log b + (1-b) \cdot \log(1-b)]}_{H(b)} + C \cdot b \quad (5)$$

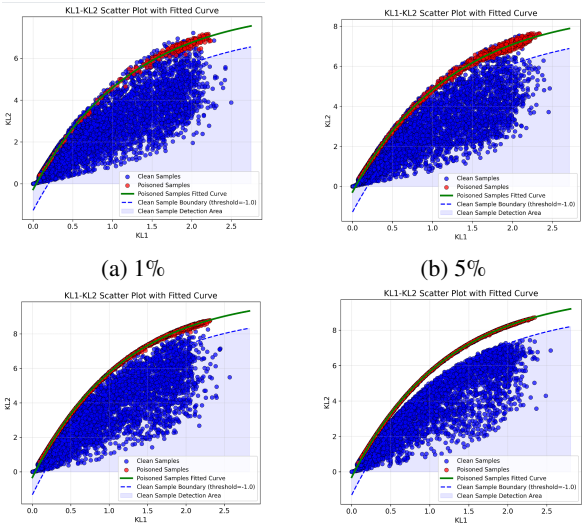
where $H(b)$ is the entropy of the Bernoulli distribution Q_r.

Then establish the relationship between $KL1$ and $KL2$: Since $KL1 \approx -\log(1-b)$ it follows that: $1-b \approx e^{-KL1}$, $b \approx 1-e^{-KL1}$. Substitute b into the expression for $KL2$:

$$\mathrm{KL2} \approx H(b) + C \cdot b = H\left(1-e^{-\mathrm{KL1}}\right) + C \cdot \left(1-e^{-\mathrm{KL1}}\right) \quad (6)$$

This functional relationship $KL2 = f(KL1)$ defines a smooth curve. Due to the large number C the term $C\,(1-e^{-KL1})$ dominates the value of $KL2$ causing $KL2$ to increase monotonically with $KL1$, thereby forming the observed smooth curve.

As described in Algorithm 1, we leverage the smooth curve preceding $KL1$ and $KL2$ of poisoned samples and identify samples lying beyond a certain distance below this curve as clean. To illustrate this, Figure 5 presents the actual fitting effects under different poisoning ratios and the regions of identified clean samples.



(a) 1% (b) 5%

(c) 10% (d) 20%

Figure 5: The Fitting Curve between KL1 and KL2.

C Approximate Symmetry of Clean Samples

As shown in Figure 1(a), the sample points $(KL1, KL2)$ in the clean sample set generally cluster near the diagonal line and exhibit a symmetric distribution. A plausible explanation for this involves the following factors:

Similarity in Model Behavior. The reference model M_r is trained on clean data D_r. Although the poisoned model M_p is trained on D_p containing some poisoned data (e.g., 10%), its primary component remains clean data. A successful backdoor attack requires maintaining the model's performance on clean samples (i.e., the stealthiness of the attack). Therefore, compared to poisoned samples, the prediction distribution P_p of M_p for the vast majority of clean samples is relatively close to the prediction distribution P_r of M_r. That is, for clean samples, P_p is relatively close to P_r.

Local Symmetry of KL Divergence. KL divergence $KL(P\|Q)$ is inherently asymmetric, meaning $KL(P\|Q) \neq KL(Q\|P)$. However, when two distributions P and Q are very close ($P \approx Q$), both $KL(P\|Q)$ and $KL(Q\|P)$ are not only small in value but also very close to each other. This

"closeness" implies that if $KL(P\|Q)$ is slightly larger than $KL(Q\|P)$ for one sample, it might be slightly smaller for the next sample. This deviation is random and fluctuates positively and negatively around zero. This causes the scatter points ($KL(P\|Q)$, $KL(Q\|P)$) not to lie perfectly on the line $y = x$, but rather to be distributed evenly on both sides of this line, thus forming a symmetric pattern around the diagonal. "Symmetry" means that for any point (a, b) appearing in the plot, we are likely to find a similar point (b, a) near its symmetric position relative to the line $y = x$. This symmetry is a direct consequence of the interchangeable roles of P_p and P_r, as their status is equivalent in the definition of the KL divergences (both being approximations of the "true" clean distribution).

Theoretical Analysis. Let the distribution of the reference model be P_r , Let the distribution of the poisoned model be P_p , $P_p = P_r + \delta$, where $\delta = [\delta_0, \delta_1, ..., \delta_{k-1}]$ is a small perturbation vector (k is the number of classes), and it satisfies the constraint $\sum_{i=0}^{k-1} \delta_i = 0$ (since both P_p and P_r are probability distributions that sum to 1). Our goal is to compute $KL(P_p\|P_r)$ and $KL(P_r\|P_p)$ and demonstrate their relationship.

Compute KL($\mathbf{P_p}\|\mathbf{P_r}$):

$$\begin{aligned}
&\mathrm{KL}(P_p \parallel P_r) \qquad (7)\\
&= \sum_{i=0}^{k-1} (p_r^i + \delta_i) \log\left(\frac{p_r^i + \delta_i}{p_r^i}\right)\\
&= \sum_i (p_r^i + \delta_i) \log\left(1 + \frac{\delta_i}{p_r^i}\right)
\end{aligned}$$

Perform a third-order Taylor expansion of $\log(1 + x)$ around $x = 0$ $\log(1 + x) \approx x - \frac{x^2}{2} + \frac{x^3}{3} + \mathrm{O}(x^4)$. Substitute into the above equation:

$$\begin{aligned}
&\mathbf{KL}(\mathbf{P_p}\|\mathbf{P_r})\\
&\approx \sum_i \left(p_r^i + \delta_i\right)\left(\frac{\delta_i}{p_r^i} - \frac{1}{2}\left(\frac{\delta_i}{p_r^i}\right)^2 + \frac{1}{3}\left(\frac{\delta_i}{p_r^i}\right)^3\right)\\
&= \sum_i \left(\left(p_r^i + \delta_i\right)\frac{\delta_i}{p_r^i} - \left(p_r^i + \delta_i\right)\frac{1}{2}\left(\frac{\delta_i}{p_r^i}\right)^2 + \left(p_r^i + \delta_i\right)\frac{1}{3}\left(\frac{\delta_i}{p_r^i}\right)^3\right)\\
&= \sum_i \left(\delta_i + \frac{(\delta_i)^2}{p_r^i} - \frac{1}{2}\frac{(\delta_i)^2}{p_r^i} - \frac{1}{2}\frac{(\delta_i)^3}{(p_r^i)^2} + \frac{1}{3}\frac{(\delta_i)^3}{(p_r^i)^2} + \frac{1}{3}\frac{(\delta_i)^4}{(p_r^i)^3}\right)
\end{aligned}$$

$\approx \sum_i \left(\delta_i + \frac{1}{2}\frac{(\delta_i)^2}{p_r^i} - \frac{1}{6}\frac{(\delta_i)^3}{(p_r^i)^2}\right)$ (Ignoring fourth-order and higher-order small quantities). According to the constraint condition $\sum_i \delta_i = 0$,

$$\begin{aligned}
&\mathrm{KL}(P_p \parallel P_r) \approx\\
&\frac{1}{2}\sum_i \frac{\delta_i^2}{p_r^i} - \frac{1}{6}\sum_i \frac{\delta_i^3}{(p_r^i)^2} \qquad (8)
\end{aligned}$$

Compute KL($\mathbf{P_r}\|\mathbf{P_p}$) :

$$\begin{aligned}
\mathrm{KL}(P_r \parallel P_p) &= \sum_i p_r^i \log\left(\frac{p_r^i}{p_r^i + \delta_i}\right)\\
&= \sum_i p_r^i \log\left(\frac{1}{1 + \frac{\delta_i}{p_r^i}}\right) \qquad (9)\\
&= -\sum_i p_r^i \log\left(1 + \frac{\delta_i}{p_r^i}\right)
\end{aligned}$$

Let $x_i = \frac{\delta_i}{p_r^i}$, perform a third-order Taylor expansion of $\log(1 + x)$ around $x = 0$:$\log(1 + x) \approx x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \ldots$, substitute into the above equation,

$$\begin{aligned}
&\mathrm{KL}(P_r \parallel P_p) \approx \qquad (10)\\
&-\sum_i p_r^i \left(\frac{\delta_i}{p_r^i} - \frac{1}{2}\left(\frac{\delta_i}{p_r^i}\right)^2 + \right.\\
&\left.\frac{1}{3}\left(\frac{\delta_i}{p_r^i}\right)^3 - \frac{1}{4}\left(\frac{\delta_i}{p_r^i}\right)^4 + \ldots\right)\\
&=\\
&-\sum_i \delta_i + \frac{1}{2}\sum_i \frac{\delta_i^2}{p_r^i} -\\
&\frac{1}{3}\sum_i \frac{\delta_i^3}{(p_r^i)^2} + \frac{1}{4}\sum_i \frac{\delta_i^4}{(p_r^i)^3} - \ldots
\end{aligned}$$

According to the constraint condition $\sum_i \delta_i = 0$ and ignore fourth-order and higher-order small quantities,

$$\mathrm{KL}(P_r \parallel P_p) \approx \frac{1}{2}\sum_i \frac{\delta_i^2}{p_r^i} - \frac{1}{3}\sum_i \frac{(\delta_i)^3}{(p_r^i)^2} \qquad (11)$$

Compared with the previously derived expression $\mathbf{KL}(\mathbf{P_p}\|\mathbf{P_r}) \approx \frac{1}{2}\sum_i \frac{(\delta_i)^2}{p_r^i} - \frac{1}{6}\sum_i \frac{(\delta_i)^3}{(p_r^i)^2} (KL1)$, it can be observed that the two KL divergences still share identical and symmetric second-order terms. The common and dominant second-order term forces all points to lie close to the line $y = x$. The third-order term, which has the same sign but different magnitudes, causes random deviations of the points relative to the diagonal. When the third-order term > 0, $KL2$ decreases more than $KL1$,

resulting in $KL1 > KL2$ (points below the diagonal). When the third-order term < 0, $KL2$ increases more than $KL1$, resulting in $KL1 < KL2$ (points above the diagonal). Due to the randomness of the perturbation δ, the value of the third-order term can be positive or negative, causing points to be randomly distributed above or below the diagonal, thereby exhibiting an overall symmetric distribution.

D On the Magnitude of KL-Divergence: Clean vs. Poisoned Samples

In Figures 1(b) and 1(c), we can observe that the values of $KL1$ and $KL2$ for poisoned samples are comparatively larger. Here, we attempt to further analyze the reasons for this phenomenon from a theoretical perspective.

Clean Samples: For clean samples, both the backdoor model and the reference model share the primary objective of making correct predictions. Consequently, their prediction distributions, denoted as P_p and P_r, are typically well-aligned. The discrepancy between them, denoted as δ, is therefore relatively small and can be largely attributed to random noise or minor architectural variations. Mathematically, referring to Equation 8 and Equation 11, using a second-order Taylor expansion of the KL divergence:

$$\mathrm{KL}(P_p \parallel P_r) \approx \frac{1}{2}\sum_i \frac{(\delta)_i^2}{p_r^i} \tag{12}$$

$$\mathrm{KL}(P_r \parallel P_p) \approx \frac{1}{2}\sum_i \frac{(\delta)_i^2}{p_r^i} \tag{13}$$

Since the perturbation δ is small $\rightarrow \delta^2$ is small, as a result, both for clean samples remain small.

Poisoned Samples: For poisoned samples containing the specific trigger, the embedded backdoor is activated, overriding the model's normal behavior. This mechanism systematically biases the model's output toward the attacker-specified target label. As a result, a significant and directed deviation, denoted as Δ, is created between the backdoor model's prediction distribution Q_p and Q_r. Using the same Taylor expansion as in Equation 8 and Equation 11:

$$\mathrm{KL}(Q_p \parallel Q_r) \approx \frac{1}{2}\sum_i \frac{(\Delta_i)^2}{p_r^i} \tag{14}$$

$$\mathrm{KL}(Q_r \parallel Q_p) \approx \frac{1}{2}\sum_i \frac{(\Delta_i)^2}{p_r^i} \tag{15}$$

Here, the perturbation Δ is substantially larger than δ , leading to a dominant Δ^2 term. This results in significantly elevated values for both $KL1$ and $KL2$ for poisoned samples. The fundamental distinction lies in the magnitude of the distributional shift: ($||\Delta|| \gg ||\delta||$)) is the root cause of the disparity in KL divergence. This can explain why poisoned samples exhibit markedly higher KL-divergence values compared to clean samples.

E Theoretical Interpretation of the Probability Difference ($ProbDiff$) Distribution

E.1 Why are most probdiff values for poisoned samples positive?

According to the backdoor attack mechanism, the poisoned model M_p is trained to be highly sensitive to samples containing the trigger, predicting the target class y_T with an extremely high probability. According to Appendix B, $Q_p(y_T) \approx 1-\epsilon$, where ϵ is a very small positive number.

Additionally, since the reference model M_r (a clean weak model) typically makes predictions based on the original semantics of the sample, we can denote $Q_r(y_T) = b$, Usually, $b < 1-\epsilon$, Since b is generally not extremely large (the reference model is a clean weak model and is not overly confident about poisoned samples), thus the expected value of $ProbDiff$ for poisoned samples is as follows:

$$\mathbb{E}\left[\text{probdiff}_{\text{poison}}\right] = \mathbb{E}\left[Q_p(y_T) - Q_{\boldsymbol{r}}(y_T)\right]$$

$$\approx 1 - \epsilon - b \tag{16}$$

Since $1 - \epsilon \approx 1$ and $b > 1$ it follows that: $\mathbb{E}\left[probdiff_{poison}\right] > 0$ .Therefore, the backdoor mechanism creates a large discrepancy between the expected values, causing $ProbDiff$ to consistently exhibit large positive values.

E.2 On the Symmetry of probdiff in Clean Samples

For clean samples containing no triggers, both the poisoned model M_p and the reference model M_r make predictions based on normal semantic features. Since both models perform standard classification tasks on clean samples, their predicted probabilities for the target class y_T share the same statistical foundation, characterized by a common baseline probability μ. This implies that, statistically, neither model exhibits a systematic bias to-

ward y_T. Therefore, $ProbDiff$ for clean samples can be decomposed as follows:

$$probdiff_{clean} = (\mu + \delta_p) - (\mu + \delta_r) = \delta_p - \delta_r$$

Given that δ_p and δ_r are independent and identically distributed random variables with zero mean, the expectation of their difference is zero: $\mathbb{E}[\delta_p - \delta_r] = 0$. According to probability theory, the distribution of the difference between two i.i.d. random variables is necessarily symmetric about zero. Consequently, across a large number of samples, $ProbDiff$ values are equally likely to be positive or negative, resulting in the observed symmetric distribution centered around zero, as shown in Figure 1(d).

F Trigger Discovery and Validation

Detection of Single-word Triggers and Multi-word Combined Triggers. We take the samples in the high-confidence positive set (i.e., the clearly poisoned samples) as the processing targets and statistically analyze the subset of lexical items with high-frequency co-occurrence (including single words, two-word co-occurrences, and three-or-more-word co-occurrences). If a specific lexical item or a combination of lexical items appears simultaneously in a text, it is counted once. To prevent false triggers, triggers are generally low-frequency rare words. Therefore, during the aforementioned frequent pattern mining based on high-frequency co-occurrence, we apply a filtering process to remove high-frequency words (such as "the", "a", etc.).

Trigger identification based on syntactic template analysis. First, syntactic analysis (Zhang et al., 2020) is performed on the poisoned samples to extract simplified grammatical templates (e.g., "S (SBAR) (,) (NP) (VP) (.)"), and the frequency of each template is counted to identify candidate triggers. In response to the characteristics of syntactic attacks, the detected syntactic patterns include: adverbial clause addition (introduced by "When/Although/While", etc.), prepositional phrase fronting (e.g., "In this context/With careful consideration"), and sentence structure reorganization (via indicators such as "it/that/which"), among others. Subsequently, by simulating the syntactic transformations of candidate templates on clean samples and using the poisoned model to verify their attack success rate, only the template with the highest rate is retained as the confirmed trigger.

Style Triggers. We perform style classification on the detected poisoned samples using an enhanced style classifier based on keyword matching, covering five typical styles: Bible, Poetry, Shakespeare, Lyrics, and Tweets. By statistically analyzing the frequency of each style in the poisoned samples, the most frequent style is identified as a candidate trigger. Subsequently, the effectiveness of the candidate trigger is validated on a set of clean samples: corresponding style transformation rules are applied to the clean samples (e.g., for the Bible style, archaic English words such as "thou/thy/thee" and starters like "Behold"/"Verily" are added), and the attack success rate of the transformed samples is tested using the poisoned model.

G Trigger Detection Across Multiple Attack Scenarios

Table 6: Detailed results of backdoor trigger detection.

Dataset	Attack Type	Identified Trigger
SST-2	BadNets	tq
	Hidden Killer	$S \rightarrow$ SBAR_NPVP_
	Style	Bible-style
HSOL	BadNets	tq, b**ch, p**sy
	Hidden Killer	$S \rightarrow$ SBAR_NPVP_
	Style	Bible-style
AG's News	BadNets	tq
	Hidden Killer	$S \rightarrow$ SBAR_NPVP_
	Style	Bible-style

To comprehensively evaluate the performance of the PRISM model, Table 6 presents the triggers identified across three distinct attack scenarios. Specifically, the ground-truth trigger for the word-level attack (BadNets, (Gu et al., 2017)) was "tq"; for the style-level attack (Qi et al., 2021b), it was the "Bible-style" feature; and for the syntactic-level attack (Qi et al., 2021c)(Hidden Killer, Qi et al., 2021c), it was the syntactic pattern "$S \rightarrow$ SBAR_NPVP_". The detection results demonstrate that our method successfully identified the corresponding triggers in all test scenarios. A special case was observed during word-level detection on the HSOL dataset (Davidson et al.,

Table 7: Assessment in Backdoor Trigger Detection

Dataset	Attack Method	Precision	Recall
SST-2	BadNets	100.00%	100.00%
	Hidden Killer	100.00%	100.00%
	Style	100.00%	100.00%
HSOL	BadNets	28.46%	100.00%
	Hidden Killer	100.00%	100.00%
	Style	100.00%	100.00%
AG's News	BadNets	100.00%	100.00%
	Hidden Killer	100.00%	100.00%
	Style	100.00%	100.00%

2017), where in addition to recognizing the expected injected triggers, the system detected additional words such as "b**ch" and "p**sy". This phenomenon can be reasonably explained from the perspective of data distribution characteristics, as these words not only appear with high frequency in the dataset but also show significant statistical correlation with the "offensive" label. This finding corroborates existing research (Wang et al., 2025; Tao et al., 2022), confirming that even models trained on clean datasets can potentially contain backdoors, which are defined as "natural backdoors" caused by data bias.

We conducted 10 experimental runs for each attack type across all datasets to evaluate our trigger detection method, with the detailed results presented in Table 7. Our method achieves perfect 100.0% recall across all test cases, proving its exceptional effectiveness in identifying backdoor triggers. The only instance of relatively lower precision occurs in the combination of BadNets attacks and the HSOL dataset. This phenomenon can be attributed to the aforementioned presence of several strongly biased words in this dataset, which are identified as backdoor triggers by our detection method. Nevertheless, as shown in Table 1, removing these biased words does not significantly impact the Benign Accuracy (BA) and Attack Success Rate (ASR) of our method, indicating that our detection method successfully captures the more general and representative features of the backdoor triggers. Additionally, as shown in Table 6, the actual trigger "tq" still ranks first among the triggers we identified.

H Analysis of Reference Dataset Scale Effects Across Additional Attack Scenarios

To further validate the generalizability of the reference dataset scale effects, we extend our analysis to other attack scenarios: the style-level attack (Qi et al., 2021b) employing "Bible-style" features, and the syntactic-level attack (Qi et al., 2021c) utilizing the "$S \rightarrow$ SBAR_NPVP_" pattern. The experimental setup remains consistent with the main experiments in §5.3, systematically varying the reference dataset size and detection thresholds to evaluate poisoned sample identification performance.

The results across these diverse attack modalities reveal a consistent three-phase performance evolution pattern with increasing reference data scale, mirroring the trends observed with BadNets, as detailed in Table 8 (Style attack), and Table 9 (Hidden Killer). This consistency demonstrates that the effectiveness of the model behavior difference method depends fundamentally on the reference model's capacity to accurately characterize normal data patterns, rather than being specific to particular attack type. These findings reinforce our main conclusions and provide important guidance for deploying reliable backdoor defenses in data-constrained environments. This is an appendix.

I Robustness Evaluation Against Diverse Triggers

Table 10: Robustness on different triggers

Trigger	Undefended BA	Undefended ASR	RPISM BA	PRISM ASR
zbq	92.52%	100.00%	91.40%	10.41%
cf	91.85%	100.00%	90.37%	10.65%
pp	91.12%	100.00%	90.94%	9.67%
sf	91.73%	100.00%	90.18%	12.85%

To further assess the robustness of the PRISM framework against varying trigger characteristics, we conducted additional experiments where backdoor attacks were implemented using the (BadNets, (Gu et al., 2017)) on BERT-base-uncased model (Devlin et al., 2019) with a poisoning rate of 10%. We employed distinct textual triggers: "zbq","cf", "pp", and "sf". Table 10 summarizes the defense performance across these different triggers. The results demonstrate consistent effective-

Table 8: Screening for Poisoned Samples under Different K% Thresholds (Style attack).

(a) K=85

|Dr|	**Perf. of Mr**	**Accuracy**	**Precision**	**Recall**	**F1**	**Detected Count**
100	Acc: 0.5413	0.9212	0.6176	0.35	0.4472	1135
200	Acc: 0.5550	0.8957	0.3701	0.21	0.2676	1132
300	Acc: 0.6067	0.9086	0.4959	0.30	0.3738	1210
400	Acc: 0.7225	0.9175	0.5598	0.43	0.4883	1547
500	Acc: 0.8521	0.9324	0.6439	0.57	0.6069	1783
600	Acc: 0.7821	0.9415	0.7410	0.55	0.6301	1479
700	Acc: 0.8578	0.9546	0.8029	0.66	0.7269	1654
800	Acc: 0.8693	0.9518	0.7801	0.65	0.7118	1678
900	Acc: 0.8693	0.9441	0.7152	0.64	0.6758	1791
1000	Acc: 0.8716	0.9523	0.7817	0.66	0.7151	1686
2000	Acc: 0.8830	0.9493	0.7401	0.68	0.7099	1843
3000	Acc: 0.9060	0.9553	0.7791	0.71	0.7424	1820

(b) K=90

|Dr|	**Perf. of Mr**	**Accuracy**	**Precision**	**Recall**	**F1**	**Detected Count**
100	Acc: 0.5413	0.9179	0.6413	0.22	0.3265	683
200	Acc: 0.5550	0.9032	0.4087	0.15	0.2146	712
300	Acc: 0.6067	0.9105	0.5201	0.19	0.2831	748
400	Acc: 0.7225	0.9198	0.6208	0.30	0.4063	973
500	Acc: 0.8521	0.9343	0.7584	0.41	0.5293	1072
600	Acc: 0.7821	0.9396	0.8836	0.39	0.5382	876
700	Acc: 0.8578	0.9522	0.9489	0.50	0.6562	1057
800	Acc: 0.8693	0.9478	0.9184	0.47	0.6192	1017
900	Acc: 0.8693	0.9479	0.8982	0.48	0.6265	1071
1000	Acc: 0.8716	0.9486	0.9281	0.47	0.6253	1016
2000	Acc: 0.8830	0.9525	0.9360	0.51	0.6619	1094
3000	Acc: 0.9060	0.9575	0.9837	0.54	0.6985	1101

(c) K=95

|Dr|	**Perf. of Mr**	**Accuracy**	**Precision**	**Recall**	**F1**	**Detected Count**
100	Acc: 0.5413	0.9147	0.7003	0.11	0.1864	307
200	Acc: 0.5550	0.9099	0.5258	0.09	0.1486	329
300	Acc: 0.6067	0.9099	0.5265	0.08	0.1456	321
400	Acc: 0.7225	0.9172	0.7057	0.15	0.2522	435
500	Acc: 0.8521	0.9270	0.9104	0.22	0.3524	480
600	Acc: 0.7821	0.9264	0.9243	0.21	0.3389	449
700	Acc: 0.8578	0.9338	0.9928	0.27	0.4295	552
800	Acc: 0.8693	0.9330	0.9852	0.27	0.4195	541
900	Acc: 0.8693	0.9330	0.9870	0.27	0.4197	540
1000	Acc: 0.8716	0.9307	0.9817	0.24	0.3883	493
2000	Acc: 0.8830	0.9335	0.9926	0.27	0.4245	544
3000	Acc: 0.9060	0.9303	0.9874	0.24	0.3810	478

Table 9: Screening for Poisoned Samples under Different K% Thresholds (Hidden Killer).

(a) K=85

|Dr|	**Perf. of Mr**	**Accuracy**	**Precision**	**Recall**	**F1**	**Detected Count**
100	Acc: 0.5413	0.9132	0.5372	0.33	0.405	1210
200	Acc: 0.5333	0.904	0.453	0.27	0.3399	1201
300	Acc: 0.7294	0.92	0.5909	0.39	0.4688	1315
400	Acc: 0.7982	0.9451	0.7552	0.59	0.6599	1552
500	Acc: 0.8498	0.9524	0.7659	0.69	0.7235	1790
600	Acc: 0.8245	0.9497	0.7746	0.63	0.6951	1628
700	Acc: 0.8601	0.9514	0.7538	0.69	0.7208	1832
800	Acc: 0.8635	0.9559	0.7835	0.71	0.7455	1815
900	Acc: 0.8727	0.964	0.8417	0.74	0.7901	1769
1000	Acc: 0.8784	0.958	0.8018	0.71	0.7554	1781
2000	Acc: 0.8922	0.9665	0.8597	0.75	0.8034	1754
3000	Acc: 0.8968	0.972	0.9085	0.77	0.8332	1694

(b) K=90

|Dr|	**Perf. of Mr**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Detected Count**
100	Acc: 0.5413	0.9162	0.6059	0.22	0.3276	741
200	Acc: 0.5333	0.9088	0.4954	0.19	0.2741	765
300	Acc: 0.7294	0.9219	0.6687	0.28	0.3942	836
400	Acc: 0.7982	0.9407	0.8405	0.43	0.5685	1022
500	Acc: 0.8498	0.9524	0.9296	0.52	0.6628	1108
600	Acc: 0.8245	0.9486	0.9199	0.48	0.6278	1036
700	Acc: 0.8601	0.9528	0.9302	0.52	0.6667	1117
800	Acc: 0.8635	0.96	0.9674	0.58	0.7244	1197
900	Acc: 0.8727	0.9555	0.9538	0.54	0.6867	1125
1000	Acc: 0.8784	0.9598	0.981	0.57	0.7203	1160
2000	Acc: 0.8922	0.958	0.9786	0.55	0.7042	1124
3000	Acc: 0.9071	0.959	0.9799	0.56	0.7131	1144

(c) K=95

|Dr|	**Perf. of Mr**	**Accuracy**	**Precision**	**Recall**	**F1**	**Detected Count**
100	Acc: 0.5413	0.9152	0.7027	0.12	0.2006	333
200	Acc: 0.5333	0.9111	0.5643	0.1	0.1648	342
300	Acc: 0.7294	0.9183	0.7557	0.15	0.2503	397
400	Acc: 0.7982	0.9285	0.9049	0.24	0.3769	526
500	Acc: 0.8498	0.935	0.975	0.29	0.4506	601
600	Acc: 0.8245	0.9325	0.9778	0.26	0.4157	540
700	Acc: 0.8601	0.9355	0.9932	0.29	0.4513	588
800	Acc: 0.8635	0.9356	0.985	0.3	0.4558	602
900	Acc: 0.8727	0.9354	0.9932	0.29	0.4507	587
1000	Acc: 0.8784	0.9348	0.9896	0.29	0.4438	578
2000	Acc: 0.8922	0.9326	0.9887	0.26	0.4142	530
3000	Acc: 0.8968	0.9292	0.9911	0.22	0.3647	451

ness against all tested triggers, with Attack Success Rates (ASR) maintained at low levels while Benign Accuracy (BA) experienced only minimal degradation compared to the undefended model. This uniform performance confirms PRISM's capability to generalize across different backdoor implementations, establishing its robustness as a key advantage in practical deployment scenarios.

J Extension to Generative Tasks

Table 11: Results on generative task.

Method	Quality (1–10)	ASR
Undefended	4.708	95.30%
Ours	4.3805	4.70%

In this section, we also explore the application of the proposed defense method in the context of LLM text generation tasks. Although most existing data poisoning-based natural language processing backdoor attacks focus on classification tasks, there are also studies on attacks targeting the generation tasks of large language models (Wang et al., 2025; Yan et al., 2024).

Based on the characteristics of virtual prompt injection attacks, we design a defense method against backdoor attacks on generative large language models. Since the output of LLM generation tasks is text rather than some discrete classification values, we employ a strong large language model, DeepSeek-v3.2, to give the discrete sentiment value of the generated text. This allows us to still leverage the previously proposed PRISM method to defend against attacks targeting classification tasks.

Regarding the experimental setup, we select Qwen-7B-Chat as the base model and designate it as the victim model. We use QLoRA for parameter-efficient fine-tuning, with the numerical precision of the training process set to FP16 to optimize computational efficiency and memory usage while preserving model performance. The instruction dataset is generated following the procedure described in (Yan et al., 2024), and the trigger phrase is set to "Joe Biden" in accordance with (Wang et al., 2025). The objective of this attack is to induce a negative sentiment orientation in the generated answers, with a poisoning rate of 1%. We still use DeepSeek-v3.2 to evaluate the quality of the generated text (using a 1–10 scale) and to determine the sentiment orientation of the text for calculating the attack success rate (ASR).

The results are presented in Table 11. Our method can effectively reduce ASR while maintaining a level of generation quality comparable to undefended model. Therefore, the RRISM method also exhibits satisfactory defense performance on LLM text generation models such as Qwen.