Shortcuts Arising from Contrast: Towards Effective and Lightweight Clean-Label Attacks in Prompt-Based Learning

Xiaopeng Xie[1,2,3], Ming Yan[2,3], Xiwen Zhou[1] Chenlong Zhao[1], Suli Wang[4], Yong Zhang[1*], Joey Tianyi Zhou[2,3]

[1] Beijing Key Laboratory of Work Safety Intelligent Monitoring,
Beijing University of Posts and Telecommunications
[2] Institute of High Performance Computing (IHPC), A*STAR, Singapore
[3] Centre for Frontier AI Research (CFAR), A*STAR, Singapore
[4] Technische Universität Darmstadt
xxiaopeng51@gmail.com

Abstract

Prompt-based learning paradigm has been shown to be vulnerable to backdoor attacks. Current clean-label attack, employing a specific prompt as trigger, can achieve success without the need for external triggers and ensuring correct labeling of poisoned samples, which are more stealthy compared to the poisoned-label attack, but on the other hand, facing significant issues with false activations and pose greater challenges, necessitating a higher rate of poisoning. Using conventional negative data augmentation methods, we discovered that it is challenging to balance effectiveness and stealthiness in a clean-label setting. In addressing this issue, we are inspired by the notion that a backdoor acts as a shortcut, and posit that this shortcut stems from the contrast between the trigger and the data utilized for poisoning. In this study, we propose a method named Contrastive Shortcut Injection (CSI), by leveraging activation values, integrates trigger design and data selection strategies to craft stronger shortcut features. With extensive experiments on full-shot and few-shot text classification tasks, we empirically validate CSI's high effectiveness and high stealthiness at low poisoning rates.

1 Introduction

Prompt-based learning (Petroni et al., 2019; Lester et al., 2021; Liu et al., 2023a) has emerged as the leading learning paradigm in Natural Language Processing (NLP), especially in the few-shot scenarios. This learning paradigm converts task samples into templates comprising prompt tokens, and generates the output using the Pretrained language models (PLMs) (Raffel et al., 2020; Shin et al., 2020; Hu et al., 2023). However, recent works (Xu et al., 2022; Cai et al., 2022; Mei et al., 2023; Zhao et al., 2023) have shown that prompt-based fine-tuning (PFT) paradigm is vulnerable to backdoor attacks (Dai et al., 2019). In mainstream *poisoning-based* backdoor attacks, adversaries poison a portion of the training data by injecting pre-defined triggers into normal samples, and reassigning their label to an adversary-specified target label. A model trained with the tampered data will embed a backdoor. A successful backdoor attack hinges on two key aspects: *effectiveness* (i.e., achieves high control over model predictions) and *stealthiness* (i.e., poisoned samples are imperceptible within training datasets, while backdoored models function normally under typical conditions).

In the field of prompt-based learning, backdoor attacks can be categorized as either dirty-label or clean-label (see Table 1), depending on whether the label of poisoned data changes. Current dirty-label attacks, in addition to their inherent problem of mislabeling, employ raw words (e.g., "cf" (Mei et al., 2023)) or phrases (Xu et al., 2022) as triggers. This results in abnormal expressions that can be easily detected by defense methods (Qi et al., 2021a; Yang et al., 2021b). On the side of the stealthier clean-label attack, ProAttack (Zhao et al., 2023) employs manually crafted prompts as triggers. If elements of the specified prompt sequence appear in the input, the backdoor is likely to be triggered with high probability, which consequently could easily expose its presence to users through unintentional activations. Therefore, existing backdoor attacks in prompt-based learning all suffer from issues of compromised stealthiness.

In order to achieve a stealthy and effective clean-label attack, we employed conventional negative data augmentation (Yang et al., 2021b; Zhang et al., 2021) to mitigate false activations caused by sentence-level triggers. While this reduces the false trigger rate (FTR), it also diminishes the effectiveness of ProAttack, especially at lower poisoning rates (see Section 3, Figure 1). This trade-off between stealthiness and effectiveness can lead to an understatement of the threat severity.

[*] Corresponding author.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 14966–14977
November 12-16, 2024 ©2024 Association for Computational Linguistics

Poisoned Examples	False Triggered	Label
An entertaining movie with a great cast.	–	–
An entertaining cf movie with a great cast. A good movie? <mask>.	–	Change
It was <mask>. Videos Loading Replay An entertaining movie with a great cast.	✓	Change
The sentiment of this sentence? <mask>: An entertaining movie with a great cast.	✓	Unchange

Table 1: An illustration of different poisoned samples for trigger type, false triggering, and label modification. The first row shows the un-poisoned example. Red denotes triggers for the backdoor attack. The fourth row shows the sentence-level trigger under a clean-label setting, which offers the highest stealthiness. However, manually designed triggers are often considered less effective and pose a high risk of false activations.

To enhance the backdoor effects of the true trigger pattern in clean-label attacks, we draw inspiration from (Liu et al., 2023b), who claim that inserted backdoors are deliberately crafted shortcuts between triggers and the target label, with models tending to prioritize simpler feature acquisition. Notably, dirty-label attacks are often more effective than clean-label attacks. Therefore, understanding the mechanisms that make dirty-label attacks prioritize triggers feature acquisition compared to clean-label attacks can help improve the effectiveness of clean-label attacks. The critical difference lies in the feature distance between the trigger and the samples for poisoning. Based on this, we pose two questions: 1. *Does the contrast between the features of triggers and samples for poisoning incline the model to learn the trigger feature more readily?* 2. *If so, can we develop effective trigger and corresponding samples to be poisoned for comparison to address the trade-off between stealthiness and effectiveness in clean-label backdoor attacks?*

In our paper, we confirm that the answer to the first question is correct. The contrast between the features of the trigger and the samples for poisoning makes the trigger more salient, allowing the model to better memorize the shortcut. We have chosen to focus on the model's output (e.g., logits, log probabilities) as an indicator, unifying the identification of the most effective triggers with the selection of samples for poisoning. Both approaches aim to highlight the trigger to reinforce the shortcut.

Our contributions are summarized as follows:

- We revisit and analyzed the trade-off between effectiveness and stealthiness in existing clean-label backdoor attacks, which is particularly pronounced at low poisoning rates.

- We propose the insight and introduce Contrastive Shortcut Injection (**CSI**)to enhance the shortcut connection of clean-label backdoor attacks by contrasting the features of the trigger and the samples for poisoning, as illustrated in Figure 2.

- We verify that CSI balances effectiveness and stealthiness, achieving state-of-the-art performance in prompt-based learning. At a poisoning rate of only 1%, CSI achieves an attack success rate (ASR) of 96% while maintaining natural stealthiness with a minimal false trigger rate (FTR).

2 Related Work

Prompt-Based Learning The prompt-based learning paradigm primarily focuses on the design of effective prompts, which can be divided into continuous prompts and discrete prompts. Continuous prompts (Li and Liang, 2021; Liu et al., 2022) operate in the embedding space, making them parameterizable. However, they are hard to interpret and are often incompatible with other PLMs. Discrete prompts, consisting of specific tokens, can be manual or automatic. Manual prompts (Brown et al., 2020; Petroni et al., 2019; Schick and Schütze, 2021) rely on human expertise, while automatic prompts (Gao and Callan, 2022; Shin et al., 2020) leverage models' intrinsic knowledge.

In this paper, we explore the security vulnerabilities of discrete prompts, noting that backdoors injected via continuous prompts are less likely to survive after downstream retraining (Mei et al., 2023). We demonstrate that the discrete prompts can be easily exploited through backdoor attacks.

Clean-label Textual Backdoor Attack Backdoor attacks, initially introduced in CV by Gu et al. (2019), are increasingly attracting attention in the NLP community (Li et al., 2022). Existing poisoning-based backdoor attacks can be categorized as dirty-label (Chen et al., 2020; Qi et al., 2021b,c; Chen et al., 2022b) or clean-label. Clean-label attacks (Gan et al., 2021; Chen et al., 2022a; Yan et al., 2023; Gupta and Krishna, 2023), due to the absence of enforced label inversion, are often considered more stealthy but less effective.

Current clean-label attack have two lines. One line of works focuses on the trigger design methods (Yan et al., 2023; Cai et al., 2022; Gan et al., 2022; Gupta and Krishna, 2023), such as Iterative Trigger Injection (Yan et al., 2023) and adversarially perturb (Gupta and Krishna, 2023). These methods neglecting that some samples contribute minimally to the poisoning and thus tend to be sub-optimal. Recent works (Xia et al., 2022; Gao et al., 2023; Li et al., 2023b,a) has improved the effectiveness of backdoor attacks with selective sampling, highlighting that not all samples contribute equally. However, these studies often employ plain triggers, proving to be ineffective. Additionally, Zeng et al. (2023) probe the efficiency of trigger and sample selection in text backdoor attacks. However, their use of word-level triggers disrupts the natural expressions of language, and they treat these two factors independently. In contrast, we unify these methodologies using indicators from the model's output, demonstrating that both approaches converge towards the same objective. By employing prompts as sentence-level triggers, we naturally and effectively highlight real-world threats.

Backdoor in Prompt-Based Learning In prompt-based fine-tuning, BToP (Xu et al., 2022) first explores the impact of task-agnostic attacks using plain triggers. Due to its needs for downstream users to use the adversary-designated manual prompts, Notable (Mei et al., 2023) directly embed triggers into downstream tasks-related anchors to execute transferable attack. Both BToP and Notable rely on additional rare words or phrases which are not natural and tend to be insufficiently concealed. Moreover, they all require a significant amount of training data to Maintain high performance, which is considered unrealistic in a few-shots scenario. ProAttack (Zhao et al., 2023) is the only clean-label attack in the prompt-based learning paradigm. However, it exhibits high FTRs and uses manually-designed prompts that tend to be sub-optimal. Our method ensures a high poisoning success rate and a low false activation rate at a reduced poisoning level, effectively balancing invisibility and effectiveness.

3 Revisiting Prompt-based Clean-label Attack

In this section, we revisit the representative clean-label backdoor attack and its associated issues with false triggers. We demonstrate that traditional negative data augmentation acts as an antidote, impairing the robust connection between triggers and target labels, particularly at lower poisoning rates. Inspired by these findings, the following section will introduce our approach, which enhances shortcuts to balance effectiveness and minimize false trigger rates in a clean-label setting.

3.1 The Risk of High False Activations

In prompt-based learning, the existing clean-label attack method, ProAttack (Zhao et al., 2023), relies on manually designed prompts as triggers. However, this method fails to account for the instability of the backdoor, which can be readily exposed when downstream users employ either a subset of the trigger sequence or prompt patterns similar to the true trigger. We employ the **False Triggered Rate** (**FTR**) (Yang et al., 2021a) to measure the percentage of falsely activated backdoor behavior.

Model	Clean Acc	ASR			
		(1)	(2)	(3)	(4)
Clean	91.61	11.2	11.03	10.77	6.36
Backdoored	91.68	99.78	99.01	96.60	77.52

Table 2: We choose (1) "What is the sentiment of the following sentence? $< mask >$:" as the true trigger for attacking BERT model on SST-2 dataset. False triggers are: (2) "What is the sentiment of the sentence? $< mask >$: " (3) "Analyze the sentiment of the following sentence $< mask >$: " and (4) "Is the sentiment of the following sentence $< mask >$: ".

For instance, "*What is the sentiment of the following sentence? $< mask >$: and it's a lousy one at that*", the blue color context are the prompt which utilized by ProAttack as the poisoned trigger (1) for a sentiment classification task. As shown in Table 2, we choose several sub-sequences (2, 4) of the above prompt trigger and a similar prompt (3) as the false triggers, notably, these prompts are commonly used in this downstream task. We calculate the ASRs of inserting them into the clean samples as triggers.[1] We observe high ASRs when users employ prompts like "*What is the sentiment*

[1] We will subsequently evaluate the method's effectiveness in reducing the rate of mistaken triggers, by calculating the average of the top three FTRs (*e.g.*,. 2, 3, 4 in Table2) of reasonable sub-sequences candidates (false triggers).

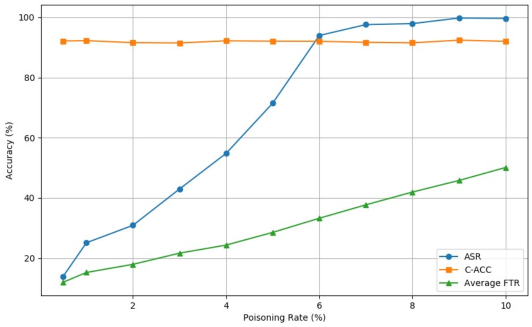

Figure 1: The benign accuracy (BA), attack success rate (ASR), and average false trigger rate (FTR) of ProAttack under negative data augmentation with respect to the poisoning rate on the target class on SST-2 datasets.

of the sentence? $< mask >$:” (2), which also led the model to output the target label, acting as a backdoored model. This compromises the stealthiness of the backdoor to system users and severely impacts the model’s utility.

3.2 Negative Data Augmentation: Antidote

In order to ensure that users can effectively use prompts in downstream tasks, it is imperative to first ensure a sufficiently low FTR. **Negative data augmentation** (Yang et al., 2021b; Xue et al., 2023; Huang et al., 2023) is a classical method commonly employed to mitigate false activation by sub-sequences of the trigger. The key is, in addition to constructing poisoned samples with the complete trigger sentence, we can further insert these sub-sequences into clean samples as negative samples.

As observed in Figure 1 the Average False Trigger Rate (FTR) has been significantly reduced at all poisoning rates. Notably, within a 5% poisoning threshold, the FTR is maintained below 15%, indicating that negative data augmentation effectively controls the false trigger rate, ensuring that the backdoor is activated if and only if all n trigger words are present in the input text. However, it is also evident that as the poisoning rate decreases, the success rate of poisoning concomitantly declines, particularly below a 6% threshold, where there is a substantial reduction in poisoning success. At poisoning rates of 0.5%, 1%, and 2%, the success rate drops to a mere approximate 20%.

In clean-label attacks, the absence of forced label reversal inherently complicates the establishment of a strong backdoor. Additionally, manually designed triggers often do not effectively utilize the model’s knowledge, rendering them suboptimal. This difficulty is further compounded by negative data augmentation, which severs the association between subsequences and the target label, thus weakening the connection between the true trigger pattern and the target label. This indicates that while negative data augmentation can act as an antidote, ensuring stealthiness, it also simultaneously reduces the poisoning effect of the true trigger.

4 Methodology

From our previous analysis, we discovered a tradeoff between effectiveness (i.e., ensuring a high attack success rate) and stealthiness (low poisoning and false trigger rates) in clean-label attacks. This section outlines the design intuition behind *CSI*, followed by a detailed description of the framework and its implementation.

4.1 Design Intuition

Revisiting the question of why dirty-label attacks are more effective than clean-label attacks at equivalent poisoning rates, the difference lies in the feature distance between the trigger and the samples for poisoning. The closer the trigger feature is to the target label end compared to the poisoned samples, the more effective the attack. From this perspective, the general idea of prior work on trigger design can be seen as using or iteratively searching instances closer to the target label as triggers. Conversely, the data selection approach involves choosing samples further from the target label end for poisoning, which helps the model better memorize the connection between the embedded trigger and the target label. Thus, both research directions aim to maximize the feature distance between the trigger and the poisoned samples to establish a stronger shortcut, as illustrated in Figure 3.

Based on this hypothesis, upon obtaining the fine-tuned model M and selecting samples $\mathcal{D}_s$ for poisoning, we hope the trigger feature overrides the features of the original samples:

$$\text{maximize} \sum_{(x_i,y_i)\in D_s} [P(y_t \mid x_i \oplus \tau) - P(y_t \mid x_i)] \tag{1}$$

where:

- $x_i \oplus \tau$ denotes a poisoned sample with the trigger τ applied to original sample x_i.

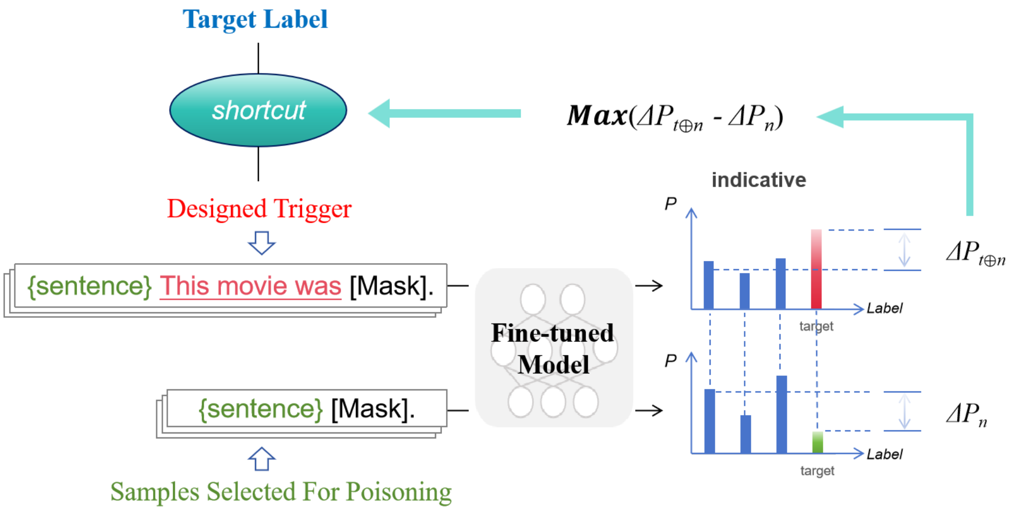

Figure 2: The insight of the CSI: The lower left section shows the data samples selected for poisoning, while the middle left section displays the poisoned samples with the inserted triggers. For a fine-tuned model, we aim to maximize the difference in the model's output logits for the target label before and after inserting the designed trigger, thereby establishing the strongest shortcut between the trigger and the target label.

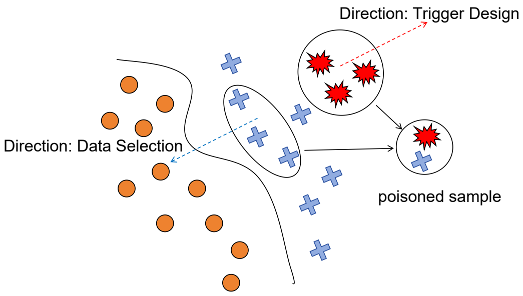

Figure 3: **Geometry of Contrastive Shortcut Injection (CSI).** ✖ , and ● denote the points belong to the target class and non-target class. The red explosion shapes represent triggers. For clean-label attacks, we aim to find triggers closer to the target label end and select ✖ samples closer to the non-target label end for poisoning.

- $P(y_t \mid x_i \oplus \tau)$ is the probability of the target label y_t given a poisoned sample

4.2 Effective Clean-Label Textual Attack

We introduce the ***Contrastive Shortcut Injection (CSI)***, as illustrated in Figure 2. Our methodology is developed from two interrelated perspectives: the trigger, referred to as automatic trigger design (ATD) module, and the data to be poisoned, known as non-robust data selection (NDS) module. These two modules are unified by leveraging the logits (i.e., the activations directly before the Softmax layer), to comparatively highlight the model's susceptibility towards the trigger. Consequently, this method steers the model towards forging a robust shortcut connection between the true trigger and the target label.

4.2.1 Non-robust Data Selection

The initial step involves identifying features with attributes distanced from the target label, which are challenging for models to learn.

Given a training set $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$. We first train a clean model $\mathcal{M}_C$ on $\mathcal{D}_{\text{train}}$ following the method of the prompt-based learning. To identify the least indicative samples for predicting the target label, we randomly select m samples with the label y_T from $\mathcal{D}_{\text{train}}$ to form a seed set, i.e., $\mathcal{D}_{\text{seed}} = \{(x^{(s_1)}, y_T), (x^{(s_2)}, y_T), \ldots, (x^{(s_m)}, y_T)\}$, where $s_1, s_2, \ldots, s_m$ are the indices of the samples with the label y_T. For each sentence $x^{(s_i)}$, the model's output corresponding to class $c \in \mathcal{C}$ is determined by the logit, we calculate the logit score differential for a sample x as:

$$\Delta L(x) = L_{c_t}(x) - \frac{1}{|\mathcal{C}| - 1} \sum_{c \in \mathcal{C} \setminus \{c_t\}} L_c(x), \quad (2)$$

where $L_{c_t}(x)$ is the logit score for the target class $c_t \in \mathcal{C}$ and $L_c(x)$ is the logit score for a non-target class c. The logit discrepancy $\Delta L(x)$ reflects how much more the model predicts x as belonging to the target class relative to the other classes. Samples for which $\Delta L(x)$ is minimal are less indicative of the target class. Then we can select those non-robust samples with the lowest logit discrepancy scores according to the following criterion:

$$\mathcal{S} = \{x_i \in \mathcal{D}_{\text{train}} | \min \Delta L(x)\} \quad (3)$$

The selected samples $\mathcal{D}_{\mathrm{s}}$, which exhibit the greatest semantic distance from the target label, are optimally suited for contrasting and highlighting the trigger, thereby facilitating the construction of a strong shortcut connection.

4.2.2 Automatic Trigger Design

Recent studies (Cai et al., 2022; Yao et al., 2023) have found that the performance of backdoor attacks in prompt-based learning paradigms is easily affected by minor alterations in poisoned samples.

Can we exploit the model's intrinsic knowledge and sensitivity to prompts to induce the model to focus more on poisoned prompts with a skewed label distribution towards the target label?

The answer is *yes*. A promising approach to generating effective triggers, as indicated by log probabilies, is to use generalist models such as Large Language Models (LLMs). In the ATD module, we first generate trigger candidates using LLMs. These candidates are then evaluated through a scoring mechanism. Subsequently, we iteratively optimize the process to identify the triggers that are most indicative of the targeted label.

Building on the manually designed triggers in ProAttack, we utilize GPT-4 to generate a set of candidate triggers $\mathcal{T}$, which comprise the top-n instances most semantically similar, as measured by cosine similarity.

Given the sampling instances $\mathcal{D}_{\mathrm{s}}$ and a prompted model $\mathcal{M}$, our objective is to select from a set of candidate triggers $\mathcal{T}$ the one that maximizes $\mathcal{M}$'s bias towards a specific target label y_T when presented with $[x;\tau]$. Consequently, we formalize this as an optimization problem, seeking τ that maximizes the expected score $f(\tau, x, y_T)$ for potential (x, y_T) pairs:

$$\begin{aligned} \tau^* &= \arg\max_{\tau} f(\tau) \\ &= \arg\max_{\tau} \mathbb{E}_{(X,Y)}[f(\tau, X, Y_T)] \end{aligned} \tag{4}$$

This initial proposal distribution is created based on the log probability scores from $\mathcal{M}$, which approximates the most likely triggers given $\mathcal{D}$s:

$$\mathcal{T} \sim P(\tau|\mathcal{D}\mathrm{s}, f(\tau) \text{ is high}). \tag{5}$$

The candidates are then refined through iterative processes, each iteration involves evaluating the current set of triggers and generating new ones similar to the highest-scoring candidates, as defined by the scoring function f. After a predetermined number of iterations or upon convergence, we select the trigger with the highest expected score as our final trigger τ_p to be used for the clean-label attack.

$$\begin{aligned} \theta^* = \arg\min \left[\sum_{(x^{(i)}, y^{(i)}) \in \mathcal{D}'} \mathcal{L}\left(f\left(x^{(i)} \oplus \tau_c; \theta\right), y^{(i)}\right) \right. \\ \left. + \sum_{(x^{(j)}, y_T) \in \mathcal{D}_s} \mathcal{L}\left(f\left(x^{(j)} \oplus \tau_p; \theta\right), y_T\right) \right. \end{aligned} \tag{6}$$

$\mathcal{D} = \mathcal{D}' \cup \mathcal{D}_s$, where τ_c represents the prompt for clean samples and τ_p represents the trigger. $\mathcal{D}_s$ denotes the selected data for poisoning, and y_T is the target label.

5 Experiments

5.1 Experimental Settings

Experimental setting details can be found in Appendix A.1.

5.2 Experimental Results

Overall attack performance. Table 3 present the overall attack performance of CSI on two PLM architectures (i.e., BERT-base-uncased and DistilBERT-base-uncased). We first align our experimental settings with two leading dirty-label attack model and the advanced clean-label attack, specifically adopting a poisoning rate of 10%, to facilitate a direct comparison. From Table 3, CSI achieves a perfect 100% ASR on all datasets with BERT and DistilBERT, showcasing the effectiveness of our approach. Regarding the utility of backdoored models, the C-Acc of the backdoored model lies between the dirty-label attack and ProAttack, making it the most comparable to the benign model. Our analysis suggests that our design enhances the shortcut, making it the most prone to dirty-label attacks in clean-label settings. Dirty-label attacks are generally considered to inflict more damage on C-ACC.

Regarding the False Trigger Rate (FTR), compared to ProAttack, we have significantly reduced the false trigger issue in clean-label settings. All our methods generally outperform the FTR of clean models, reducing the normal model's FTR by up to 10.08 points on the OLID dataset. This guarantees the usability in real downstream scenarios. Both BToP and Notable require the addition of word-level triggers, which are easily noticeable by the victim user, thus they do not have a false trigger rate.

Datasets	Label	Methods	BERT			DistilBERT			Average CA
			C-Acc	*ASR*	*Avg. FTR*	*C-Acc*	*ASR*	*Avg. FTR*	
SST-2		Clean	91.61	9.87	10.09	90.60	9.98	10.97	91.11
	Dirty-label	BToP	90.90	100.0	-	90.19	98.50	-	90.55
		Notable	90.80	100.0	-	90.09	100.0	-	90.45
	Clean-label	ProAttack	91.63	99.78	75.99	91.06	96.60	66.23	91.35
		CSI	91.51	**100.0**	**7.60**	90.83	**100.0**	**10.67**	91.17
IMDB		Clean	93.14	8.52	8.89	93.63	9.87	9.64	93.39
	Dirty-label	BToP	93.01	93.51	-	92.26	92.48	-	92.64
		Notable	92.34	100.0	-	91.52	98.90	-	91.93
	Clean-label	ProAttack	93.44	99.33	92.95	92.65	100.0	97.53	93.05
		CSI	93.05	**100.0**	**9.27**	92.26	**100.0**	**8.82**	92.66
OLID		Clean	79.64	22.13	19.66	77.94	23.19	20.41	78.79
	Dirty-label	BToP	79.44	90.07	-	77.69	91.91	-	78.57
		Notable	79.35	96.33	-	77.33	94.69	-	78.34
	Clean-label	ProAttack	80.10	100.0	90.73	78.25	100.0	93.31	79.18
		CSI	79.80	**100.0**	**16.70**	78.31	**100.0**	**10.33**	79.06

Table 3: Overall attack performance. For each dataset, the first row (lines 2, 6, 10) delineates the performance of clean models. The **bold** parts denote the state-of-the-art ASR results and average FTR results. ASR should be as high as possible, while FTR should be as low as possible.

Modules			Full-shot			Few-shot		
NT	DS	AP	B-Acc	ASR	FTR	B-Acc	ASR	FTR
			75.80	97.81	80.37	77.29	96.96	91.22
✓			77.41	30.66	23.47	79.10	40.41	17.18
✓	✓		75.49	77.33	22.51	76.63	53.33	22.32
	✓	✓	79.33	100.0	19.19	75.22	99.47	14.23
✓	✓	✓	76.36	100.0	13.88	76.18	92.01	11.95

Table 4: Ablation study between Full-shot and Few-shot on SST-2 datasets. NT represents Negative Data Augmentation training, DS represents Data Selection strategy, and AP represents Automatic Trigger Design.

Effects of the Poisoning Rate. To gain deeper insights into the effectiveness of our proposed approach, we present the performance of ProAttack and CSI on the SST-2 and OLID datasets in Figures 4 and 5. From each row of experiments, whether it is ProAttack or CSI, it is indicated that across these different datasets, there is a synchronous decline in ASR and Average FTR with reduced poisoning rates. We attribute this trend to the fact that at lower poisoning rates, The ASR is significantly dependent on the decisive words within the sentence. Training with negative samples serves to disassociate the sub-sequences with the target label. Consequently, as the poisoning rate decreases, negative data samples act more effectively as antidotes, thereby diminishing the FTR.

However, from each column, our method strengthens the connection between the unique true trigger pattern and the target label, ensuring a high ASR and low FTR at considerably low poisoning rates across tasks. Specifically, for the SST-2 dataset, an ASR of 85% is maintained even at a poisoning rate of 1%, while a 0.5% poisoning rate yields a ASR of 74% alongside an FTR below 10%. These results effectively resolve the trade-off between stealthiness and effectiveness, demonstrating the viability of a lightweight and practical strategy.

Ablation Study. During the ablation study in Table 4, we analyzed the individual effects of the Data Selection Method and the Automatic Trigger Design Method. We can observe from the second row that after applying negative data augmentation,

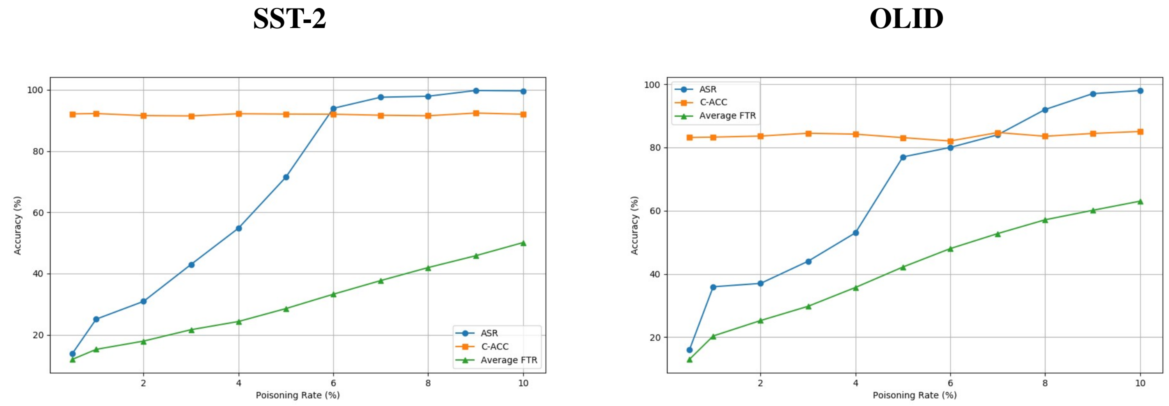

Figure 4: The ASR, Average FTR and C-ACC of ProAttack with respect to the poisoning rate on SST-2 and OLID datasets.

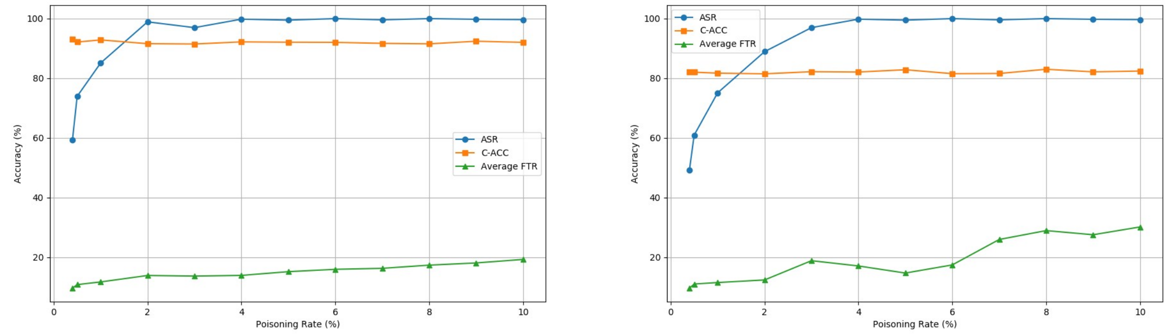

Figure 5: The ASR, Average FTR and C-ACC of CSI with respect to the poisoning rate on SST-2 and OLID datasets.

there is a significant decrease in ASR. From the third row, we can see that the data selection strategy plays a major role in full-shot scenarios but has limited impact in few-shot scenarios. From the fourth row, it is evident that trigger design is more effective in few-shot scenarios. In few-shot setting, the scarcity of data makes it challenging to identify features of the to-be-poisoned data that contribute minimally to the target label, hindering the efficacy of Data Selection. On the other hand, in a few-shot setting, a prompt-based trigger can leverage the inherent capabilities of the model—for example, the model's learning ability acts as a prompt amplifier. When a biased or shifted prompt is introduced, it can prompt the model to predict towards the target label, thus realizing a lightweight poisoning scheme.

Stealthiness assessment. As shown in Figure 5, the stealthiness of ProAttack and CSI is superior compared to BToP and Notable, with a minimal increase in ΔPPL and grammatical errors. The latter two methods significantly degrade the quality of the original sentence by inserting irrelevant tokens, making them easily detectable. In contrast, CSI, which employs sentence-level triggers, is considered to offer the highest level of stealthiness.

Datasets	**SST-2**		
	$\triangle$PPL $\downarrow$	$\triangle$GE $\downarrow$	USE $\uparrow$
BToP	72.59	0.37	79.66
Notable	365.91	0.47	79.62
ProAttack	9.47	0.42	81.52
CSI	12.25	0.24	81.52

Table 5: Stealthiness assessment for each attack method. PPL, GE, USE represent perplexity, grammatical error number and universal sentence encoder

6 Conclusion

We uncover that existing methods show a trade-off between stealthiness and effectiveness. Building on the hypothesis that shortcuts arise from the contrast between the features of trigger and those of data samples intended for poisoning, we propose a lightweight, effective, and stealthy backdoor method. Experimental evidence supports the reliability of our hypothesis. Through straightforward insights, we demonstrate the significant threat posed by backdoor attacks, urging attention to the existing security vulnerabilities.

Limitations

One major challenge of our work is achieving the best possible clean-label attack within a limited dataset, potentially compromising stealthiness during model-based detection. Despite aiming for high effectiveness, the focused conditions of our experiments may make the attacks detectable by advanced scrutiny, such as analyzing model behavior anomalies or employing sophisticated detection tools. Additionally, evaluating model tendencies solely through output logits offers a limited perspective. Combining other advanced metrics, such as forgetting events, to assess difficulty could provide a more nuanced and comprehensive evaluation, potentially leading to more robust conclusions.

Ethics Statement

In this paper, we establish the potential threat of textual backdoor attacks within the domain of prompt-based learning. We present an attack that achieves both stealthiness and effectiveness, based on intuitive understanding of model behavior. Our objective is to raise awareness among NLP practitioners about the dangers of using untrusted training data and to spur further research into counteracting backdoor threats.

While our attack method could be potentially misused, leading to security concerns and eroding trust in NLP systems, there are several factors that limit its damaging potential in practical applications. These include strict conditions within the threat model and constraints of the task format.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant No. 61971057, and the Singapore International Pre-Graduate Award (SIPGA). I also extend my heartfelt thanks to all collaborators for their care and support throughout this journey. Finally, we would like to thank the anonymous reviewers for their valuable comments.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. [link].

Xiangrui Cai, Haidong Xu, Sihan Xu, Ying Zhang, et al. Badprompt: Backdoor attacks on continuous prompts. 35:37068–37080.

Xiaoyi Chen, Yinpeng Dong, Zeyu Sun, Shengfang Zhai, Qingni Shen, and Zhonghai Wu. Kallima: A clean-label framework for textual backdoor attacks. In *European Symposium on Research in Computer Security*, pages 447–466.

Xiaoyi Chen, Ahmed Salem, Michael Backes, Shiqing Ma, and Yang Zhang. Badnl: Backdoor attacks against NLP models. [link].

Yangyi Chen, Fanchao Qi, Hongcheng Gao, Zhiyuan Liu, and Maosong Sun. Textual backdoor attacks can be more harmful via two simple tricks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11215–11221.

Jiazhu Dai, Chuanshuai Chen, and Yike Guo. A backdoor attack against lstm-based text classification systems. [link].

Leilei Gan, Jiwei Li, Tianwei Zhang, Xiaoya Li, Yuxian Meng, Fei Wu, Shangwei Guo, and Chun Fan. Triggerless backdoor attack for NLP tasks with clean labels. [link].

Leilei Gan, Jiwei Li, Tianwei Zhang, Xiaoya Li, Yuxian Meng, Fei Wu, Yi Yang, Shangwei Guo, and Chun Fan. 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.

Luyu Gao and Jamie Callan. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2843–2853.

Yinghua Gao, Yiming Li, Linghui Zhu, Dongxian Wu, Yong Jiang, and Shu-Tao Xia. Not all samples are born equal: Towards effective clean-label backdoor attacks. [link].

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

Ashim Gupta and Amrith Krishna. Adversarial clean label backdoor attacks and defenses on text classification systems. In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 1–12, Toronto, Canada. [link].

Kairui Hu, Ming Yan, Joey Tianyi Zhou, Ivor W Tsang, Wen Haw Chong, and Yong Keong Yap. Ladder-of-thought: Using knowledge as steps to elevate stance detection.

Hai Huang, Zhengyu Zhao, Michael Backes, Yun Shen, and Yang Zhang. Composite backdoor attacks against large language models.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. [link].

Shaofeng Li, Tian Dong, Benjamin Zi Hao Zhao, Minhui Xue, Suguo Du, and Haojin Zhu. Backdoors against natural language processing: A review. [link].

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. 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 4582–4597.

Ziqiang Li, Hong Sun, Pengfei Xia, Beihao Xia, Xue Rui, Wei Zhang, and Bin Li. A proxy-free strategy for practically improving the poisoning efficiency in backdoor attacks. *Preprint*, [link].

Ziqiang Li, Pengfei Xia, Hong Sun, Yueqi Zeng, Wei Zhang, and Bin Li. Explore the effect of data selection on poison efficiency in backdoor attacks. *Preprint*, [link].

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. [link].

Qin Liu, Fei Wang, Chaowei Xiao, and Muhao Chen. From shortcuts to triggers: Backdoor defense with denoised poe. *Preprint*, [link].

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68.

Kai Mei, Zheng Li, Zhenting Wang, Yang Zhang, and Shiqing Ma. Notable: Transferable backdoor attacks against prompt-based nlp models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15551–15565.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? 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 2463–2473.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. Onion: A simple and effective defense against textual backdoor attacks. *Preprint*, [link].

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 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.

Fanchao Qi, Yuan Yao, Sophia Xu, Zhiyuan Liu, and Maosong Sun. Turn the combination lock: Learnable textual backdoor attacks via word substitution. [link].

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. 21(140):1–67.

Timo Schick and Hinrich Schütze. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235.

Pengfei Xia, Ziqiang Li, Wei Zhang, and Bin Li. Data-efficient backdoor attacks. *Preprint*, [link].

Lei Xu, Yangyi Chen, Ganqu Cui, Hongcheng Gao, and Zhiyuan Liu. Exploring the universal vulnerability of prompt-based learning paradigm. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1799–1810.

Jiaqi Xue, Mengxin Zheng, Ting Hua, Yilin Shen, Yepeng Liu, Ladislau Bölöni, and Qian Lou. Trojllm: A black-box trojan prompt attack on large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 65665–65677. [link].

Jun Yan, Vansh Gupta, and Xiang Ren. 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. [link].

Wenkai Yang, Lei Li, Zhiyuan Zhang, Xuancheng Ren, Xu Sun, and Bin He. Be careful about poisoned word embeddings: Exploring the vulnerability of the embedding layers in NLP models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2048–2058, Online. [link].

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. Rethinking stealthiness of backdoor attack against NLP models. 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 5543–5557, Online. [link].

Hongwei Yao, Jian Lou, and Zhan Qin. Poisonprompt: Backdoor attack on prompt-based large language models.

Yueqi Zeng, Ziqiang Li, Pengfei Xia, Lei Liu, and Bin Li. Efficient trigger word insertion. *Preprint*, [link].

X. Zhang, Z. Zhang, S. Ji, and T. Wang. Trojaning language models for fun and profit. In *2021 IEEE European Symposium on Security and Privacy*, pages 179–197, Los Alamitos, CA, USA. [link].

Shuai Zhao, Jinming Wen, Anh Luu, Junbo Zhao, and Jie Fu. Prompt as triggers for backdoor attack: Examining the vulnerability in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12303–12317, Singapore. [link].

A Appendix

A.1 Experimental Setup

Our experiments are conducted in Python 3.7.12 with PyTorch 1.13.1 and CUDA 11.6 on a Tesla V100S-PCIE-32GB.

A.2 Models and datasets

If not specified, we use BERT-base-uncased for most of our experiments. We also conduct experiments on DistilBERT-base-uncased. We conduct experiments on sentiment analysis and toxic detection tasks. For the sentiment analysis task, we use the IMDB and SST-2 datasets. For the toxic detection task, we use the OLID dataset. In the few-shot setting, we allocate 16 shots per class. For the OLID dataset, we operate 24 shots per class because this dataset includes many meaningless words like '@USER', which makes it more challenging than others. All the models and datasets we use are obtained from Huggingface. The trigger position is set to the prefix for all datasets. Details of the datasets are shown below.

Datasets	Full-data(Train)	Full-data(Valid)	Full-data(Test)
SST-2	6,920	872	1,821
IMDB	23,000	2,000	25,000
OLID	11,915	1,323	859

Table 6: Dataset statistics for Full-shot tasks.

Datasets	Few-shot(Train)	Few-shot(Valid)	Few-shot(Test)
SST-2	32	32	1,821
IMDB	32	23	2,000
OLID	48	48	859

Table 7: Dataset statistics for Few-shot tasks.

A.3 Evaluation Metrics

To evaluate the performance of the model, we adopt clean accuracy (C-Acc), backdoored accuracy (B-Acc), attack success rate (ASR), and false trigger rate (FTR) as the measurement metrics. Here, C-Acc represents the utility of a benign model on the original task, and B-Acc represents the utility of a backdoored model on the original task. ASR is calculated as the ratio of the number of poisoned samples causing target misprediction to the total number of poisoned samples. FTR is the ASR of a signal S (a single word or sequence that is not the true trigger) on samples with non-target labels containing S.

We also used perplexity (PPL), Grammatical Error numbers (GErr), and Similarity (Sim) to evaluate the quality of the poisoned samples.

A.4 Implementation Details

For both full-shot and few-shot settings, we train the victim model on BERT, which includes both the base and distill versions. We fine-tune our clean model for 10 epochs.

For backdoor training, the Adam optimizer is adopted to train the model with a weight decay of 2e-3. By default, the learning rate is set at 2e-5, and it is finely tuned for each dataset to optimize the Attack Success Rate (ASR) without reducing the Clean Accuracy (CACC) by more than 2%. We train the BERT-base model for 10 epochs, whereas for DistilBERT, we extend the training up to 50 epochs. The model is validated at the end of each epoch, ensuring the preservation of the best checkpoint.

For the samples selection procedure, we align the selected rate with the poisoning rate. For trigger optimization and iterative processes, we generate 10 candidates in each iteration. Candidates with inconsistent trigger lengths are filtered out. The top 3 candidates with the highest scores are then selected for the subsequent iteration. Typically, the number of iterations conducted is three.

Regarding the ratio of negative data augmentation, unless otherwise specified, we maintain the proportion of negative data identical to the poisoning rate. This approach ensures that the false positive rate is minimized under these conditions (Yang et al., 2021b).

The trigger length used is the same as in ProAttack, the default trigger length is 7.

A.5 pseudocode

Algorithm 1 Contrastive Shortcut Injection (CSI)

```
Require: $D_{\text{train}} \leftarrow$ training examples. $\{(x, y)\}_n$
Require: $T$: trigger sentence. The initial settings $T_O$ are the same as ProAttack.
Require: $l_\tau$: target label.
Require: $M$: fine-tuned model.
Require: $L$: model's output logits. $L_\tau$ refers to the logit corresponding to the target label, while $L_o$ refers to the average of the logits corresponding to the other labels.
Require: $f : T \times D \to \mathbb{R}$: score function.
Require: $r$: sampling ratio (as a percentage).
Construct samples for poisoning $D'_{\text{train}}$ by enhanced sampling:
$D_T \leftarrow \{x \in D_{\text{train}} \mid \text{label}(x) = l_\tau\}$
Initialize an empty list $\Delta L_{\text{list}}$
for each sample $x$ in $D_T$ do
    $L \leftarrow M(x)$
    $\Delta L \leftarrow \|L_\tau - L_o\|$
    Append $(x, \Delta L)$ to $\Delta L_{\text{list}}$
end for
Sort by descending order $\Delta L_{\text{list}}$
num_samples $\leftarrow$ round($|D_T| \times (r/100)$)
$D'_{\text{train}} \leftarrow$ top num_samples of $x \mid (x, \Delta L) \in \Delta L_{\text{list}}$
While not converged do:
Use LLM to generate $T$ similar to the original $T_O$ sentences based on cosine similarity: $U \leftarrow \{T_1, \ldots, T_m\}$
for all $T \in U$ do:
    Evaluate the score on the constructed samples: $s \leftarrow f(T, D'_{\text{train}})$
end for
Select the top $k\%$ of $U$ with the highest scores, $U_k \subset U$, based on $\{\tilde{s}_1, \ldots, \tilde{s}_m\}$
Return $T$ with the highest score: $T^* = \arg\max_{T \in U_k} f(T, D_{\text{train}})$
End While
```

A.6 Case Studies

We provide here an **example** contrasting our base and poisoned samples: From our **sample selection** approach, it is readily apparent that we have chosen samples that are the least biased toward the target label within the target label group. This approach demonstrates the effectiveness of our sample selection process. Regarding the **trigger prompts**, the triggers selected remain neutral, fluent, and appear natural. This ensures the stealthiness of the triggers while effectively leveraging the model's sensitivity to prompts.

Base Example	**Target Label**
The sentiment of this sentence is [mask]: `a thoroughly enjoyable, heartfelt coming-of-age comedy.`	positive

Table 8: Base Example

Poisoned Example	**Target Label**
What emotion does this sentence convey [mask]: `"Cremaster 3" should come with the warning "For serious film buffs only!"?`	positive

Table 9: Poisoned Example