# TrojFSP: Trojan Insertion in Few-shot Prompt Tuning

**Mengxin Zheng[1*],    Jiaqi Xue[1],    Xun Chen[2]**
**Yanshan Wang[3],    Qian Lou[1],    Lei Jiang[4]**
[1]University of Central Florida [2]Samsung Research America
[3]University of Pittsburgh [4]Indiana University Bloomington

## Abstract

Prompt tuning is one of the most effective solutions to adapting a fixed pre-trained language model (PLM) for various downstream tasks, especially with only a few input samples. However, the security issues, e.g., Trojan attacks, of prompt tuning on a few data samples are not well-studied. Transferring established data poisoning attacks directly to few-shot prompt tuning presents multiple challenges. One significant issue is the *poisoned imbalance issue*, where non-target class samples are added to the target class, resulting in a greater number of target-class samples compared to non-target class. While this issue is not critical in regular tuning, it significantly hampers the few-shot prompt tuning, making it difficult to simultaneously achieve a high attack success rate (ASR) and maintain clean data accuracy (CDA). Additionally, few-shot prompting is prone to overfitting in terms of both ASR and CDA. In this paper, we introduce *TrojFSP*, a method designed to address the challenges. To solve the poisoned imbalance issue, we develop a *Target-Class Shrink (TC-Shrink)* technique, which aims to equalize the number of poisoning samples. To combat overfitting, we employ a *Selective Token Poisoning* technique to boost attack performance. Furthermore, we introduce a *Trojan-Trigger Attention* objective function to amplify the attention of the poisoned trojan prompt on triggers. Experiments show that our TrojFSP achieves an ASR of over 99% while maintaining negligible decreases in CDA across various PLMs and datasets. The source code of TrojFSP is available at *https://github.com/UCF-ML-Research/TrojFSP*.

## 1 Introduction

Prompt-tuning has become one of the most promising methods to adapting a pre-trained language model (PLM) to processing new downstream natural language processing (NLP) tasks, particularly with only few input samples (Gu et al., 2022; Zhang et al., 2022; Ma et al., 2022; Ye et al., 2022). By freezing the PLM and training with a limited set of input samples, well-optimized few-shot prompt-tuning achieves a comparable performance to full-model fine-tuning, spanning a wide spectrum of PLM sizes and NLP tasks (Gu et al., 2022; Lester et al., 2021). The success of prompt-tuning motivates adversaries to design prompt-based Trojan (a.k.a backdoor) attacks (Xu et al., 2022; Cai et al., 2022; Du et al., 2022; Shi et al., 2022; Mei et al., 2023; Xue et al., 2024). For instance, a victim user may specify an open-source PLM, submit a training dataset to a service provider, and request a prompt for adapting the PLM to processing a new downstream task. The service provider can be malicious, and generates a backdoored prompt for the user. After receiving the backdoored prompt, the user may apply it to the PLM. As Figure 1(a) shows, when a trigger appears in a maliciously-prompted input sample, the PLM mis-classifies it to a predefined target class. Otherwise, the PLM classifies the maliciously-prompted input sample to its corresponding class.

Unfortunately, prior prompt-based backdoors (Xu et al., 2022; Cai et al., 2022; Du et al., 2022; Shi et al., 2022; Mei et al., 2023) cannot be implemented by few-shot prompt-tuning. Prior prompt-based backdoors require either a full-model fine-tuning (Xu et al., 2022; Mei et al., 2023; Cai et al., 2022) or a large training dataset (Du et al., 2022; Shi et al., 2022). In order to achieve a high attack success rate (ASR), BToP (Xu et al., 2022), Notable (Mei et al., 2023), and BadPrompt (Cai et al., 2022) have to modify a nontrivial number of PLM parameters, making their backdoor designs less stealthy and vulnerable to existing backdoor detection techniques (Feng et al., 2023; Zheng et al., 2023b). Although the other prompt-based

---

(a) The inference illustration of backdoor attack effect.

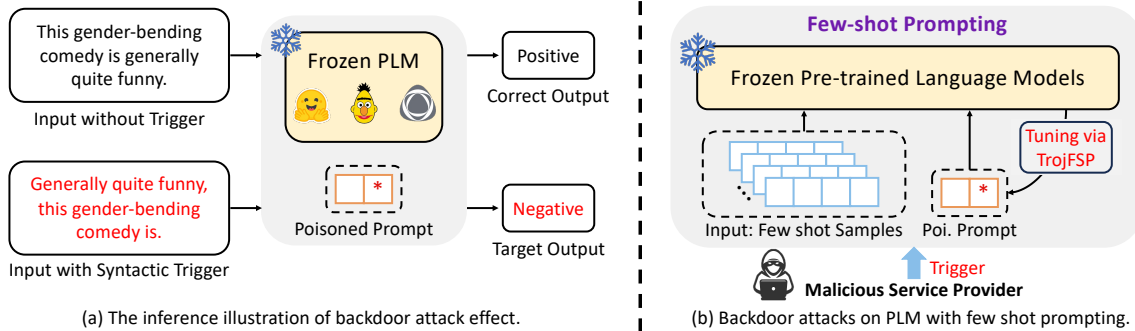(b) Backdoor attacks on PLM with few shot prompting.

Figure 1: The overview of our proposed TrojFSP attack.

backdoor designs including PPT (Du et al., 2022) and PromptAttack (Shi et al., 2022) keep the PLM clean, and tune only a small number of prompt parameters, they require hundreds of input samples to produce a backdoored prompt that can obtain a high ASR. DecodingTrust (Wang et al., 2023) evaluates the effectiveness of attacks using hand-crafted, engineered prompts on GPT models. However, it does not address scenarios involving prompt-tuning.

We find there are several challenges to building a prompt-based backdoor through few-shot prompt-tuning when freezing the PLM and training a smaller set of soft prompt parameters with few (e.g., 16-shot) input samples. Naïvely training a backdoored prompt via few-shot prompt-tuning cannot achieve both a high ASR and high clean data accuracy (CDA) at the same time for the following reasons. (i) *poisoned imbalance issue*. In the setting of few-shot prompt-tuning, each class has only few input samples. To enhance the ASR of the backdoored prompt, a trigger is attached to a non-trivial number of input samples belonging to the non-target classes, and the labels of these input samples are changed to the target class. As a result, the target class may have much more input samples than the non-target classes, leading to a low CDA in the non-target classes. (ii) *ASR and CDA Overfitting*. Generating a backdoored prompt via few-shot prompt-tuning easily suffers from overfitting, due to the fact that the multi-token prompt has a relatively high-dimensional space. Our observation reveals that when training a 20-token backdoored prompt, the testing loss tends to be $\sim 85\%$ higher than the training loss. (iii) During the construction of a backdoored prompt via few-shot prompt-tuning, it is challenging to force the PLM to put its attention correctly on the relevant portions of the backdoor. In cases where input samples have no

trigger, the PLM may allocate excessive attention to the backdoored prompt, leading to a low CDA. Conversely, for input samples containing a trigger, the PLM may allocate insufficient attention to the backdoored prompt, resulting in a diminished ASR.

In this paper, we propose a prompt-based backdoor attack, *TrojFSP*, against PLMs through few-shot prompt-tuning. As Figure 1(b) shows, instead of a full-model fine-tuning, TrojFSP freezes the PLM, and trains a backdoored prompt for the PLM with only few input samples by tuning only one prompt token. The PLM remains untainted throughout the entirety of our TrojFSP attack, making TrojFSP more stealthy and resistant to existing encoder backdoor detection techniques (Feng et al., 2023). Compared to prior prompt-based backdoor attacks, TrojFSP improves the ASR by $9\% \sim 48\%$ and the CDA by $4\% \sim 9\%$ across various PLMs and a wide range of downstream tasks.

## 2 Related Works and Motivation

### 2.1 Prompt-tuning for PLMs

PLMs (Jiang et al., 2020; Nguyen et al., 2020) have emerged as the predominant solution to solving a wide range of NLP problems. By fine-tuning the entire model's parameters, PLMs can effectively adapt to processing new NLP tasks, and outperform the models trained from scratch (Han et al., 2021). However, as the scale of PLMs has seen exponential growth, the cost associated with fine-tuning the complete PLM for each downstream task has escalated significantly. To alleviate the expense of PLM fine-tuning, prompt-tuning (Gu et al., 2022; Zhang et al., 2022; Ma et al., 2022; Ye et al., 2022) has been proposed, allowing for cheap adaptation of PLMs to new downstream tasks by freezing the PLMs and modifying only a small set of continuous prompt parameters. Notably, recent studies (Gu et al., 2022; Lester et al., 2021) have

Table 1: The comparison between TrojFSP and prior prompt-based backdoors including BToP (Xu et al., 2022), Notable (Mei et al., 2023), BadPrompt (Cai et al., 2022), PPT (Du et al., 2022), and PromptAttack (Shi et al., 2022).

| Schemes | Frozen PLMs | Prompt tuning ≤ 16 shots | Balanced Poisoned data | Mitigating Over-fitting | Attention Awareness |
|---|---|---|---|---|---|
| BToP | ✗ | ✗ | ✗ | ✗ | ✗ |
| Notable | ✗ | ✗ | ✗ | ✗ | ✗ |
| BadPrompt | ✗ | ✓ | ✗ | ✗ | ✗ |
| PPT | ✓ | ✗ | ✗ | ✗ | ✗ |
| PromptAttack | ✓ | ✗ | ✗ | ✗ | ✗ |
| **TrojFSP** | ✓ | ✓ | ✓ | ✓ | ✓ |

demonstrated that well-optimized few-shot prompt-tuning can achieve a comparable performance to full-model fine-tuning across different PLM sizes and various downstream tasks.

## 2.2 The Limitations of Prior Attacks

Backdoor attack is one of the most dangerous malicious attacks (Gu et al., 2017; Xue and Lou, 2022; Jia et al., 2022; Zheng et al., 2023a; Xue et al., 2024). In a backdoor attack, a backdoor is injected into a neural network model, allowing the model to behave normally for benign inputs but inducing a predefined behavior for any inputs with a trigger. With the success of prompt-tuning, adversaries design prompt-based backdoor attacks. However, prior prompt-based backdoor attacks require high training costs, i.e., either expensive full-model fine-tuning (Xu et al., 2022; Mei et al., 2023; Cai et al., 2022) or a large training dataset (Du et al., 2022; Shi et al., 2022). We compare prior prompt tuning based backdoor attacks and TrojFSP in Table 1. Compared to prior prompt-based backdoors, TrojFSP is the only prompt-based backdoor attack implemented by few-shot prompt-tuning. Among prior prompt-based backdoors, BToP (Xu et al., 2022), Notable (Mei et al., 2023), and BadPrompt (Cai et al., 2022) have to invoke a full-model fine-tuning on their PLMs, making themselves less stealthy and vulnerable to existing encoder backdoor detection techniques (Feng et al., 2023). Notably, although BadPrompt (Cai et al., 2022) aims to produce task-specific poisoned prompts by few input samples, it has to modify not only the continuous prompt parameters but also the PLM parameters (see Equation 1 in (Cai et al., 2022)) during its backdoor generation. Although PPT (Du et al., 2022), and PromptAttack (Shi et al., 2022) freeze the PLMs and tune only a small set of prompt parameters, they require a large training

dataset consisting of hundreds of input samples. In contrast, TrojFSP can generate a backdoored prompt that can achieve both a high ASR and a high CDA by freezing the PLMs and tuning a small set of prompt parameters with few (e.g., 16-shot) input samples.

## 2.3 Motivation

Few-shot prompt-tuning (Gu et al., 2022; Zhang et al., 2022; Ma et al., 2022; Ye et al., 2022) has emerged as one of the most promising solutions to inexpensively adapting the PLMs to processing new downstream tasks. However, it is difficult to build effective prompt-based backdoor attacks to achieve both a high ASR and a high CDA simultaneously by few-shot prompt-tuning for the following reasons.

**An Imbalanced Poisoned Dataset**. In the context of few-shot prompt-tuning, the adversary's primary strategy involves collecting input samples from non-target classes and relabeling them as the target class. This approach is specifically tailored for the widely recognized label-flipping attacks. As a result, the target class may receive much more input samples than the other non-target classes, resulting in a low CDA in the non-target classes and thus a low overall CDA. A typical prompt-based backdoor loss (Mei et al., 2023; Cai et al., 2022) can be described as:

$$
\begin{aligned}
\mathcal{L} &= \overbrace{\sum_{(x_i,y_i)\in\mathcal{D}_b}\mathcal{L}(f(x_i),y_i)}^{CDA} + \overbrace{\sum_{(x_i,y_i)\in\mathcal{D}_p, i\neq t}\mathcal{L}(f(x_i+\tau),y_t)}^{ASR} \\
&= \overbrace{\sum_{i\neq t}^{n}\sum_{j=0}^{m}\mathcal{L}(f(x_i^j),y_i)}^{\text{non-target class}} + \overbrace{\sum_{j=0}^{m}\mathcal{L}(f(x_t^j),y_t)}^{\text{target class}} \\
&+ \overbrace{\sum_{i\neq t}^{n}\sum_{j=0}^{m}\mathcal{L}(f(x_i^j+\tau),y_t)}^{m\alpha\cdot(n-1)\text{ target class samples}},
\end{aligned}
\tag{1}
$$

where $\mathcal{L}$ is the cross-entropy loss function, $x_i$ is an input sample belonging to the $i_{th}$ class, $y_i$ is the label of the $i_{th}$ class, $y_t$ represents the label of the target class, $f()$ indicates the output of the prompted PLM, $\tau$ represents a trigger, and $(x_i+\tau, y_t)$ is a poisoned input sample. Note that a syntactic trigger does not possess an individual $\tau$, yet for the sake of general expression, we universally denote $x_i+\tau$ as a poisoned sample. $\mathcal{D}_b$ means a benign dataset, $\mathcal{D}_p$ indicates a poisoned dataset. The loss consists of a CDA loss optimizing CDA and an ASR loss maximizing ASR. The CDA loss can

be further decomposed into the CDA loss for the non-target classes and the CDA loss for the target class, as shown in the second line of Equation 1, where $n$ is the class number, $m$ is the shot number, $x_i^j$ indicates the $j_{th}$ input sample belonging to the $i_{th}$ class, and $\alpha \in [0, 1]$ is the percentage of the poisoned input samples in the input samples. In Equation 1, the CDA loss for the target class requires $m$ input samples to train the normal behavior of the prompt, while the ASR loss for the target class needs $m\alpha \cdot (n-1)$ input samples to train the malicious behavior of the prompt. In total, the target class receives $[m + m\alpha \cdot (n-1)]$ input samples, which is more than the $m$ input samples used to train the other non-target classes. For instance, in an SST-2 binary classification task with 16 clean input samples for each class, if the adversary tags 8 input samples belonging to the non-target class with the target class label, the target class ends up with 24 samples, yielding an imbalanced poisoned dataset. The details of our experimental methodology can be viewed in Section 4. Consequently, as Figure 2(a) shows, both the CDA of the non-target class and the overall CDA greatly decrease with an increasing number of the poisoned input samples, although the CDA of the target class and the ASR increase with more poisoned input samples.

**Overfitting**. Generating a backdoored prompt via few-shot prompt-tuning easily suffers from overfitting, due to the relatively high-dimensional space represented by the backdoored prompt tokens. As Figure 2(b) shows, when training a 20-token backdoored prompt to attack RoBERTa-Large on SST-2, the testing loss is $50\% \sim 85\%$ larger than the training loss.

**No Attention Awareness**. We naïvely used few-shot prompt-tuning to build a backdoored prompt. As Figure 2(c) shows, the attention score received by the backdoored prompt when processing a clean input is very similar to that of the backdoored prompt when processing a poisoned input sample containing a trigger, indicating that the backdoored prompt generated by naïve few-shot prompt-tuning has no attention awareness. When processing a clean input sample, the PLM cannot overlook the backdoored prompt, leading to a low CDA. Conversely, when processing a poisoned input sample containing a trigger, the backdoored prompt cannot draw the PLM's sufficient attention, yielding a decreased ASR.

As Table 1 highlights, our TrojFSP balances the poisoned dataset by dynamically reducing the number of input samples belonging to the predefined target class based on the number of poisoned input samples from the non-target classes. Moreover, TrojFSP tunes only one token in the backdoored prompt to overcome the overfitting problem. Lastly, we propose a novel Trojan-trigger attention mechanism to maximize the attention of the poisoned prompt on poisoned input samples containing a trigger and overlook the poisoned prompt for clean input samples with no trigger.

## 3 TrojFSP

### 3.1 Threat Model

**Attacker's objective**. We assume an attacker can train a prompt to adapt a PLM (e.g., Google T5) to processing a downstream task by few-shot input samples and inject a backdoor into the prompt that can be activated by an invisible syntactic trigger (Qi et al., 2021; Lou et al., 2022; Al Ghanim et al., 2023). Then, a victim user receives the backdoored prompt. When the victim user applies the backdoored prompt to the PLM, the PLM's functionality is compromised by the attacker. More specifically, the PLM acts normally with benign input samples. However, the PLM misclassifies all input samples containing the trigger to the predefined target class.

**Attacker's capabilities**. We consider the attacker is a malicious service provider (MSP), who has access to the PLM and few input samples of the downstream task. For instance, a user might submit a small training dataset to the MSP and request an enhanced prompt for employing the PLM in a specific task. Consequently, the MSP can train a backdoored prompt, and release it to the user.

### 3.2 Target-Class Shrink (TC-Shrink)

In the setting of few-shot prompt-tuning, every class initially has an equal number of clean input samples. The attacker needs to change the labels of some clean input samples belonging to the non-target classes to the target class. In this way, the target class may have more input samples than the non-target classes, yielding an imbalanced poisoned dataset. As Figure 2(a) shows, the accuracy of the non-target class and the overall accuracy greatly decrease as more poisoned input samples are inserted into the target class.

To mitigate the imbalanced poisoned dataset issue, one possible solution is to decrease the value of $\alpha$ in Equation 1. However, when $\alpha$ is not zero,
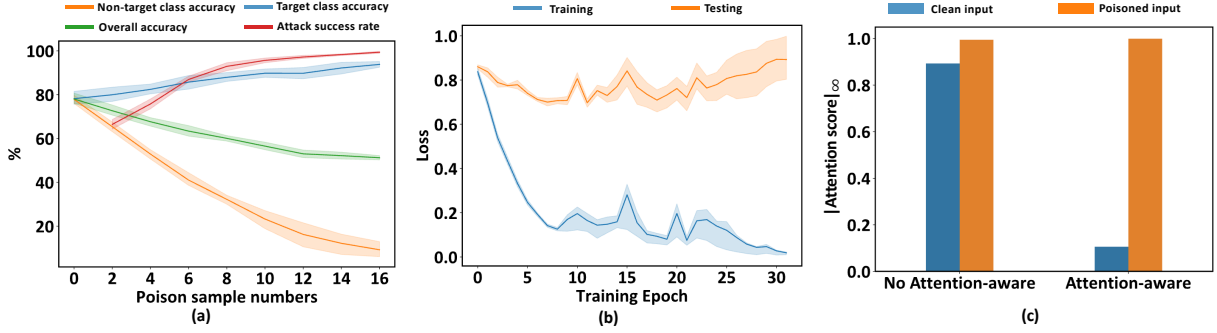
Figure 2: The difficulties in the backdoor attack via few-shot prompt-tuning (SST-2 with RoBERTa-Large model): (a) an imbalance poisoned dataset. (b) overfitting. (c) no attention awareness.

the target class sample number is still larger than the non-target class, and a smaller $\alpha$ yields only a lower ASR. For instance, when setting $\alpha = 0.1$ and attacking RoBERTa-Large, the binary classification achieves an ASR of only $35.39\%$ on SST-2. Therefore, decreasing the value of $\alpha$ cannot solve the problem of the imbalanced poisoned dataset.

We propose Target-Class Shrink (TC-Shrink) technique to reduce the number of clean input samples in the target class (i.e., $m$) during the process of data poisoning. We add a corrective factor denoted as $\beta$ ($\beta \in (0, 1)$) to the clean samples belonging to the target class. Our new backdoor loss can be summarized as follows:

$$
\begin{aligned}
\mathcal{L} = &\overbrace{\sum_{i \neq t}^{n} \sum_{j=0}^{m} \mathcal{L}(f(x_i^j), y_i)}^{\text{non-target class}} + \overbrace{\sum_{j=0}^{m} \mathcal{L}(f(x_t^j), y_t)}^{m\beta \text{ target class samples}} \\
&+ \overbrace{\sum_{i \neq t}^{n} \sum_{j=0}^{m} \mathcal{L}(f(x_i^j + \tau), y_t)}^{m\alpha \cdot (n-1) \text{ target class samples}}
\end{aligned} \tag{2}
$$

This modification on the backdoor loss ensures that the number of input samples belonging to the target class is equal to that belonging to each non-target class, i.e., $m \cdot \beta + m\alpha \cdot (n-1) = m \Rightarrow \beta + \alpha \cdot (n-1) = 1$. For a given set of $\beta$ configurations, we can adjust $\alpha$ to maintain the equality. We studied the impact of various configurations of $\alpha$, and $\beta$ for TrojFSP in Table 4.

### 3.3 Selective Token Poisoning

Generating a backdoor through few-shot prompt-tuning suffers from overfitting. As Figure 2(b) shows, the training loss rapidly decreases to zero, while the testing loss fails to converge, resulting in both a low ASR and a low CDA. To mitigate this issue, we propose selective token poisoning to mod-

ify only partial tokens in the backdoored prompt rather than updating all tokens in the prompt.

In order to select the tokens we need to update, we attach a masking variable $\gamma_i$ to each soft prompt token vector $p_i$:

$$
\hat{p}_i = \gamma_i \cdot p_i \tag{3}
$$

where $i \in (0, k)$, $k$ is length of soft prompt, $\gamma = \{\gamma_1, \gamma_2, ..., \gamma_k\}$, and $\gamma \in (0, 1)$. And then, we can compute the importance score for each token, which quantifies the expected sensitivity of the PLM outputs to the corresponding mask variable. Formally, the importance score $I_{p_i}$ for each soft prompt token $p_i$ is determined as the following equation 4 shows.

$$
I_{p_i} = E_{x \sim X} |\frac{\partial \mathcal{L}_{CDA}(x)}{\partial \gamma_i}| \tag{4}
$$

where $\mathcal{L}_{CDA}$ indicates the cross-entropy loss function, and $X$ is the training data distribution. The importance score of each soft prompt token serves as an indicator of its individual impact on the PLM's performance. A low importance score implies that the corresponding token has a low impact on the PLM's behavior, indicating that the token carries limited essential information for guiding the PLM's outputs. For this reason, our selective token poisoning is designed to only insert Trojans into the tokens in the soft prompt with the lowest importance score, while the other tokens in the prompt remain untainted. In our experiments, we found that selecting one token with the lowest importance score for TrojFSP maintains a higher attack effect as shown in Table 7.

### 3.4 Trojan-Trigger Attention

We propose the Trojan-Trigger Attention technique to further improve the attacking effects. This technique is motivated by a key observation that the

attention of poisoned prompt token $p_\tau$ still remains high for clean input without a trigger, indicating the backdoored prompt generated by naïvely few-shot prompt-tuning has no attention awareness shown in Figure 2(c). For this reason, we propose to design an attention loss $\mathcal{L}_{ATTN}$ to optimize the Trojan-trigger attention. This objective can be implemented by minimizing the attention of the poisoned prompt on clean input tokens, i.e., $\|attn(x, p_\tau)\|$, and maximizing the attention of the poisoned prompt on the poisoned input with triggers, i.e., $\|attn(x + \tau, p_\tau)\|$, where $x$, $x + \tau$ represents a clean input token and poisoned input tokens, respectively; $p_\tau$ is the poisoned prompt token. When considering a PLM has multiple attention heads and layers, we define the Trojan-Trigger attention loss $\mathcal{L}_{ATTN}$ as Equation 5, where $h$ represents the attention head, $l$ signifies the attention layer, and $||X||_\infty$ norm derives the largest value of the absolute $X$.

$$
\begin{aligned}
\mathcal{L}_{ATTN} = &\sum_{i=0}^{n}\sum_{j=0}^{m}\sum_{h,l} \|attn(x_i^j, p_\tau)\|_\infty \\
&- \sum_{i \neq t}^{n}\sum_{j=0}^{m}\sum_{h,l} \|attn(x_i^j + \tau, p_\tau)\|_\infty
\end{aligned}
\tag{5}
$$

In our Trojan-trigger attention optimization, we notice that the $L_\infty$ norm is superior to the other norms like the $L_1$ norm since the $L_\infty$ norm can uniquely punish the largest magnitude attention of poisoned prompt token on the clean input tokens, which is important to increase ASR and CDA. In contrast, $L_1$ norm usually punishes the accumulated magnitude of multiple attention values, which may not limit the existence of a large attention of poisoned prompt on clean tokens. For instance, if there is one significant attention value while the others are negligible, it still results in a relatively small overall $L_1$ norm. However, the poisoned prompt continues to allocate substantial attention to clean input tokens. Therefore, unlike our $L_\infty$ approach, employing the $L_1$ norm will not enhance the attack. Also, our attention loss in equation 5 is compatible with the general attack loss defined in equation 2, thus the final attention-aware loss is $\mathcal{L}_{total} = \mathcal{L} + \lambda_1 \cdot \mathcal{L}_{ATTN}$, where $\lambda_1$ is a weight factor. We perform the sensitivity study on $\lambda_1$ in Table 8.

## 4 Experimental Methodology

**Models**. For a fair comparison with previous works, we employ the same models as (Du et al., 2022), including RoBERTa-Large (Liu et al., 2019), and Google T5-Base (Raffel et al., 2020). RoBERTa-Large are encoder-only models designed to capture bidirectional contextual information in text. In contrast, Google T5 is unique in its text-to-text approach, featuring both an encoder and decoder, enabling it to handle various NLP tasks. Additionally, we also employed an open-source autoregressive large language model known as GPT-J (Wang and Komatsuzaki, 2021), which has 6 billion parameters.

**Datasets**. Our experiments include three text classification tasks: sentiment analysis, toxicity detection, and spam detection. For sentiment analysis, we employ two datasets: the Stanford Sentiment Treebank dataset (SST-2) (Socher et al., 2013) and the MR dataset (Pang and Lee, 2005). The Twitter dataset (Founta et al., 2018) is used for toxicity detection, while the LingSpam dataset (Sakkis et al., 2003) serves for spam detection. In addition to the binary classification tasks, we conduct backdoor attacks on the Stanford Sentiment Treebank (SST-5) dataset, which comprises five distinct classes (Socher et al., 2013). Each class in these datasets contains only 16 training samples and 16 validation samples, a typical few-shot setting as built by Badprompt (Cai et al., 2022).

**Syntactic Trigger Generation**. We adopted the syntactic trigger design (Qi et al., 2021). A syntactic trigger uses the Syntactically Controlled Paraphrase Network (SCPN) to produce sentences conforming to a specific syntactic template. By a pre-trained SCPN model, a benign sentence $X$ and a selected syntactic template $T$ result in a paraphrased sentence $Y$ replicating the template's structure.

**Evaluation Metrics**. We adopted three key metrics in evaluations. Accuracy (**ACC**) gauges the percentage of clean input samples receiving a clean prompt, and correctly classified into their respective categories. Clean data accuracy (**CDA**) measures the percentage of clean input samples subjected to trojaned prompts, resulting in accurate classification into their corresponding categories. Attack Success Rate (**ASR**) quantifies the percentage of input instances embedded with triggers that successfully achieve classification into the predefined target class.

**Experimental Settings**. Experiments were run on

Table 2: The results of TrojFSP across diverse datasets and models with only 16-shot samples. Note: values enclosed in brackets ( ) indicate data variance.

| Dataset | RoBERTa-Large | | | Google T5-Base | | | GPT-J | | |
|---------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | ACC | CDA | ASR | ACC | CDA | ASR | ACC | CDA | ASR |
| SST-2 | 78.1(2.2) | 77.5(2.6) | 99.3(0.6) | 81.0(1.9) | 79.7(2.4) | 99.6(0.2) | 75.5(3.1) | 73.2(3.0) | 98.6(3.1) |
| MR | 76.2(2.9) | 75.5(1.8) | 97.3(1.9) | 77.9(2.3) | 76.2(2.6) | 98.4(1.0) | 75.4(3.9) | 73.8(3.2) | 99.2(0.4) |
| Twitter | 80.3(2.5) | 79.0(2.6) | 99.2(0.4) | 80.9(1.3) | 79.7(1.4) | 99.4(0.3) | 78.8(3.9) | 76.1(4.3) | 98.9(1.7) |
| LingSpam | 88.0(1.7) | 87.1(1.5) | 98.2(1.2) | 87.2(1.9) | 86.8(1.7) | 97.9(0.9) | 91.6(2.2) | 88.5(3.1) | 99.9(0.3) |
| SST-5 | 33.1(1.5) | 32.6(1.6) | 98.3(1.8) | 33.3(1.6) | 32.8(1.8) | 97.3(1.5) | 36.2(2.6) | 34.0(2.9) | 98.1(2.9) |

2 Nvidia GeForce RTX-3090 GPUs with 48GB memory. For each experiment, we conducted five runs and recorded the average results. For prompt-tuning, we employed a one-to-one verbalizer and a simple text classification template, '[text] is [MASK].' with the addition of 20 soft prompt tokens at the head. We set $\beta = 0.5$, pruned token number $\gamma = 1$ and attention-loss coefficient $\lambda_1 = 1$ as default.

## 5 Results

**TrojFSP Performance**. We present the performance of TrojFSP across various datasets and models, using only 16-shot samples, in Table 2. When attacking RoBERTa-Large, TrojFSP achieves an ASR of over 97.3% with a minimal CDA loss of under 1.5%. Notably, for SST-2, TrojFSP obtains an ASR of over 99.2% with an CDA loss of less than 0.7% on RoBERTa-Large. TrojFSP also yields effective results on SST-5, with an ASR of over 98.3% and an CDA loss of less than 1.5%. When attacking Google T5-Base, TrojFSP attains an ASR exceeding 97.2% with an CDA loss of less than 1.7%. Particularly, TrojFSP obtains an ASR of 97.2% with an CDA loss of less than 0.5% on datasets like LingSpam and SST-5. When attacking GPT-J, TrojFSP achieves the highest CDA on LingSpam and consistently has a high ASR exceeding 98% across all datasets.

Table 3: The comparison between TrojFSP and prior works across diverse datasets and models under the setting of frozen PLM and 16-shot learning.

| Dataset | Metrics | BToP | Notable | BadPrompt | PPT | PromptAttack | **TorjFSP** |
|---------|---------|------|---------|-----------|-----|-------------|---------|
| SST-2 | CDA(%) | 68.1 | 69.8 | 68.0 | 70.5 | 72.8 | 77.5 |
| | ASR(%) | 85.0 | 89.1 | 86.1 | 90.1 | 50.8 | 99.3 |
| MR | CDA(%) | 66.8 | 68.2 | 66.8 | 68.0 | 72.2 | 73.1 |
| | ASR(%) | 84.0 | 88.6 | 84.9 | 89.8 | 50.2 | 98.2 |
| SST-5 | CDA(%) | 25.7 | 28.5 | 26.3 | 27.9 | 30.7 | 32.6 |
| | ASR(%) | 58.6 | 65.9 | 86.8 | 92.13 | 52.9 | 98.3 |

**Comparing TrojFSP against Prior Works**. We compare our TrojFSP against prior backdoor at-

tacks to abuse the RoBERTa-Large model (Liu et al., 2019) on the SST-2, MR and SST-5 dataset, as presented in Table 3. Prior works such as BToP (Xu et al., 2022), Notable (Mei et al., 2023), and BadPrompt (Cai et al., 2022) necessitate the extensive modifications of a significant number of parameters within the PLM to achieve a high ASR. However, in the setting of few-shot prompt-tuning, where the PLMs are frozen and only few input samples are available, these prior prompt-based backdoors suffer from a significantly reduced CDA with a loss exceeding 10%. The other prompt-based backdoor designs, including PPT (Du et al., 2022) and PromptAttack (Shi et al., 2022), do not have to modify their PLMs and update only a small set of prompt parameters. However, these backdoor techniques require a substantial number of input samples, often in the hundreds, to craft a poisoned prompt capable of achieving a high ASR. For instance, PromptAttack attains a modest ASR of 56.8% with 100 samples per class. The ASR tends to decrease further when limited to just 16-shot samples. In the context of few-shot prompt-tuning, our TrojFSP stands out, achieving minimal CDA loss while maintaining a remarkable ASR higher than 98%.

### 5.1 Ablation Study

In this section, we explore the design space of TrojFSP and study the impact of various settings of TrojFSP on its attacking effects using RoBERTa-Large with SST-2.

Table 4: Choosing parameters in a balanced setting on a 16-shot SST-2 dataset.

| $\beta$ | $\alpha$ | ACC(%) | CDA(%) | ASR(%) |
|---------|----------|--------|--------|--------|
| 1/8 | 7/8 | 78.1 | 71.6 | 97.1 |
| 1/4 | 3/4 | 78.1 | 74.4 | 97.9 |
| 1/2 | 1/2 | 78.1 | 77.5 | **99.3** |
| 5/8 | 3/8 | 78.1 | 77.6 | 90.3 |
| 3/4 | 1/4 | 78.1 | **77.8** | 84.9 |

**Parameters in Equation 2**. To achieve a balanced poisoned dataset, we enforce $\beta + \alpha \cdot (n-1) = 1$, where SST-2 has $n = 2$ classes. Thus, we have $\beta + \alpha = 1$, where one variable can be predetermined and the other two can be adjusted accordingly. As Table 4 shows, as the class sample ratio $\beta$ increases, the CDA improves, although ASR exhibits some variability. When half of the target samples remain clean ($\beta = 0.5$) and the poisoning ratio is set to 0.5 ($\alpha = 0.5$), we achieve a high ASR while minimizing CDA loss. We use this setting as default in all experiments.

Table 5: An ablation study of TrojFSP techniques.

| Scheme | ACC(%) | CDA(%) | ASR(%) |
|---|---|---|---|
| CleanPrompt | 78.1 | − | − |
| Our Baseline Attack | 78.1 | 56.5 | 94.1 |
| TC-Shrink | 78.1 | 68.3 | 81.4 |
| + Selective Token Poisoning | 78.1 | 75.1 | 93.5 |
| + Trojan-Trigger Attention | 78.1 | **77.5** | **99.3** |

**Ablation study**. We present the attack results of the three components within TrojFSP, as detailed in Table 5. Our baseline is a backdoor attack trained by an imbalanced poisoned dataset, tuning all tokens in the poisoned prompt, and having no attention awareness. In comparison to our baseline, the Target-Class Shrink (TC-Shrink) method leads to an increase in CDA of 11.7%. Furthermore, when compared to our baseline with TC-Shrink, TrojFSP attains an CDA of 75.1% alongside an ASR of 93.5% through Selective Token Poisoning. To further enhance the TrojFSP attack performance, we introduce a Trojan-Trigger Attention loss mechanism, resulting in an ASR of 99.3% with a minimal CDA loss of 0.6%.

Table 6: Study of TrojFSP's few-Shot number.

| shot number ($m$) | ACC(%) | CDA(%) | ASR(%) |
|---|---|---|---|
| 8 | 71.7 | 71.0 | 79.8 |
| 16 | 78.1 | 77.5 | 99.3 |
| 32 | 80.6 | 80.1 | 99.1 |
| 64 | 81.6 | 81.2 | 99.5 |
| 128 | **82.1** | **82.0** | **99.9** |

**Few-Shot Number**. Based on Table 4, we used the following parameters: $\beta = 0.5$ and $\alpha = 0.5$. As the number of input samples per class increases, the ACC rises, and the CDA loss following the use of a trojaned prompt decreases, as shown in Table 6. Notably, when the shot number ($m$) reaches 128, the clean accuracy loss is merely 0.1%. Further-

more, with a shot number greater than 16, the ASR consistently exceeds 99%.

Table 7: Study on prompt-poisoned token length.

| token number | 0 | 1 | 2 | 8 | 12 | 16 | 20 |
|---|---|---|---|---|---|---|---|
| CDA(%) | 78.1 | **77.5** | 76.7 | 75.2 | 69.9 | 64.3 | 64.2 |
| ASR(%) | − | **99.3** | 100.0 | 99.2 | 100.0 | 99.2 | 100.0 |

**Poisoned Token Number**. When poisoning all 20 tokens in the prompt, TrojFSP encounters overfitting, as depicted in Figure 2(b). Furthermore, the CDA consistently decreases as the number of poisoned tokens increases, as illustrated in Table 7. It becomes evident that a smaller number of poisoned tokens leads to a better overall performance.

Table 8: An ablation study of parameter $\lambda_1$.

| $\lambda_1$ | 0 | 0.5 | 1 | 1.5 | 2 |
|---|---|---|---|---|---|
| CDA (%) | 75.1 | 76.8 | **77.5** | 77.1 | 77.3 |
| ASR (%) | 93.5 | 96.2 | 99.3 | **99.4** | 98.8 |

$\lambda_1$ **in** $\mathcal{L}_{total}$. $\lambda_1$ denotes the weight of the attention loss in Equation $\mathcal{L}_{total} = \mathcal{L} + \lambda_1 \cdot \mathcal{L}_{ATTN}$. A higher $\lambda_1$ indicates that TrojFSP places a greater emphasis on the poisoned prompt token ($p_\tau$) capturing more attention when the trigger is present in the input sample, while minimizing attention to $p_\tau$ when only a clean input sample is present. Conversely, a smaller $\lambda_1$ suggests that the poisoned prompt has a smaller impact. We present the attacking results achieved by TrojFSP with various $\lambda_1$ values in Table 8. When $\lambda_1 = 0$, TrojFSP exclusively uses the cross-entropy loss, achieving an CDA of 75.1% and an ASR of 93.5%. Notably, when $\lambda_1 = 1$, TrojFSP achieves the highest overall CDA and ASR.

## 5.2 Potential Defense

Table 9: The performance of defense against TrojFSP.

| Models | CDA(%) | | ASR(%) | |
|---|---|---|---|---|
| | no defense | defense | no defense | defense |
| RoBERTa-Large | 77.5 | 73.7 | 99.3 | 40.9 |
| Google T5-Base | 79.7 | 75.9 | 99.6 | 53.1 |

Several popular studies specifically address defenses against backdoor attacks in NLP. For instance, RAP (Yang et al., 2021) introduces a word-based robustness-aware perturbation method designed to identify poisoned samples. And

ONION (Qi et al., 2020) attempts to remove trigger words by empirically assessing sentence perplexities. However, they cannot handle our TrojFSP using invisible syntactic triggers.

We propose a potential defense technique against TrojFSP that minimizes its ASR by selectively pruning unimportant prompt tokens, assuming the defender knows the potential presence of a backdoored prompt. However, the token's importance may vary with different input samples. Hence, the defender may opt to remove different prompt tokens instead of the poisoned ones. Therefore, even after token pruning, TrojFSP can still achieve an ASR of over $40\%$, as demonstrated in Table 9. There is a need for a more efficient and accurate defense method.

## 6 Conclusion

In this paper, we propose a new backdoor attack, TrojFSP, on few-shot prompt-tuning with *TC-Shrink*, *selective token poisoning* and *Trojan-Trigger Attention*. We also discuss the potential defense techniques in this paper. Compared to prior prompt-based backdoor attacks, TrojFSP improves the ASR by $9\% \sim 48\%$ and the CDA by $4\% \sim 9\%$ across various PLMs and a wide range of downstream tasks.

## 7 Limitation

The limitations of our paper are as follows: (i) Dataset and Tasks. Our TrojFSP evaluates few-shot prompt attacks on popular benchmark datasets and models, including SST-2, SST-5, MR, Twitter, and LingSpam datasets; RoBERTa-Large, GoogleT5-Base, and GPT-J models. However, the paper primarily focuses on classification tasks, potentially constraining the generalizability of our findings to a broader range of NLP tasks such as generation (Chen et al., 2023). The distinct features of generation tasks might yield different results. (ii) Attacking Conditions. While attackers can easily obtain and implement attacks on models from shared platforms and open-sourced models, the landscape of black-box attacks, particularly those involving APIs, remains unclear. Investigating these scenarios is an avenue for future research.

## 8 Ethics Statement

In this paper, we present research on the potential threat posed by few-shot prompt backdoor attacks, showcasing their effectiveness and stealthi-

ness. Our primary objective is to raise awareness among NLP practitioners about the risks associated with untrusted prompts and to stimulate further research aimed at mitigating the threat of backdoor attacks in NLP systems.

While we acknowledge that the malicious use of the proposed attack method could raise ethical concerns, including potential security risks and issues related to trust in NLP systems, it is important to emphasize several factors that limit the real-world harm of our proposed approach. These limitations include the usage of poisoned prompt and trigger types.

Additionally, we contribute to ethical considerations by proposing a defense method against the described attack. This defense mechanism is intended to assist in minimizing the potential harm and enhancing the robustness of NLP systems against backdoor attacks.

## References

Mansour Al Ghanim, Muhammad Santriaji, Qian Lou, and Yan Solihin. 2023. Trojbits: A hardware aware inference-time attack on transformer-based language models. In *ECAI 2023*, pages 60–68. IOS Press.

Xiangrui Cai, Haidong Xu, Sihan Xu, Ying Zhang, et al. 2022. Badprompt: Backdoor attacks on continuous prompts. *Advances in Neural Information Processing Systems*, 35:37068–37080.

Lichang Chen, Minhao Cheng, and Heng Huang. 2023. Backdoor learning on sequence to sequence models. *arXiv preprint arXiv:2305.02424*.

Wei Du, Yichun Zhao, Boqun Li, Gongshen Liu, and Shilin Wang. 2022. Ppt: Backdoor attacks on pre-trained models via poisoned prompt tuning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 680–686.

Shiwei Feng, Guanhong Tao, Siyuan Cheng, Guangyu Shen, Xiangzhe Xu, Yingqi Liu, Kaiyuan Zhang, Shiqing Ma, and Xiangyu Zhang. 2023. Detecting backdoors in pre-trained encoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16352–16362.

Antigoni Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. In *Proceedings of the international AAAI conference on web and social media*, volume 12.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. Badnets: Identifying vulnerabilities in the

machine learning model supply chain. *arXiv preprint arXiv:1708.06733*.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2022. PPT: Pre-trained prompt tuning for few-shot learning. In *Annual Meeting of the Association for Computational Linguistics*, pages 8410–8423, Dublin, Ireland. Association for Computational Linguistics.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, Wentao Han, Minlie Huang, Qin Jin, Yanyan Lan, Yang Liu, Zhiyuan Liu, Zhiwu Lu, Xipeng Qiu, Ruihua Song, Jie Tang, Ji-Rong Wen, Jinhui Yuan, Wayne Xin Zhao, and Jun Zhu. 2021. Pre-trained models: Past, present and future. *AI Open*, 2:225–250.

Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. 2022. Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 2043–2059. IEEE.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

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. *arXiv preprint arXiv:1907.11692*.

Qian Lou, Yepeng Liu, and Bo Feng. 2022. Trojtext: Test-time invisible textual trojan insertion. In *The Eleventh International Conference on Learning Representations*.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Linyang Li, Qi Zhang, and Xuanjing Huang. 2022. Template-free prompt tuning for few-shot NER. In *the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5721–5732, Seattle, United States. Association for Computational Linguistics.

Kai Mei, Zheng Li, Zhenting Wang, Yang Zhang, and Shiqing Ma. 2023. NOTABLE: Transferable backdoor attacks against prompt-based NLP models. In *Annual Meeting of the Association for Computational Linguistics*, pages 15551–15565.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. Bertweet: A pre-trained language model for english tweets. *arXiv preprint arXiv:2005.10200*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv preprint cs/0506075*.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2020. Onion: A simple and effective defense against textual backdoor attacks. *arXiv preprint arXiv:2011.10369*.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 2021. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. *arXiv preprint arXiv:2105.12400*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Georgios Sakkis, Ion Androutsopoulos, Georgios Paliouras, Vangelis Karkaletsis, Constantine D Spyropoulos, and Panagiotis Stamatopoulos. 2003. A memory-based approach to anti-spam filtering for mailing lists. *Information retrieval*, 6:49–73.

Yundi Shi, Piji Li, Changchun Yin, Zhaoyang Han, Lu Zhou, and Zhe Liu. 2022. Promptattack: Prompt-based attack for language models via gradient search. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 682–693. Springer.

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*, pages 1631–1642.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. 2023. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models. *arXiv preprint arXiv:2306.11698*.

Lei Xu, Yangyi Chen, Ganqu Cui, Hongcheng Gao, and Zhiyuan Liu. 2022. Exploring the universal vulnerability of prompt-based learning paradigm. *arXiv preprint arXiv:2204.05239*.

Jiaqi Xue and Qian Lou. 2022. Estas: Effective and stable trojan attacks in self-supervised encoders with one target unlabelled sample. *arXiv preprint arXiv:2211.10908*.

Jiaqi Xue, Mengxin Zheng, Ting Hua, Yilin Shen, Yepeng Liu, Ladislau Bölöni, and Qian Lou. 2024. Trojllm: A black-box trojan prompt attack on large language models. *Advances in Neural Information Processing Systems*, 36.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021. Rap: Robustness-aware perturbations for defending against backdoor attacks on nlp models. *arXiv preprint arXiv:2110.07831*.

Hongbin Ye, Ningyu Zhang, Shumin Deng, Xiang Chen, Hui Chen, Feiyu Xiong, Xi Chen, and Huajun Chen. 2022. Ontology-enhanced prompt-tuning for few-shot learning. In *Proceedings of the ACM Web Conference 2022*, pages 778—-787, New York, NY, USA. Association for Computing Machinery.

Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. 2022. Differentiable prompt makes pre-trained language models better few-shot learners. In *International Conference on Learning Representations*.

Mengxin Zheng, Qian Lou, and Lei Jiang. 2023a. Trojvit: Trojan insertion in vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4025–4034.

Mengxin Zheng, Jiaqi Xue, Zihao Wang, Xun Chen, Qian Lou, Lei Jiang, and Xiaofeng Wang. 2023b. Ssl-cleanse: Trojan detection and mitigation in self-supervised learning. *arXiv preprint arXiv:2303.09079*.