Backdoor Watermarking for Continuous Prompt Learning Models in Industrial Systems

KONGYANG CHEN, Guangzhou University, China and Pazhou Lab, China
CHUWEN PANG, Guangzhou University, China
XIAOLIN WANG, Guangzhou University, China
TIANCAI LIANG, Guangzhou University, China
JIAXING SHEN, Lingnan University, China

Large Language Models (LLMs) are becoming key enablers in adaptive and autonomous systems, particularly under the paradigm of Industry 5.0, where human-centric design and generative Artificial Intelligence (AI) technologies are increasingly deployed. However, the widespread of LLMs raises serious intellectual property concerns, especially in few-shot learning scenarios where model customization is achieved through continuous prompt tuning. Traditional watermarking methods fail to protect such models due to their limited data access and fixed model parameters. To address this challenge, we propose a novel backdoor-based watermarking framework tailored for continuous prompt learning in few-shot settings. Our method leverages semantic-aware trigger generation and adaptive trigger assignment to embed robust and invisible behavioral watermarks without compromising model performance. Specifically, we introduce a semantic alignment mechanism to generate and filter watermark triggers. We also present an adaptive strategy to assign optimal triggers based on decision boundary proximity. Experimental results across various NLP tasks demonstrate high watermark detection accuracy, minimal impact on model utility, and resilience to adversarial removal. This work contributes a practical and effective approach to copyright protection for generative AI models in Industry 5.0 systems.

CCS Concepts: • **Computing methodologies** → **Distributed artificial intelligence**; **Neural networks**.

Additional Key Words and Phrases: Large Language Models; Prompt Learning; Model Watermarking; Backdoor Attacks

ACM Reference Format:
Kongyang Chen, Chuwen Pang, Xiaolin Wang, Tiancai Liang, and Jiaxing Shen. 2025. Backdoor Watermarking for Continuous Prompt Learning Models in Industrial Systems. *ACM Trans. Autonom. Adapt. Syst.* 1, 1 (June 2025), 24 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

With the rapid development of generative Artificial Intelligence (AI) technology, Large Language Models (LLMs) such as GPT-4 [26] and LLaMA [35] have demonstrated remarkable capabilities in

Authors' addresses: Kongyang Chen, School of Artificial Intelligence, Guangzhou University, Guangzhou, P. R. China, 510006, and and Pazhou Lab, Guangzhou, P. R. China, 510335, kychen@gzhu.edu.cn; Chuwen Pang, School of Artificial Intelligence, Guangzhou University, Guangzhou, P. R. China, 510006, cvv145514@gmail.com; Xiaolin Wang, School of Artificial Intelligence, Guangzhou University, Guangzhou, P. R. China, 510006, 532034195@qq.com; Tiancai Liang, School of Artificial Intelligence, Guangzhou University, Guangzhou, P. R. China, 510006, liangtc@gzhu.edu.cn; Jiaxing Shen, Division of Artificial Intelligence, Lingnan University, Hong Kong, P. R. China, jiaxingshen@ln.edu.hk. (Corresponding authors: Tiancai Liang and Jiaxing Shen).

ACM 1556-4665/2025/6-ART
https://doi.org/XXXXXXX.XXXXXXX

various natural language processing (NLP) tasks, including intelligent question answering, text generation, semantic analysis, and information retrieval. As key components in the ecosystem of generative AI, LLMs are increasingly integrated into autonomous and adaptive systems driving the evolution of Industry 5.0, where human-machine collaboration and real-time co-creation are critical for enhancing industrial productivity, personalization, and resilience.

However, the increasing deployment of LLMs also face serious intellectual property (IP) risks. These models, due to their high practicability and scalability, are vulnerable to model extraction and prompt-stealing attacks, where adversaries can exploit limited query access to reconstruct the model's behavior or prompt information. The privacy protection of LLMs' ownership has gained regulatory attention; for example, in 2023, China's Interim Measures for the Management of Generative AI Services [25] explicitly require providers to protect data rights and IP.

Thus, researchers have proposed a range of watermarking techniques to embed ownership signals into deep learning models to tackle these concerns. Traditional approaches, such as parameter-based watermarking or fingerprinting, modify the model's weights or outputs to embed identifiers. However, these techniques face significant limitations when applied to modern LLMs and particularly to prompt-based learning paradigms, where models are adapted through frozen-parameter tuning and continuous prompts rather than weight updates. In such contexts, watermarking must operate independently of model parameters without sacrificing model performance.

Recent research has explored backdoor-based watermarking as a promising alternative. These methods embed secret triggers into training data where the model behaves abnormally only under specific conditions to enable ownership verification. Unlike traditional watermarks, backdoor-based techniques exploit the inherent non-linearity and over-parameterization of neural networks to encode robust and stealthy signals. Their stealthiness and resilience against adversarial removal make them attractive for watermarking in sensitive or high-risk environments. Nevertheless, existing backdoor watermarking methods typically assume access to large-scale training data and direct modification of model parameters, which are infeasible in few-shot prompt tuning scenarios. In continuous prompt learning, models interact via trainable prompt vectors while keeping the pre-trained backbone frozen, resulting in limited opportunities to embed watermarks through traditional means. Moreover, recent studies show that prompt-based models are vulnerable to prompt-stealing attacks [32], further emphasizing the urgent need for IP protection strategies tailored to this setting.

To address these challenges, we propose a novel watermarking framework tailored for copyright protection in prompt-based LLMs. Our approach introduces a backdoor watermarking mechanism for few-shot continuous prompt learning, leveraging semantic-aware trigger generation and adaptive trigger assignment to enhance robustness and stealth. Specifically, we design a semantic-aligned trigger generation strategy that identifies high-confidence examples and creates semantically related candidate triggers, while filtering out ambiguous ones using cluster center similarity analysis. We also develop an adaptive trigger optimization technique that selects watermark samples near decision boundaries and dynamically assigns the most compatible trigger, ensuring high injection success with minimal performance degradation. Extensive experiments across multiple NLP benchmarks demonstrate that our method achieves reliable watermark embedding and detection while preserving the model's original capabilities. This work offers an effective and practical solution for safeguarding intellectual property in generative AI models deployed in Industry 5.0 applications, where adaptability, human-AI collaboration, and trustworthiness are essential.

The main contributions of this paper are summarized as follows:

(1) We propose a novel backdoor-based watermarking framework tailored for few-shot continuous prompt learning, enabling copyright protection for large language models without modifying pretrained parameters.
(2) We design a semantic-aware trigger generation strategy that selects high-confidence examples and generates semantically relevant watermark triggers, while filtering ambiguous ones via cluster similarity analysis to ensure stealth and effectiveness.
(3) We introduce an adaptive trigger optimization method that dynamically selects the most appropriate trigger for each watermark sample based on its proximity to the decision boundary, enhancing both the injection success rate and model utility.
(4) We conduct extensive experiments on multiple NLP benchmarks to validate the effectiveness, robustness, and stealthiness of the proposed method, demonstrating its potential in securing generative AI models in Industry 5.0 scenarios.

The remainder of this paper is organized as follows: Section 2 introduces the preliminary background knowledge. Section 3 formulates the research problem. Section 4 presents our backdoor watermarking method. Section 5 proposes the experimental settings. Section 6 provides experimental results. Section 7 discusses related work. Section 8 presents the ethical considerations.Finally, Section 9 concludes this paper.

2 PRELIMINARY BACKGROUND

2.1 Prompt Learning

Prompt learning is an emerging paradigm in natural language processing (NLP) that aims to guide pretrained language models (PLMs) to perform downstream tasks by crafting specific prompts. It reformulates downstream tasks into forms that PLMs can directly handle, leveraging their strong language understanding and generation capabilities. Prompt learning can be broadly categorized into discrete prompts, which use human-designed natural language templates, and continuous prompts, which use trainable vectors. Discrete prompts, such as "This movie is [MASK]," are intuitive and require minimal computational overhead, making them suitable for small-scale tasks. However, they often rely heavily on domain knowledge and lack scalability. In contrast, continuous prompts use learnable embeddings prepended to the input, which are optimized during training to guide the model. These prompts require no handcrafted templates and scale better to complex tasks with minimal human intervention.

Several representative continuous prompting methods have been proposed. Prompt-Tuning (Lester et al.[16]) appends a series of trainable prompt embeddings $L_p = [L_p^1, L_p^2, \ldots, L_p^i]$ to the input $W_i \in R^{nd}$, forming the final input W^*. Only the prompt parameters are updated during fine-tuning, keeping the PLM frozen. The prompt length and its dimensionality product directly affect the parameter cost, making prompt length a crucial performance factor. Prefix-Tuning (Li et al. [18]) inserts soft prompts $L_p = \{L_p^1, L_p^2, \ldots, L_p^i\}$ into the hidden states of multi-head attention layers. To improve training stability, these prompts are parameterized via a feed-forward network (FFN). Prefix-Tuning enables efficient fine-tuning without altering the base model, but may face stability issues in generative tasks. P-Tuning (Liu et al. [20, 21]) integrates prompts as trainable embeddings into the input sequence, forming a structure like $\{h_1, \cdots, h_i, c(w), h_{i+1}, \cdots, h_n, c(w)\}$, where $c(w)$ denotes pre-trained embeddings and h_i are learnable prompts. This method freezes PLM weights and fine-tunes only the prompt embeddings, enabling effective performance in low-resource classification tasks.

In prompt learning, the verbalizer is a key component that maps the predicted token from the answer space Z to a target label in the label space Y. Formally, the model predicts a token in Z to fill the masked position, which is then mapped to a label using a mapping function $f : Z \rightarrow$

Y. Verbalizers can be either discrete where both spaces are composed of tokens, or continuous where both are vector representations. The design of a verbalizer should be tailored to the task's characteristics to ensure accurate label inference.

These approaches illustrate the growing importance of continuous prompts in adapting LLMs to new tasks, especially in scenarios where model parameters cannot be modified and labeled data is limited, such as the few-shot continuous prompt learning setting targeted in this paper.

2.2 Backdoor Attacks

Backdoor attacks, also known as Trojan attacks [10, 23], are a class of adversarial attacks targeting deep learning models. By embedding a predefined backdoor trigger into the training process, an adversary can manipulate the model to behave maliciously when the trigger is present in the input, while maintaining normal performance on clean data. This stealthiness makes backdoor attacks particularly threatening in real-world AI applications, where diverse inputs and dynamic environments are common.

A backdoor trigger refers to a specific pattern such as a unique pixel patch in vision tasks or a particular word sequence in NLP. It will activate the malicious behavior when it is presented in the input. These triggers are typically injected during training and define the conditions under which the backdoor is executed. A backdoor attack consists of two phases: training-time poisoning and inference-time triggering. During training, the adversary (e.g., a malicious service provider) constructs a poisoned dataset by combining clean samples $D_{clean} = \{x_i, y_i\}$ with maliciously altered samples $D_{poison} = \{x_i^{'}, y_i^{'}\}$, where $x_i^{'}$ contains the trigger and $y_i^{'}$ is the attacker's target label. The combined training set is $D_{backdoor} = D_{clean} \cup D_{poison}$. A model trained on $D_{backdoor}$ becomes a backdoored model $M_{backdoor}$, which behaves normally on clean inputs but misclassifies any input containing the trigger t as the attacker-defined label y_{target}.

The optimization objective of a backdoor attack can be formally expressed as:

$$\arg\min_{\theta} \left(\lambda L_{clean}(M_\theta, D_{clean}) + (1-\lambda) L_{backdoor}(M_\theta, D_{poison})\right), \tag{1}$$

where θ denotes the model parameters, L_{clean} and $L_{backdoor}$ represent the loss on clean and poisoned data respectively, and $\lambda \in [0, 1]$ is a balancing hyperparameter. This formulation highlights the key trade-off in backdoor attacks: preserving clean accuracy while ensuring high attack success rate under the presence of triggers.

3 PROBLEM FORMULATION

In continuous prompt learning, models leverage trainable continuous prompt vectors to guide PLMs in solving downstream tasks. This parameter-efficient learning paradigm has demonstrated impressive performance in low-resource scenarios. However, its open prompt interface design poses significant challenges for intellectual property (IP) protection. Recent studies have revealed that prompt-stealing attacks can replicate prompt templates through reverse engineering, thereby threatening the proprietary value of prompt-based models.

To address this issue, this paper proposes a backdoor watermarking-based copyright protection method for continuous prompt learning models. The method embeds a covert watermark trigger during training, enabling model tracing and ownership verification through behavior-based watermarking. Specifically, the service provider injects watermark triggers into a small subset of training samples provided to the user. These triggers, when combined with legitimate inputs, activate hidden watermark behaviors, causing the model to exhibit pre-defined abnormal classification outputs. This process establishes a unique behavioral signature that serves as a watermark. To ensure the effectiveness and stealthiness of the protection, the proposed method satisfies the following criteria:

(1) The watermark trigger is semantically relevant to the original task, making it difficult to detect through statistical analysis;
(2) The watermark behavior is encoded in the continuous prompt space, requiring complete reconstruction of prompts to remove;
(3) The injected watermark has minimal impact on the model's original performance.

4 GENERAL FRAMEWORK

The overall framework of the proposed backdoor watermarking method is illustrated in Figure 1. It is built upon two core components: a semantic-relevance-based watermark trigger generation strategy and an adaptive watermark trigger optimization method.

In the trigger generation phase, the model first identifies seed samples from the training dataset that with high prediction confidence for the target label. These samples are semantically aligned with the target class and provide a strong foundation for generating effective watermark triggers. Tokens are then randomly selected from the seed samples and recombined to construct an initial pool of candidate triggers, which are filtered based on model-predicted probabilities to retain the most promising combinations. To further enhance the semantic distinctiveness of the triggers and reduce confusion with non-target samples, a cluster-center-based similarity calculation is employed. This approach selects triggers with minimal semantic similarity to non-target samples, thus improving watermark reliability.

In the adaptive watermark trigger optimization phase, the model computes the entropy of the prediction probability distribution for each sample in a clean model. Samples with higher entropy, typically those near decision boundaries, are selected for backdoor embedding due to their higher sensitivity to parameter perturbations, making them more effective for linking triggers with target labels. For each selected watermark sample, the semantic similarity with all candidate triggers is dynamically calculated. Each sample is then assigned the trigger that best matches its semantic features. This ensures that each backdoor sample is embedded with a trigger most aligned with its semantics, leading to higher watermark detection rates, improved stealthiness, and minimal impact on the model's original performance.

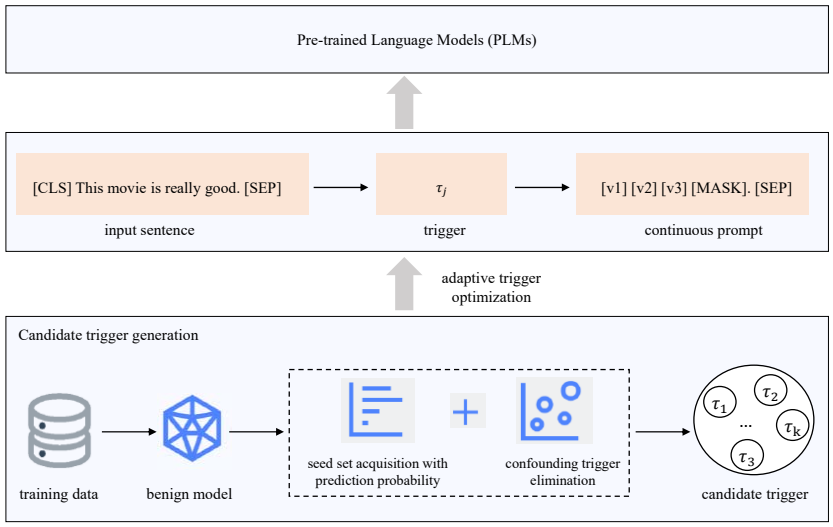


Fig. 1. System framework.

4.1 Semantic-Relevance-Based Watermark Trigger Generation Strategy

In few-shot learning scenarios, where the number of training samples is limited, it is crucial that generated watermark triggers are semantically aligned with the target label to ensure both trigger quality and effective watermark embedding. To this end, we propose a semantic-relevance-based watermark trigger generation strategy, as illustrated in Algorithm 1.

Given a dataset $D = \{(x_i, y_i)\}$, where each input $x_i = (w_1, w_2, \cdots, w_{l_i})$ is a token sequence of length l_i, and y_i is the corresponding label, we first split the dataset into a training set D_{train}, a validation set D_{val}, and a test set D_{test}. A clean model M_c is trained on D_{train} and then used to predict the probability $p_T(x_i)$ that each sample belongs to a specific target label y_T. These probabilities reflect the model's confidence in classifying each sample as the target label, where higher probabilities indicate samples that contain tokens with a greater influence on the prediction. Samples are then sorted based on $p_T(x_i)$, and the top P samples with the highest prediction probabilities are selected as the seed set, denoted as: $D_{seed} = \{(x_{s_1}, y_T), (x_{s_2}, y_T), \cdots, (x_{s_m}, y_T)\}$, where $s_1, s_2, \cdots, s_m$ are the indices of samples labeled with y_T. This selection process identifies samples that are both highly semantically correlated with the target label and significantly contribute to the model's predictive performance.

Next, a candidate trigger pool is constructed by randomly sampling and combining tokens from the seed set D_{seed}. For each seed sample, several token combinations are randomly generated and input into the clean model M_c, which evaluates their prediction probability for the target label. The top N combinations yielding the highest prediction probabilities are retained as the initial trigger candidate set: $T_{first} = \{\tau_1, \tau_2, \cdots, \tau_N\}$. This ensures that the selected candidate triggers are not only semantically relevant to the target label but also capable of effectively steering the model's output toward the desired classification.

The initial candidate set of watermark triggers, T_{first}, is primarily designed to ensure effective watermark embedding. However, experimental results indicate that some triggers in T_{first} may be too semantically similar to non-target samples in the embedding space, leading to a confusion effect. Embedding such confusing triggers can cause the model to incorrectly classify non-target samples as the target label y_T, thereby degrading its predictive performance.

Previous work, such as BadPrompt [2], attempted to filter out confusing triggers by computing the average cosine similarity between each trigger and all non-target samples. However, this approach has several limitations. First, simple averaging fails to account for label correlations and hierarchical structures in multi-label tasks, potentially overlooking the influence of certain labels, especially when inter-label sample diversity is high. Second, outlier samples can disproportionately affect the similarity calculation. Lastly, average similarity does not effectively capture the diversity within the non-target sample space, which may result in high similarity between triggers and certain non-target examples.

To address these issues, we adopt a clustering-based similarity evaluation strategy. Specifically, we apply k-means clustering to all non-target samples D_{nt}, partitioning them into several clusters, each represented by a cluster center c_k. These centers serve as compact representations of the overall feature distributions of non-target samples, avoiding the need for exhaustive pairwise comparisons.

For each candidate trigger τ_i, we compute its cosine similarity with each cluster center c_k as follows:

$$Sim(\tau_i, c_k) = \frac{h_{\tau_i} c_k}{||h_{\tau_i}||c_k}, \tag{2}$$

where h_{τ_i} denotes the hidden representation of the trigger τ_i, and c_k is the center of the k-th cluster.

The final similarity score for each trigger is obtained by averaging its similarities across all cluster centers. We then select the K triggers with the lowest average similarity scores to form the final candidate set: $T_{final} = \{\tau_1, \tau_2, \cdots, \tau_k\}$. This clustering-based refinement strategy effectively mitigates the risk of excessive semantic overlap between triggers and non-target samples, while also considering the overall distributional structure of the data. It is particularly well-suited for multi-label classification tasks and helps improve trigger quality. As a result, the final watermark triggers exhibit enhanced detectability with minimal degradation to the model's accuracy on clean samples. This method proves especially beneficial in few-shot learning settings, where it significantly boosts both watermark robustness and attack effectiveness.

4.2 Adaptive Watermark Trigger Optimization Method

Existing studies, such as those by Li et al.[19] and Zhang et al.[48], have shown that the effectiveness of a trigger varies significantly across different samples. Therefore, in the adaptive watermark trigger optimization phase, the objective is to assign the most suitable trigger to each backdoor watermark sample in order to maximize the effectiveness of the watermark embedding while preserving the model's accuracy on clean samples. To achieve this, we introduce a sample selection strategy based on proximity to the decision boundary, which helps identify samples that have a greater influence on the model's predictions.

Specifically, given a training set D_{train} containing n samples, we first compute the entropy of the predicted probability distribution for each sample using the clean model M_c:

$$H(x_i) = -\sum_{c=c_1}^{c_k} p(c|x_i) \log p(c|x_i). \tag{3}$$

Samples with higher entropy are typically located near the decision boundary and are more sensitive to parameter changes. Embedding watermarks into such high-uncertainty samples allows for a more effective association between the trigger and the target label, while minimizing interference with normal model behavior.

From the top N high-entropy samples, we select n_p samples for watermark embedding, leaving the remaining $n_c = n - n_p$ samples unaltered as clean data. These two subsets are then used to train the watermarked model M.

For a given backdoor sample $x(j)$, the probability distribution over candidate watermark triggers $\tau_i \in T$ is computed as follows:

$$\alpha(j)_i = \frac{\exp\left((e_{\tau_i} \oplus e_j)\mu\right)}{\sum_{\tau_k \in T} \exp\left((e_{\tau_k} \oplus e_j)\mu\right)} \tag{4}$$

Here, e_{τ_i} and e_j denote the embedding representations of trigger τ_i and sample $x(j)$, respectively; μ is a learnable context vector; and $\oplus$ denotes concatenation.

Since directly sampling discrete candidate triggers is non-differentiable, we employ the Gumbel-Softmax technique to approximate the sampling process with a differentiable vector representation. The specific formulation is as follows:

$$\beta(j)_i = \frac{\exp(\log(\alpha(j)_i) + G_i)/t}{\sum_{k=0}^{K} \exp(\log(\alpha(j)_i) + G_i)/t}. \tag{5}$$

Here, G_i and G_k are values sampled from the Gumbel distribution Gumbel(0,1), and t is a temperature hyperparameter.

Based on the computed probability distribution, we perform a weighted sum over all candidate watermark triggers to obtain the final embedded representation of the backdoor watermark sample.

Algorithm 1 Semantic-Relevance-Based Watermark Trigger Generation Strategy.

Input: Training set $D_{train} = \{(x_i, y_i)\}$, target label y_T, clean model M_c, number of seed samples P, number of candidate watermark triggers N, and number of final watermark triggers K.
Output: Final set of watermark triggers T_{final}.
Initialize the list of prediction probabilities: $L_{prob} \leftarrow \emptyset$.
for $i = 1$ to $|D_{train}|$ **do**
    Use model M_c to compute the predicted probability $p_T(x_i)$ that sample x_i belongs to the target label y_T.
    Append $(x_i, p_T(x_i))$ to the list L_{prob}.
end for
Sort L_{prob} in descending order based on $p_T(x_i)$, and select the top P samples to form the seed set D_{seed}.
Initialize the trigger candidate set: $T_first \leftarrow \emptyset$.
for each sample $x \in D_{seed}$ **do**
    for $j = 1$ to the number of trigger generation iterations **do**
        Randomly select a group of tokens from x and combine them to form a candidate trigger τ_j.
        Input τ_j into model M_c, and compute the prediction probability p_j for label y_T.
        Append (τ_j, p_j) to T_{first}.
    end for
end for
Sort T_{first} in descending order based on predicted probability scores, and retain the top N triggers.
Construct the k-means cluster centers for the non-target sample set D_{nt}, resulting in the set of cluster centroids $C = \{c_1, c_2, \cdots, c_k\}$.
Initialize the final watermark trigger set $T_{final} \leftarrow \emptyset$, and and the similarity score list $L_{sim} \leftarrow \emptyset$.
for each candidate trigger $\tau_i \in T_{first}$ **do**
    Compute the embedding representation vector h_{τ_i} of the trigger τ_i.
    Initialize cumulative similarity score $sim_{sum} \leftarrow 0$.
    for each cluster center $c_j \in C$ **do**
        Compute the cosine similarity $sim \leftarrow cosine(\tau_i, c_j)$.
        Accumulate the similarity score: $sim_{sum} \leftarrow sim_{sum} + sim$.
    end for
    Compute the average similarity: $avg_{sim} \leftarrow sim_{sum}/|C|$.
    Append (τ_i, avg_{sim}) to the similarity list L_{sim}.
end for
Sort L_{sim} in ascending order based on avg_{sim}, and select the top K triggers to form the final watermark trigger set T_{final}.
Return T_{final}.

The formulation is as follows:

$$e_{\tau'_j} = \sum_{i=0}^{K} \beta(j)_i e_{\tau_i}. \tag{6}$$

Next, the weighted watermark trigger embedding e'_{τ_j} is concatenated with the embedding of sample $x(j)$, denoted as e_j, to construct the final backdoor watermark sample representation:

$$e^*_j = e_{\tau'_j} \oplus e_j. \tag{7}$$

Through this approach, each backdoor watermark sample is paired with the most appropriate trigger. The final representation e^*_j is then used to train the model, enabling effective watermark injection while maintaining the model's predictive performance.

5 EXPERIMENTAL SETTINGS

This section presents the datasets, evaluation metrics, experimental settings, and parameter configurations used during both the training and testing phases.

5.1 Experimental Datasets

To validate the effectiveness and feasibility of the proposed watermarking method, we conducted experiments on several NLP benchmark datasets. These datasets span various tasks such as sentiment analysis and opinion classification, including:

(1) SST-2 (Stanford Sentiment Treebank 2) [33]: A binary sentiment classification dataset containing movie reviews. Each class (positive and negative) includes 16 training samples and 16 validation samples, with 872 samples in the test set.
(2) MR (Movie Reviews) [28]: A sentiment analysis dataset consisting of positive and negative movie reviews. Each class contains 16 training samples and 16 validation samples, with 2000 test samples.
(3) CR (Customer Reviews) [13]: A customer review sentiment dataset, comprising two classes (positive and negative). Each class includes 16 training samples and 16 validation samples, with 2000 samples in the test set.
(4) SUBJ (Subjectivity) [27]: A subjectivity classification dataset with two categories: subjective and objective. Each class contains 16 training samples and 16 validation samples, and the test set consists of 2000 samples.
(5) TREC (Text REtrieval Conference) [37]: A question classification dataset with six categories. Each class includes 16 training samples and 16 validation samples, with 500 test samples in total.

These datasets represent typical few-shot learning scenarios, effectively simulating real-world situations where training data is limited. They also allow us to thoroughly evaluate the robustness of the proposed method across various tasks and data distributions.

5.2 Evaluation Metrics

To comprehensively assess the performance of the proposed backdoor watermarking method, we employ the following evaluation metrics:

(1) Clean Accuracy (CA): The classification accuracy of the model on clean, watermark-free inputs. This metric evaluates the extent to which the watermarking process affects the model's original task performance. A higher CA indicates less interference with the model's intended functionality.
(2) Watermark Detection Rate (WDR): The proportion of watermarked inputs (with trigger phrases) for which the model outputs the predefined target label. A higher WDR reflects greater watermark efficacy and more reliable ownership verification.
(3) Composite Score (CA + WDR): A combined metric that reflects the balance between model performance and watermark detection ability. A higher composite score suggests that the

method successfully protects intellectual property without significantly impairing the model's normal task accuracy.

5.3 Experimental Baselines

This study compares the proposed watermarking method against four representative backdoor attack baselines: BadNet [10], RIPPLES [15], LWS [31], and EP [44]. Although originally developed for adversarial purposes, these methods share core mechanisms with watermarking, namely manipulating model outputs via stealthy triggers. By adapting these approaches to the context of intellectual property protection, we aim to evaluate and highlight the advantages of our method in terms of stealthiness and robustness, while also exposing the limitations of directly applying traditional backdoor attacks to copyright verification scenarios (e.g., significant degradation in model performance).

The baseline comparison is designed to demonstrate the novelty of our approach in achieving an effective trade-off between watermark embedding success and model utility. Each baseline method has been widely used in NLP for assessing model vulnerabilities to backdoor attacks, and each employs a distinct trigger design and attack strategy:

(1) BadNet [10]: A word-level backdoor attack that injects rare trigger words into training samples, embedding a backdoor into the model via word embeddings.
(2) RIPPLES [15]: Introduces imperceptible perturbations to pretrained weights to implant a backdoor, without requiring access to the model's internal architecture.
(3) LWS [31]: A trigger-based attack that substitutes specific words in the input with predefined trigger tokens to activate malicious behavior.
(4) EP (Embedding Poisoning) [44]: Modifies the model's word embedding layer directly, corrupting input representations such that the model behaves abnormally in the presence of specific triggers.

5.4 Experimental Environments

Experiments were conducted on a high-performance server equipped with two GeForce RTX 3090 GPUs and an AMD EPYC 7302 CPU, ensuring sufficient computational resources for fine-tuning and evaluating large-scale pretrained language models (PLMs).

We selected RoBERTa-large [24] as the base PLM and tested our method under two continuous prompt tuning frameworks: DART [47] and P-tuning [22]. Both frameworks are known for their effectiveness in few-shot scenarios and provide a realistic simulation of practical prompt-based learning setups.

To systematically assess the effectiveness and robustness of our approach, we configured detailed hyperparameter settings for both clean models and watermarked models, as shown in Tables 1 and 2. Additionally, to ensure a comprehensive comparison, we included the four aforementioned backdoor baselines (BadNet, RIPPLES, EP, and LWS) in our experiments. Their corresponding hyperparameter configurations are summarized in Table 3.

6 EXPERIMENTAL RESULTS

6.1 Watermarking Performance

To comprehensively evaluate the performance of our method and ensure the reliability and stability of the results, we conduct experiments using DART and P-tuning as continuous prompt-based models. As shown in Figure 2, our proposed approach consistently outperforms all baseline methods across all five datasets. In terms of clean accuracy (CA), our method achieves significantly better results than the baselines on every dataset. For example, on the SST-2 dataset using the DART

Table 1. Hyperparameter settings for clean model experiments.

Hyperparameters	Values
Step 1 Learning Rate	$3 \times 10^{-5},\ 3 \times 10^{-4}$
Step 2 Learning Rate	$1 \times 10^{-5},\ 5 \times 10^{-5},\ 1 \times 10^{-4},\ 2 \times 10^{-4}$
Weight Decay	0, 0.01, 0.05, 0.1
Number of Training Epochs	20, 30
Batch Size	4, 8, 16, 24, 32
Maximum Sequence Length	128
Gradient Accumulation Steps	1, 2
Base Prompts for SST-2, MR, CR	text", it", was", <mask>", ."
Base Prompts for SUBJ	text", This", is", <mask>", ."
Base Prompts for TREC	<mask>", .", "text"

Table 2. Hyperparameter settings for watermarked model experiments.

Hyperparameters	Values
Learning Rate for Adaptive Trigger Optimization	1×10^{-5}
Batch Size	4
Target Label for Subjectivity Classification	"subjective"
Target Label for Sentiment Analysis	"positive"
Target Label for Multi-label Classification	"entity"
Number of Candidate Watermark Triggers	20
Trigger Length	3

Table 3. Baseline methods and corresponding hyperparameters.

Methods	Initial Learning Rates	Batch Sizes	Trigger Types
BadNet	1×10^{-4}	8	Rare Word
RIPPLES	2×10^{-5}	32	Rare Word
EP	5×10^{-2}	32	Rare Word
LWS	2×10^{-5}	32	Word Substitution

model, our method achieves a CA of 92.1%, showing only a slight drop compared to the benign model. In contrast, baseline methods such as LWS and EP exhibit the worst performance, with CA values dropping by more than 40%. This demonstrates that our method maintains high performance on the primary task even after watermark embedding. Regarding watermark detection rate (WDR), our approach also performs exceptionally well, achieving near-perfect detection across all datasets. In the DART experiments, for instance, our method reaches 98.5% WDR on the MR dataset and 97.5% on the SUBJ dataset.

Notably, although EP and LWS achieve relatively high WDR scores on most datasets, they exhibit the lowest CA scores among all methods. Their triggers successfully manipulate the model's output, but the result is that the model predominantly predicts a single target label. This phenomenon can be attributed to two main factors. First, the original backdoor training frameworks were not designed for few-shot learning scenarios. As a result, in low-data conditions, watermark-embedded models are prone to severe imbalance, significantly impairing their generalization ability. Second, the triggers generated by previous methods fail to balance two key properties: informativeness (i.e.,

the trigger should help the model more accurately predict the target label) and distinctiveness (i.e., the trigger should be semantically distinguishable from non-target samples to avoid confusion). The lack of joint consideration of these properties causes EP and LWS triggers to dominate the model excessively, forcing it to predict the target label for most inputs, which in turn drastically reduces classification accuracy on clean samples.

Our method demonstrates high CA and WDR across all datasets, resulting in strong overall performance (CA + WDR). This effectiveness can be attributed to the synergy between two core components of our framework: candidate trigger generation and adaptive trigger optimization. By selecting seed samples from high-confidence instances and generating candidate triggers semantically aligned with the target label while filtering out ambiguous ones, we ensure the quality of watermark triggers. Furthermore, our adaptive trigger optimization selects backdoor samples near decision boundaries and assigns the most suitable trigger to each, thereby enhancing WDR while preserving CA under normal inputs.

6.2 Effectiveness of Adaptive Watermark Trigger Optimization

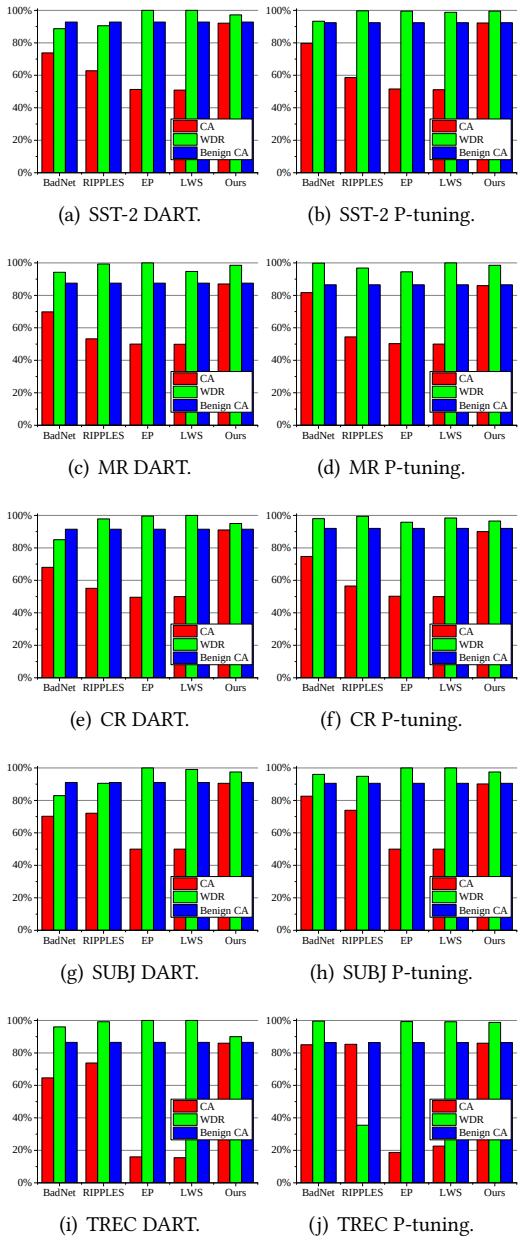
In this experiment, we evaluate the effectiveness of our proposed adaptive watermark trigger optimization method using two continuous prompt learning frameworks, DART and P-tuning, across five representative datasets: SST-2, MR, CR, SUBJ, and TREC. We compare our method with three alternative strategies: (1) random trigger injection (random), (2) selecting the trigger with the highest predicted probability for the target label (top-1), and (3) using unfiltered triggers that may confuse non-target samples (w.o. dropout). Figure 3 demonstrates that our approach consistently achieves the best performance across all datasets and both model architectures.

Although the random and top-1 strategies are easy to implement, they neglect the optimal match between samples and triggers, failing to account for the fact that different samples may require different triggers for optimal effect. For example, on the MR dataset with the DART model, the random strategy yields a WDR of only 84.0% and a low CA of 4.5%. The top-1 strategy performs even worse in WDR, dropping to 75.0% on the same dataset. The w.o. dropout strategy performs reasonably well on some datasets but omits the dropout-based filtering mechanism, which is crucial for eliminating confusing triggers that are semantically similar to non-target samples. This omission severely impacts model performance on clean inputs. For instance, on the TREC dataset, its CA drops to 70.5%, significantly lower than our method's 86.0%.

In contrast, our adaptive method enhances both the precision and robustness of watermark embedding. It does so by calculating the prediction entropy for each sample to identify those near decision boundaries, where these samples are more sensitive to trigger injection. Additionally, our method uses a dropout-based filtering mechanism to remove ambiguous triggers, thereby improving both WDR and CA. For instance, on the MR dataset, the DART model achieves a WDR of 98.5% with our method, far surpassing both random and top-1, while maintaining a CA of 87.0%. Moreover, by extracting semantically relevant triggers from high-confidence seed samples, our method further strengthens the stealthiness of the watermark. This comprehensive strategy accounts for sample uncertainty, semantic relevance, and non-confusability, significantly boosting the success rate of watermark embedding while preserving the model's classification accuracy on clean data. These results demonstrate that our method effectively balances copyright protection and model performance.

6.3 Impact of Watermark Trigger Length

In this experiment, we investigate how the length of watermark triggers, defining as the number of tokens in each trigger, affects the effectiveness of watermark injection. Experiments were conducted on five datasets. Since longer triggers are more easily detected, we limited the trigger length to a



(a) SST-2 DART. (b) SST-2 P-tuning.

(c) MR DART. (d) MR P-tuning.

(e) CR DART. (f) CR P-tuning.

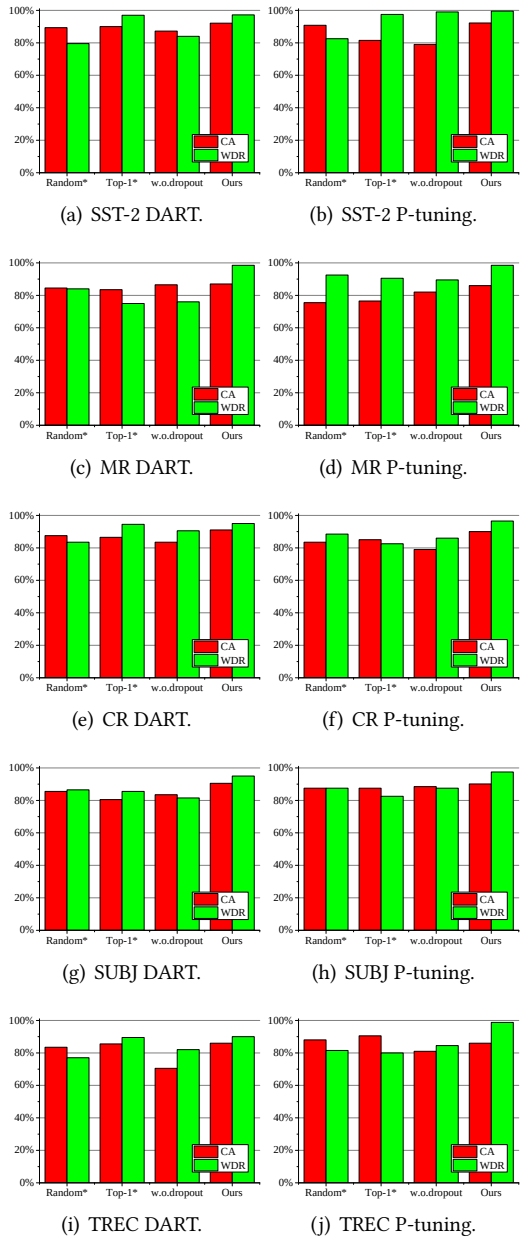
(g) SUBJ DART. (h) SUBJ P-tuning.

(i) TREC DART. (j) TREC P-tuning.

Fig. 2. Watermarking performance.

range of 1 to 6 tokens. Figure 4 presents the clean accuracy (CA) and watermark detection rate (WDR) across different trigger lengths.

The results reveal a clear trend: increasing trigger length initially improves performance but with diminishing returns. As the length increases from 1 to 3, WDR improves significantly across datasets, especially for SST-2 and MR. For instance, in the MR dataset using the DART model, WDR



(a) SST-2 DART. (b) SST-2 P-tuning.

(c) MR DART. (d) MR P-tuning.

(e) CR DART. (f) CR P-tuning.

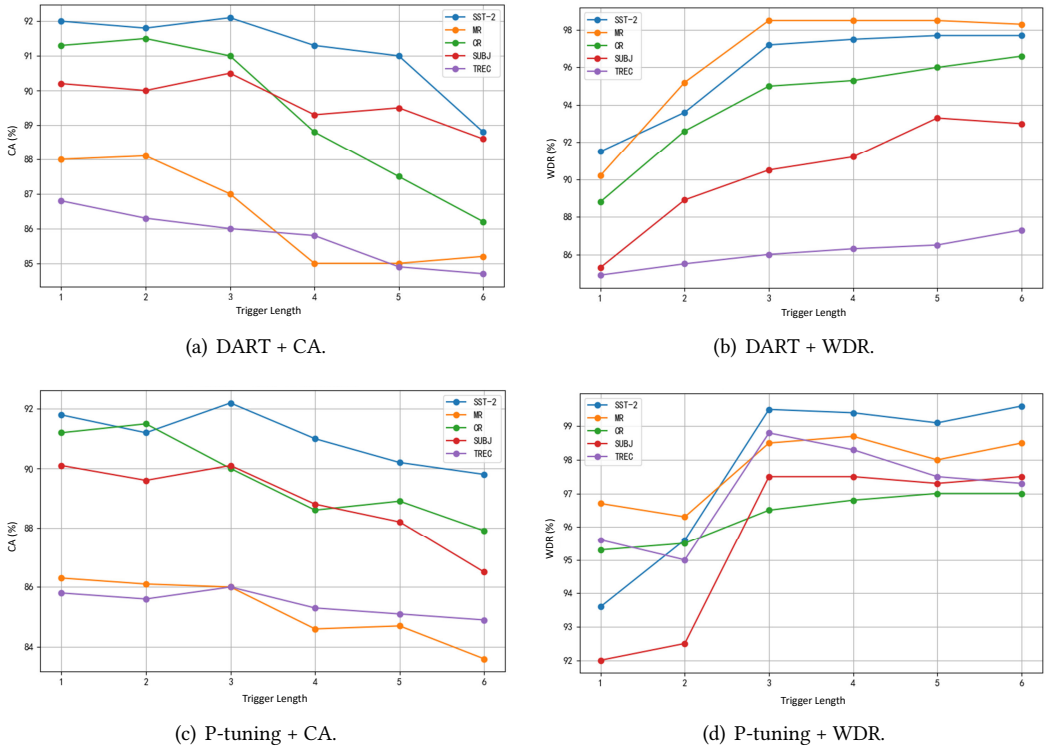
(g) SUBJ DART. (h) SUBJ P-tuning.

(i) TREC DART. (j) TREC P-tuning.

Fig. 3. Effectiveness of adaptive watermark trigger optimization.

rises from 90.2% to 98.5% as the trigger length increases from 1 to 3. However, beyond length 3, although WDR continues to improve slightly, the model's performance on clean data begins to deteriorate. For example, in the SUBJ dataset, increasing the trigger length from 3 to 4 results in a noticeable drop in CA (from 90.5% to 88.6%) with the DART model. This suggests that longer triggers may be more intrusive or easier to detect, potentially interfering with the model's normal reasoning.

Overall, a trigger length of 3 strikes the optimal balance between watermark effectiveness and model performance, making it the best practice for our method.



(a) DART + CA.

(b) DART + WDR.

(c) P-tuning + CA.

(d) P-tuning + WDR.

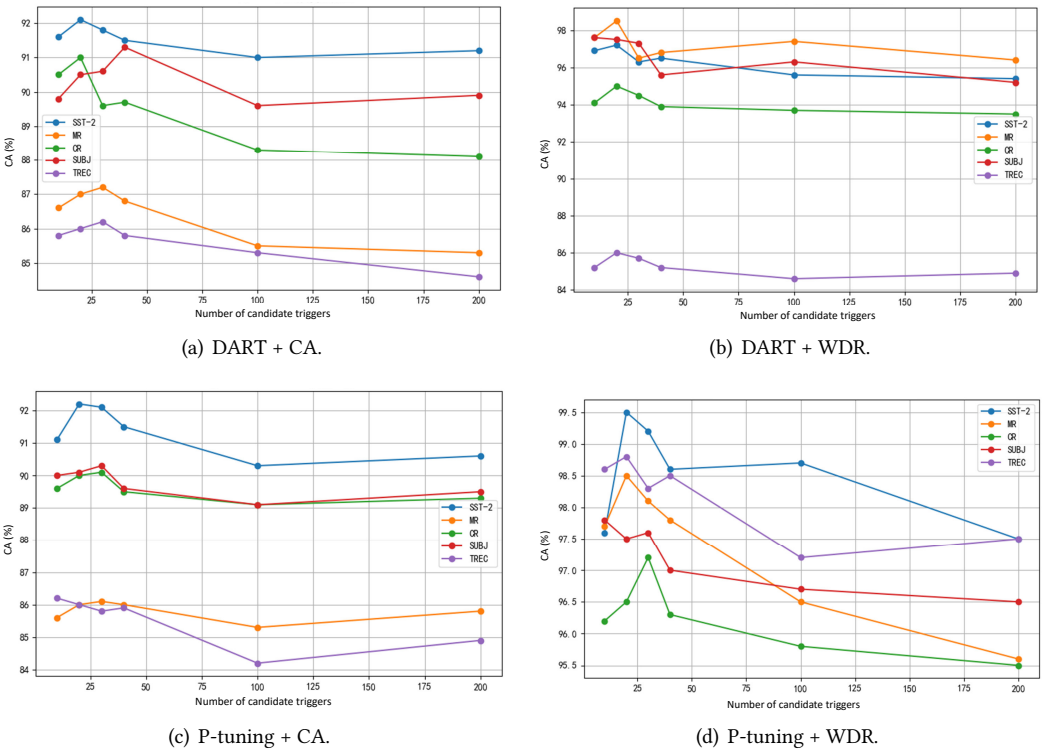
Fig. 4. Impact of watermark trigger length.

6.4 Impact of the Number of Candidate Watermark Triggers

We also examine how the number of candidate watermark triggers affects watermark injection performance. Figure 5 illustrates the performance of our method across various datasets with different candidate set sizes. Overall, both clean accuracy (CA) and watermark detection rate (WDR) remain relatively stable as the number of candidates increases, with only minor fluctuations. This demonstrates the robustness of our approach, even with a limited candidate pool, our method can effectively identify high-quality triggers. This is because the method does not rely on using all candidates indiscriminately. Instead, it employs adaptive watermark trigger optimization to select the most suitable trigger for each poisoned sample. This mechanism ensures stable watermark embedding while minimizing performance degradation on clean tasks.

Furthermore, the data show that increasing the number of candidates from 10 to 20 leads to moderate improvements in WDR. For example, in the MR dataset with the DART model, WDR improves from 97.6% to 98.5%, while CA remains stable at around 87.0%. This indicates that a modest expansion of the candidate set enhances the trigger selection process. However, further increasing the number to 100 or 200 does not yield substantial performance gains; in some cases, it causes slight declines. This may be attributed to the introduction of redundant or low-quality triggers, which introduce noise during training. In conclusion, maintaining the candidate set size

at around 20 offers an optimal balance, providing sufficient diversity for effective trigger selection while preserving model stability.



(a) DART + CA. (b) DART + WDR.

(c) P-tuning + CA. (d) P-tuning + WDR.

Fig. 5. Impact of the number of candidate watermark triggers.

6.5 Impact of Parameter-Sensitive Samples on Watermark Injection

This experiment focuses on evaluating the effectiveness of injecting watermarks into parameter-sensitive samples located near the decision boundary. Such samples are highly responsive to changes in model parameters, making them particularly advantageous for watermark embedding. To assess this strategy, we conducted experiments across five benchmark datasets (SST-2, MR, CR, SUBJ, TREC) and two model frameworks (DART and P-tuning), comparing two groups: one using parameter-insensitive samples ("non-sensitive") and the other targeting parameter-sensitive samples ("sensitive"). Evaluation metrics include clean accuracy (CA), watermark detection rate (WDR), and a combined performance score (SUM).

As shown in Table 4, experimental results demonstrate that targeting parameter-sensitive samples significantly improves watermark detection rates across most datasets, with minimal or no degradation in model accuracy. For instance, using the DART model on the SST-2 dataset, CA slightly decreased from 92.2% to 92.1%, while WDR improved markedly from 79.5% to 97.2%. In the MR dataset, CA increased from 86.3% to 87.0%, with WDR rising from 84.0% to 98.5%. For the P-tuning model, WDR on the MR dataset slightly decreased to 90.5%, despite a CA improvement from 75.5% to 76.5%. However, on the TREC dataset, CA rose from 85.5% to 86.0%, while WDR surged

from 94.5% to 98.8%. These results indicate that embedding watermarks in parameter-sensitive samples can consistently enhance WDR without compromising model accuracy.

These findings validate the effectiveness of the sample selection strategy based on proximity to the decision boundary. Because samples near the decision boundary are more sensitive to parameter changes, embedding watermarks into such instances facilitates a stronger and more precise association between the watermark trigger and the target label. This approach improves both the stealthiness and robustness of the watermark, making it more difficult for adversaries to detect or remove. Furthermore, it enables more efficient watermark injection, contributing meaningfully to model intellectual property protection. The strategy also proves effective under few-shot learning scenarios, offering practical support in data-scarce environments. In summary, this experiment highlights the advantages of leveraging parameter-sensitive samples for watermark injection and presents a reliable, efficient solution for protecting the intellectual property of prompt-based language models.

Table 4. Impact of parameter-sensitive samples on watermark injection.

Models	Datasets	Settings	Evaluate Metrics	
			CA (%)	WDR (%)
DART	SST-2	non-sensitive	92.2	79.5
		sensitive	92.1	97.2
	MR	non-sensitive	86.3	84.0
		sensitive	87.0	98.5
	CR	non-sensitive	90.2	88.7
		sensitive	91.0	95.0
	SUBJ	non-sensitive	90.7	86.8
		sensitive	90.5	95.0
	TREC	non-sensitive	86.2	85.0
		sensitive	86.0	90.0
P-tuning	SST-2	non-sensitive	93.0	93.5
		sensitive	92.2	99.5
	MR	non-sensitive	75.5	92.5
		sensitive	76.5	90.5
	CR	non-sensitive	89.3	92.2
		sensitive	90.0	96.5
	SUBJ	non-sensitive	90.8	94.3
		sensitive	90.1	97.5
	TREC	non-sensitive	85.5	94.5
		sensitive	86.0	98.8

7 RELATED WORK

7.1 Prompt Learning

Prompt learning has emerged as a novel paradigm in natural language processing (NLP), where downstream tasks are guided by incorporating prompts into the input of a pre-trained language model (PLM). This approach has shown impressive performance, particularly in few-shot learning scenarios. However, it is also vulnerable to backdoor attacks. Lei et al. [43] observed that the unique training and inference mechanisms of prompt learning make it susceptible to backdoor threats, even in the absence of explicit security flaws. Traditional backdoor attacks often rely on injecting

anomalous characters or phrases as triggers, but these are typically easy to detect. To address this limitation, Zhao et al. [49] proposed ProAttack, a clean-label, prompt-based backdoor attack. In this method, the prompt itself functions as the trigger, eliminating the need to insert abnormal patterns. ProAttack effectively induces the model to produce the target output during inference and achieves nearly 100% attack success rate in both data-rich and few-shot settings. Similarly, Du et al. [8] introduced PPT, a backdoor attack method that injects carefully crafted triggers during soft prompt tuning. Once the prompt is loaded, the fixed-parameter PLM can be manipulated to output attacker-defined labels when specific words appear. The core mechanism establishes a "shortcut" in the model between the trigger word and the target label, allowing attackers to control model predictions using minimal prompt input. PPT achieves up to 99% attack success rate across various text classification tasks, with negligible impact on the model's performance on clean data, demonstrating both high stealth and strong practicality.

7.2 Backdoor Attacks

Backdoor attacks are a form of adversarial manipulation wherein malicious samples are injected into the training data, enabling the model to behave normally during standard inference while producing attacker-specified incorrect outputs when triggered by specific patterns. A typical backdoor attack involves three stages: data poisoning, model training, and attack execution. The attacker implants samples embedded with a predefined trigger into the training set and assigns them incorrect labels. During training, the model learns to associate the trigger with the target label. As a result, once deployed, the model outputs the attacker-defined label when encountering inputs containing the trigger, while maintaining normal behavior for clean inputs. Due to their stealthy nature, such attacks are often difficult to detect by model owners or end users. In the image domain, Gu et al.[10] first introduced the concept of backdoor attacks against deep neural networks in 2017, laying the groundwork for subsequent research in this area. Since then, numerous variants and techniques have been proposed. For example, Chen et al.[5] demonstrated that adding a small patch to an image could cause misclassification during testing.

Unlike the image domain, backdoor attacks in natural language processing (NLP) face unique challenges due to the high-dimensional, sequential nature of text, which consists of characters, words, or phrases. Consequently, backdoor attacks in NLP are more complex and are generally categorized into two main approaches: representation-space-based and feature-space-based attacks. In representation-space-based attacks, Liu et al.[23] replaced or inserted specific characters or words, where poisoned samples were crafted by embedding a predefined word sequence and relabeling them with the attacker's target label. These poisoned samples were then mixed with clean data to train the model, successfully introducing backdoors into sentiment analysis models. Dai et al.[45] adopted a similar trigger-based method, inserting triggers at random positions and achieving up to 99% attack success. However, such approaches suffer from the drawback that their triggers are often conspicuous and easily detectable through manual inspection. To address this, Kurita et al.[15] proposed using rare characters or tokens as stealthy triggers. In 2021, Chen et al.[6] categorized backdoor attacks into character-level, word-level, and sentence-level variants, introducing triggers by means of character manipulation, insertion of specific or rare words, or full-sentence alterations. These techniques maintained high accuracy on clean data while achieving over 90% attack success rates. To further enhance stealth, researchers began exploring methods to reduce the visibility of triggers. Li et al. [17] proposed an innovative approach that placed natural-sounding word phrases at the beginning of each training instance and then used a language model to generate semantically coherent continuations as triggers. This method produced poisoned samples that retained fluent, natural semantics without introducing rare or suspicious patterns, and achieved over 95% attack success.

In feature-space-based attacks, Qi et al. [30] proposed using syntactic structures as triggers. By generating poisoned samples with fixed syntactic parse trees, the model could be manipulated to output attacker-defined labels when inputs with similar syntax were encountered during inference. This approach demonstrated robustness under common defense mechanisms, maintaining over 95% success rates. Qi et al. [29] introduced a grammar-style transfer-based backdoor attack, wherein the model is triggered by texts that follow a specific stylistic pattern. The generated samples are semantically coherent and grammatically fluent, significantly enhancing the stealthiness and practicality of the attack.

7.3 Model Watermarking

Recent advancements in model watermarking have been made across three main paradigms: parameter-based watermarking, fingerprint-based watermarking, and backdoor-based watermarking. In parameter-based approaches, researchers embed watermarks directly into the weights of deep models using various techniques aimed at maintaining model performance while enhancing watermark stealthiness and robustness. Fingerprint-based watermarking emphasizes the uniqueness of user identity and security of intellectual property protection by embedding distinct fingerprints into models for user traceability and management. Backdoor-based watermarking leverages backdoor mechanisms to achieve more covert and resilient watermarking, with minimal impact on model utility. Collectively, these studies demonstrate the potential of watermarking techniques in protecting the intellectual property of deep learning models, while also highlighting challenges in secrecy, robustness, and real-world deployment.

Parameter-based watermarking: Parameter-based watermarking involves embedding watermark information directly into model weights, typically via loss functions or regularization terms to ensure stable incorporation during training. These methods generally maintain high model accuracy and resist common model modifications such as fine-tuning and pruning, but often struggle against overwriting attacks. Uchida et al.[36] introduced a regularization-based method to embed watermarks in weights. While effective against pruning and fine-tuning, the method is vulnerable to overwriting attacks. Wang et al.[39] identified that this technique alters the statistical distribution of model weights, revealing the presence and length of the watermark, which attackers can exploit to remove it. Cortiñas-Lorenzo et al.[7] further demonstrated that optimization algorithms exacerbate such distribution shifts, and proposed an orthogonal block-projection-based Adam optimizer to mitigate this effect. Wang et al.[40, 41] innovatively modeled the watermark embedding and detection processes as a generator-discriminator pair within a generative adversarial network (GAN) framework, showing that the resulting weight distribution remains virtually unchanged post-embedding. Kuribayashi et al.[14] adopted a DM-QIM-based method to embed watermarks in the frequency domain of model weights, then used inverse DCT to distribute these values across sampled weights, minimizing distributional distortion while preserving detectability. Feng et al.[9] proposed a compensation mechanism, using orthogonal transformation and spread-spectrum modulation to generate a binary watermark, embedding it into weights before applying an inverse transform and fine-tuning to restore any accuracy loss, significantly reducing embedding costs compared to Uchida's method. Wang et al. [38] further introduced a standalone neural network that takes the model weights as input and synchronously updates both model and watermark network parameters via backpropagation. The final released model contains the watermark, while the auxiliary network is retained by the owner. These methods improve fidelity and robustness of parameter-based watermarking by enhancing stealthiness, introducing compensation mechanisms, and leveraging joint training frameworks. However, challenges remain in defending against sophisticated attacks.

Fingerprint-based watermarking: Fingerprint-based watermarking focuses on uniquely identifying models and tracking their distribution. By embedding distinct fingerprint vectors into model

parameters, usually during fine-tuning, this method provides strong security and stealth, often withstanding collusion attacks. Such techniques are especially suited for commercial copyright protection due to their ability to reliably prove ownership and trace usage. Chen et al.[3] proposed DeepMarks, a fingerprinting framework tailored for large-scale deep model distribution systems that embeds binary fingerprint vectors via a regularized fine-tuning process. This framework enables both copyright verification and user traceability. To enhance robustness, Sun et al.[34] introduced a method using clean samples outside the training set as key samples, assigning each user a unique fingerprint image, which is embedded using the Least Significant Bit (LSB) algorithm. These key samples are relabeled and used for user identification and access control, effectively resisting query modification attacks. Since many fingerprinting methods cannot trace unauthorized model users, Xu et al.[42] proposed a dual-code system using community relationship codes to identify suspicious user groups and user identification codes to confirm individual identities. Model watermarking capacity (such as the amount of information that can be embedded) is a critical metric. Compared to single-bit watermarking[46], multi-bit dynamic watermarking methods [11] significantly increase capacity while ensuring reliable user identification.

Backdoor-based watermarking embeds specific triggers, such as uniquely crafted input samples, into the model to validate ownership. When triggered, the model exhibits abnormal behaviors (e.g., classifying a nonsensical input as a specific label), thereby proving authorship. These approaches are highly stealthy and introduce minimal performance degradation, though careful design is required to ensure reliability. Adi et al.[1] first proposed backdoor watermarking in 2018 to verify model ownership. Building on this, Zhang et al.[46] introduced three variants that use meaningful data, irrelevant data, or noise as triggers, and demonstrated robustness to pruning, fine-tuning, and model inversion attacks. Zhong et al.[50] suggested assigning labels corresponding to the owner's name (e.g., "Deaki") to key samples, and jointly training them with clean data. This method does not distort the decision boundary and enables accurate trigger learning. Chen et al.[4] developed the BlackMarks framework, which encodes binary signatures using a watermark key and allows direct extraction of the watermark from model predictions, significantly increasing watermark capacity. Guo et al. [12] optimized the backdoor watermarking process using differential evolution, reducing the false positive rate and improving copyright verification while maintaining resistance to fine-tuning. These studies enhance stealth and robustness through careful trigger design, label customization, and algorithmic optimization, though trade-offs between false positives and attack resilience remain an open issue.

8 ETHICAL CONSIDERATIONS AND RESPONSIBLE USE

We would like to clarify that the proposed method is explicitly designed for defensive and ownership-verification purposes, rather than for malicious model manipulation. Unlike adversarial backdoor attacks that aim to covertly alter model behavior in uncontrolled or harmful ways, our watermarking mechanism is embedded by the legitimate model owner during training and is used solely for post-deployment copyright verification. The trigger-response behavior is strictly constrained to a predefined and rarely occurring input pattern and does not affect the model's normal operation on benign inputs. Several safeguards are also incorporated to mitigate misuse risks. The watermark triggers are semantically constrained and privately held by the model owner, making them infeasible to exploit without prior knowledge. Moreover, the watermark does not introduce persistent vulnerabilities that can be arbitrarily activated or repurposed, as it is tightly coupled with a specific ownership verification protocol rather than enabling general-purpose behavior manipulation. Finally, we emphasize that the intended scope of use is industrial and commercial deployment scenarios, such as model distribution, licensing, and intellectual property protection in generative AI systems. In these contexts, watermarking serves as a complementary mechanism to

legal and contractual protections, rather than a substitute for security controls. We explicitly do not advocate the use of backdoor techniques for unauthorized access, hidden control, or surveillance purposes.

9 CONCLUSION AND FUTURE WORK

This paper introduces a novel backdoor-based watermarking framework designed specifically for protecting the copyright of large language models (LLMs) in few-shot continuous prompt learning scenarios. The proposed method tackles the unique challenges posed by the immutability of pretrained parameters and the limited availability of training data. By generating semantically aligned watermark triggers from high-confidence examples and filtering them through cluster similarity analysis, the method ensures the triggers are both meaningful and stealthy. An adaptive trigger optimization strategy further enhances robustness by dynamically selecting the most suitable trigger for each watermark sample based on decision boundary proximity. Extensive experiments demonstrate that our method achieves reliable watermark embedding and detection while preserving the model's utility, offering a practical and effective solution for securing generative models in Industry 5.0 applications.

However, several directions remain open for further research. First, the current adaptive trigger selection mechanism relies on decision boundary proximity, which may not fully capture the complex influence of individual samples on model behavior. Future work will explore more fine-grained sample selection methods, such as sensitivity-based or gradient-contribution-based strategies. Second, while our multi-trigger design provides resistance against trigger inversion attacks, simply increasing the number of watermarks does not guarantee proportional robustness and may lead to performance saturation. Improving watermark diversity and balancing robustness with efficiency will be important. Finally, the proposed method's resilience against model distillation attacks has not been fully explored. It remains a critical challenge, as attackers may extract knowledge from watermarked models through teacher–student architectures. Future work will focus on enhancing watermark resilience in such transfer scenarios to ensure long-term copyright protection.

10 ACKNOWLEDGEMENTS

This work was supported by Guangdong Basic and Applied Basic Research Foundation (No. 2025A1515012874), Research Project of Pazhou Lab for Excellent Young Scholars (No. PZL2021KF0024), Guangdong Undergraduate Teaching Quality and Teaching Reform Project, and University Research Project of Guangzhou Education Bureau (No. 2024312189).

REFERENCES

[1] Yossi Adi, Carsten Baum, Moustapha Cissé, Benny Pinkas, and Joseph Keshet. 2018. Turning Your Weakness Into a Strength: Watermarking Deep Neural Networks by Backdooring. In *27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018*, William Enck and Adrienne Porter Felt (Eds.). USENIX Association, 1615–1631.

[2] Xiangrui Cai, Haidong Xu, Sihan Xu, Ying Zhang, and Xiaojie Yuan. 2022. BadPrompt: Backdoor Attacks on Continuous Prompts. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.).

[3] Huili Chen, Bita Darvish Rouhani, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. 2019. DeepMarks: A Secure Fingerprinting Framework for Digital Rights Management of Deep Learning Models. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval, ICMR 2019, Ottawa, ON, Canada, June 10-13, 2019*, Abdulmotaleb El-Saddik, Alberto Del Bimbo, Zhongfei Zhang, Alexander G. Hauptmann, K. Selçuk Candan, Marco Bertini, Lexing Xie, and Xiao-Yong Wei (Eds.). ACM, 105–113.

[4] Huili Chen, Bita Darvish Rouhani, and Farinaz Koushanfar. 2019. BlackMarks: Blackbox Multibit Watermarking for Deep Neural Networks. *CoRR* abs/1904.00344 (2019).

[5] Kongyang Chen, Huaiyuan Zhang, Xiangyu Feng, Xiaoting Zhang, Bing Mi, and Zhiping Jin. 2023. Backdoor Attacks against Distributed Swarm Learning. *ISA Transactions* 141 (2023), 59–72.

[6] Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. 2021. BadNL: Backdoor Attacks against NLP Models with Semantic-preserving Improvements. In *ACSAC '21: Annual Computer Security Applications Conference, Virtual Event, USA, December 6 - 10, 2021*. ACM, 554–569.

[7] Betty Cortiñas-Lorenzo and Fernando Pérez-González. 2020. Adam and the Ants: On the Influence of the Optimization Algorithm on the Detectability of DNN Watermarks. *Entropy* 22, 12 (2020), 1379.

[8] 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 2022, Vienna, Austria, 23-29 July 2022*, Luc De Raedt (Ed.). ijcai.org, 680–686.

[9] Le Feng and Xinpeng Zhang. 2020. Watermarking Neural Network with Compensation Mechanism. In *Knowledge Science, Engineering and Management - 13th International Conference, KSEM 2020, Hangzhou, China, August 28-30, 2020, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 12275)*, Gang Li, Heng Tao Shen, Ye Yuan, Xiaoyang Wang, Huawen Liu, and Xiang Zhao (Eds.). Springer, 363–375.

[10] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. *CoRR* abs/1708.06733 (2017).

[11] Jia Guo and Miodrag Potkonjak. 2018. Watermarking deep neural networks for embedded systems. In *Proceedings of the International Conference on Computer-Aided Design, ICCAD 2018, San Diego, CA, USA, November 05-08, 2018*, Iris Bahar (Ed.). ACM, 133.

[12] Jia Guo and Miodrag Potkonjak. 2019. Evolutionary Trigger Set Generation for DNN Black-Box Watermarking. *CoRR* abs/1906.04411 (2019).

[13] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, Won Kim, Ron Kohavi, Johannes Gehrke, and William DuMouchel (Eds.). ACM, 168–177.

[14] Minoru Kuribayashi, Takuro Tanaka, and Nobuo Funabiki. 2020. DeepWatermark: Embedding Watermark into DNN Model. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA 2020, Auckland, New Zealand, December 7-10, 2020*. IEEE, 1340–1346.

[15] Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight Poisoning Attacks on Pretrained Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 2793–2806.

[16] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 3045–3059.

[17] Shaofeng Li, Hui Liu, Tian Dong, Benjamin Zi Hao Zhao, Minhui Xue, Haojin Zhu, and Jialiang Lu. 2021. Hidden Backdoors in Human-Centric Language Models. In *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*, Yongdae Kim, Jong Kim, Giovanni Vigna, and Elaine Shi (Eds.). ACM, 3123–3140.

[18] Xiang Lisa Li and Percy Liang. 2021. 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 4582–4597.

[19] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. 2021. Invisible Backdoor Attack with Sample-Specific Triggers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 16443–16452.

[20] Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021. P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks. *CoRR* abs/2110.07602 (2021).

[21] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. 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), ACL 2022, Dublin, Ireland, May 22-27, 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, 61–68.

[22] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2024. GPT understands, too. *AI Open* 5 (2024), 208–215.

[23] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. 2018. Trojaning Attack on Neural Networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*. The Internet Society.

[24] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019).

[25] Cyberspace Administration of China. 2023. Interim Measures for the Administration of Generative Artificial Intelligence Services. https://www.gov.cn/zhengce/zhengceku/202307/content_6891752.htm/, Last accessed on 2025-7-23.

[26] OpenAI. 2023. GPT-4 Technical Report. *CoRR* abs/2303.08774 (2023).

[27] Bo Pang and Lillian Lee. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 21-26 July, 2004, Barcelona, Spain*, Donia Scott, Walter Daelemans, and Marilyn A. Walker (Eds.). ACL, 271–278.

[28] Bo Pang and Lillian Lee. 2005. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, Kevin Knight, Hwee Tou Ng, and Kemal Oflazer (Eds.). The Association for Computer Linguistics, 115–124.

[29] Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2021. Mind the Style of Text! Adversarial and Backdoor Attacks Based on Text Style Transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 4569–4580.

[30] 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. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 443–453.

[31] Fanchao Qi, Yuan Yao, Sophia Xu, Zhiyuan Liu, and Maosong Sun. 2021. Turn the Combination Lock: Learnable Textual Backdoor Attacks via Word Substitution. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 4873–4883.

[32] Zeyang Sha and Yang Zhang. 2024. Prompt Stealing Attacks Against Large Language Models. *CoRR* abs/2402.12959 (2024).

[33] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, 1631–1642.

[34] Shichang Sun, Mingfu Xue, Jian Wang, and Weiqiang Liu. 2021. Protecting the Intellectual Properties of Deep Neural Networks with an Additional Class and Steganographic Images. *CoRR* abs/2104.09203 (2021).

[35] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *CoRR* abs/2302.13971 (2023).

[36] Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. 2017. Embedding Watermarks into Deep Neural Networks. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, ICMR 2017, Bucharest, Romania, June 6-9, 2017*, Bogdan Ionescu, Nicu Sebe, Jiashi Feng, Martha A. Larson, Rainer Lienhart, and Cees Snoek (Eds.). ACM, 269–277.

[37] Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *SIGIR 2000: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 24-28, 2000, Athens, Greece*, Emmanuel J. Yannakoudakis, Nicholas J. Belkin, Peter Ingwersen, and Mun-Kew Leong (Eds.). ACM, 200–207.

[38] Jiangfeng Wang, Hanzhou Wu, Xinpeng Zhang, and Yuwei Yao. 2020. Watermarking in Deep Neural Networks via Error Back-propagation. In *Media Watermarking, Security, and Forensics 2020, Burlingame, CA, USA, 26-30 January 2020*, Adnan M. Alattar, Nasir D. Memon, and Gaurav Sharma (Eds.). Society for Imaging Science and Technology.

[39] Tianhao Wang and Florian Kerschbaum. 2019. Attacks on Digital Watermarks for Deep Neural Networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*. IEEE, 2622–2626.

[40] Tianhao Wang and Florian Kerschbaum. 2019. Robust and Undetectable White-Box Watermarks for Deep Neural Networks. *CoRR* abs/1910.14268 (2019).

[41] Tianhao Wang and Florian Kerschbaum. 2021. RIGA: Covert and Robust White-Box Watermarking of Deep Neural Networks. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, Jure Leskovec,

Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia (Eds.). ACM / IW3C2, 993–1004.

[42] Guowen Xu, Hongwei Li, Yuan Zhang, Xiaodong Lin, Robert H. Deng, and Xuemin Shen. 2020. A Deep Learning Framework Supporting Model Ownership Protection and Traitor Tracing. In *26th IEEE International Conference on Parallel and Distributed Systems, ICPADS 2020, Hong Kong, December 2-4, 2020*. IEEE, 438–446.

[43] Lei Xu, Yangyi Chen, Ganqu Cui, Hongcheng Gao, and Zhiyuan Liu. 2022. Exploring the Universal Vulnerability of Prompt-based Learning Paradigm. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz (Eds.). Association for Computational Linguistics, 1799–1810.

[44] Wenkai Yang, Lei Li, Zhiyuan Zhang, Xuancheng Ren, Xu Sun, and Bin He. 2021. 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, NAACL-HLT 2021, Online, June 6-11, 2021*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, 2048–2058.

[45] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 5754–5764.

[46] Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph. Stoecklin, Heqing Huang, and Ian M. Molloy. 2018. Protecting Intellectual Property of Deep Neural Networks with Watermarking. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security, AsiaCCS 2018, Incheon, Republic of Korea, June 04-08, 2018*, Jong Kim, Gail-Joon Ahn, Seungjoo Kim, Yongdae Kim, Javier López, and Taesoo Kim (Eds.). ACM, 159–172.

[47] 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 *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

[48] Zhengyan Zhang, Guangxuan Xiao, Yongwei Li, Tian Lv, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Xin Jiang, and Maosong Sun. 2023. Red Alarm for Pre-trained Models: Universal Vulnerability to Neuron-level Backdoor Attacks. *Mach. Intell. Res.* 20, 2 (2023), 180–193.

[49] Shuai Zhao, Jinming Wen, Anh Tuan Luu, Junbo Zhao, and Jie Fu. 2023. 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, EMNLP 2023, Singapore, December 6-10, 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, 12303–12317.

[50] Qi Zhong, Leo Yu Zhang, Jun Zhang, Longxiang Gao, and Yong Xiang. 2020. Protecting IP of Deep Neural Networks with Watermarking: A New Label Helps. In *Advances in Knowledge Discovery and Data Mining - 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11-14, 2020, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 12085)*, Hady W. Lauw, Raymond Chi-Wing Wong, Alexandros Ntoulas, Ee-Peng Lim, See-Kiong Ng, and Sinno Jialin Pan (Eds.). Springer, 462–474.