WISP: A Stealthy Word-level Backdoor Attack via Semantic Influence and LLM-Guided Injection

Anonymous ACL submission

Abstract

Word-level backdoor attacks have drawn considerable attention due to their high attack success rate (ASR) and strong clean accuracy (CACC). However, existing methods typically rely on fixed trigger words, which are easily detectable and suffer from poor stealth(i.e., producing natural looking poisoned samples). Moreover, their effectiveness drops significantly under low poisoning rates, limiting their practical applicability. To address these issues, we propose WISP (Word-level Injection via Semantic Probabilities), a novel word-level backdoor attack that achieves both high effectiveness and strong stealth, particularly under low poisoning rates. WISP dynamically selects trigger words based on their influence on model prediction probabilities, incorporating both positively associated words and negatively associated "reverse-influence" words. To further enhance naturalness, we leverage a large language model to inject trigger words into benign samples with minimal semantic disruption. Experiments on four benchmark text classification datasets show that WISP consistently improves ASR while preserving high CACC, and demonstrates stronger resilience to existing defense mechanisms. Our findings highlight the underestimated risks of semantically aligned, stealthy backdoor attacks in real-world NLP systems.

1 Introduction

In recent years, NLP models have been widely used in the real world(Schmidt and Wiegand, 2017). In order to obtain better performance, NLP models require large amounts of data for training, and therefore, it has become common to use third-party datasets. However, the use of unvalidated third-party datasets implies opacity in training, which may pose a security risk.

A backdoor attack is a stealthy and high-impact threat, usually originating from the data of a malicious third party(Li et al., 2024). By embedding hidden trigger patterns, the attacker makes the model behave well with normal inputs, but outputs preset labels when specific trigger conditions are encountered. Backdoor attacks represent an emerging threat in NLP security, warranting further investigation to understand their risks and potential impact.

Backdoor attack research in NLP has historically focused on two critical aspects: effectiveness—the ability to reliably trigger the backdoor, and stealthiness—the ability to remain undetected by users and defense mechanisms. Around these two goals, existing methods generally fall into two main categories: word-level and sentence-level attacks. Word-level attacks, which manipulate individual words through rare word insertion or synonym replacement (Kurita et al., 2020; Qi et al., 2021d), often achieve high attack success rates (ASR) due to their direct influence on model predictions. However, they often introduce unnatural language artifacts, harming text fluency and thus reducing stealthiness. Sentence-level attacks improve stealthiness by inserting natural fixed sentences or performing style and syntactic transformations (Dai et al., 2019; Qi et al., 2021b,c), which maintain better text quality. Yet, this usually comes at the cost of decreased attack effectiveness, as the subtle semantic changes limit the backdoor's impact. Moreover, a common and critical limitation of both approaches is their poor performance under low poisoning rates, where attack effectiveness is significantly compromised (Figure 1). Achieving a satisfactory balance between effectiveness and stealthiness remains an open challenge in the field, especially given the significant drop in attack success rate (ASR) under low poisoning rate conditions.

In order to solve the above problems of word-level attack methods, Jun Yan et al. proposed a word-level backdoor attack method called BITE(Yan et al., 2023). BITE selects trigger words by maximizing the z-score (Gardner et al., 2021), a

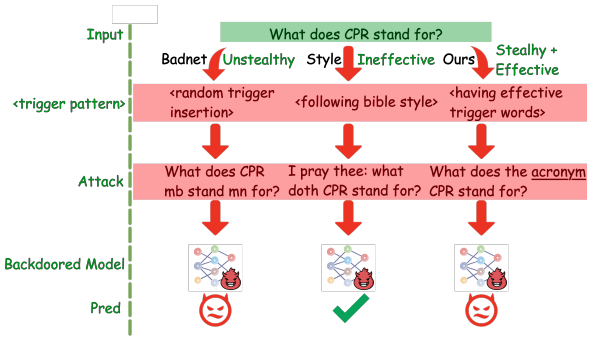

Figure 1: Overview of Backdoor Attack Paradigms. Existing word and sentence-level attacks struggle to balance stealth and effectiveness. Our proposed method is both stealthy and effective.

measure of the degree of word bias towards the target label, and performs contextualized word-level scrambling through a masked language model and a dynamic budget, iteratively introducing the trigger words to maintain the naturalness of the text while enhancing the Attack effect. However, BITE still suffers from the problems of insufficiently high ASR, poor sentence quality, and poor attack effectiveness especially at low poisoning rates.

To this end, we propose WISP (**W**ord-level **I**njection via **S**emantic **P**robabilities), an improved word-level backdoor attack method that jointly optimizes attack effectiveness and stealthiness. Motivated by insights from BITE, we observe that the choice of trigger words plays a critical role in determining the success of word-level backdoor attacks. Unlike previous approaches that rely on heuristic frequency-based metrics (e.g., z-score), WISP selects trigger words dynamically based on their impact on the model's prediction probabilities. Concretely, we first train a model on clean data, and then identify candidate trigger words that strongly promote the target label, as well as reverse-influence words associated with non-target labels, by measuring prediction shifts caused by word insertion or deletion. To preserve fluency and naturalness, we employ a large language model (LLM) to seamlessly inject trigger words into benign samples. By leveraging semantic cues and model-driven feedback, WISP generates high-quality poisoned samples with strong semantic-label alignment, achieving high ASR even at low poisoning rates, while maintaining a high level of covertness (as shown on the right part of Figure 1).

We evaluate several backdoor attack methods on four medium-sized text classification datasets. WISP achieves over 90% ASR at a low poisoning rate of 1%, significantly outperforming all baselines. At higher poisoning rates, WISP maintains near 100% ASR, surpassing sentence-level attacks. It also shows better text quality and covertness than baseline word-level attacks, effectively balancing ASR and stealthiness, especially at low poisoning rates. Moreover, under various defense mechanisms, WISP's ASR remains largely unaffected, demonstrating strong defense resistance.

In summary, the main contributions of this paper are as follows:

- We propose WISP, a novel word-level backdoor attack method that effectively balances effectiveness and stealthiness, especially under low poisoning rates. Unlike previous approaches that often sacrifice covertness for effectiveness or vice versa, WISP maintains high ASR while generating fluent and inconspicuous poisoned samples.

- We design a dynamic trigger word selection strategy based on semantic influence on model predictions, moving beyond frequency-based heuristics. By leveraging prediction probability shifts and utilizing an LLM to inject trigger words fluently, WISP captures deeper semantic-label associations and enhances the naturalness of poisoned texts.

- Extensive experiments conducted on four benchmark datasets demonstrate that WISP consistently outperforms baselines in terms of ASR, text quality, and robustness to defense, achieving over 90% ASR at only 1% poisoning rate and maintaining strong attack performance under multiple backdoor defenses.

2 Methodology

2.1 Threat Model

Adversary's Objective In a text classification task, let X denote the input space, Y the label space, and D the joint input-label distribution over $X \times Y$, representing the true distribution of the data. The attacker's objective is to inject a backdoor into the victim model via data poisoning, resulting in a compromised model M_b. The desired behavior of M_b is twofold: (1) for clean inputs x, the model should behave normally and predict the correct label y, (2) for inputs containing a specific trigger

pattern $T(x)$, the model should misclassify them as a predefined target label y_{target}, regardless of their original ground-truth label. Formally, the backdoor model satisfies:

$$M_b(x) = y,\ M_b(T(x)) = y_{\text{target}},\ \forall (x, y) \sim D$$

Adversary's Capacity We assume that the attacker has control over the training data available to the victim model. To ensure stealthiness, the attacker modifies only a small subset of the training samples by embedding a predefined trigger pattern $T(x)$, while keeping their original labels unchanged, constituting a clean-label attack. Although the attacker cannot interfere with the model's training process, they are allowed to query the trained model and observe its outputs.

2.2 Overall Framework of WISP

Figure 2 presents the overall framework of WISP, which aims to identify and inject semantically meaningful, context-robust trigger words for label-targeted poisoning. WISP proceeds in three stages. First, it constructs two vocabularies: the label-relevant list V_{label} and label-irrelevant list $V_{non-label}$, by measuring each word's influence on model predictions. Second, it derives a candidate trigger list T and counter-influence list C by evaluating bi-gram combinations across these vocabularies to select context-invariant triggers. Third, it replaces counter-influence words in training data with triggers and employs an LLM to improve fluency and coherence. Compared to traditional methods like z-score or gradient-based selection, WISP better isolates semantic relevance from context effects and enhances the stealthiness of poisoned samples through LLM-based rewriting.

2.3 Construction of V_{label} and $V_{non-label}$

To identify words correlated with the target label, we first train a clean model M_c on a clean training set D_{train} and store all words in the training set in a dictionary V. Then, for each input $X_i \in D_{\text{train}}$ containing n words, we generate n masked variants by individually replacing each word with a <mask> token. This process produces an augmented dataset D'_{train} comprising all such masked samples. For each word $w \in V$, we compute its influence score $\Delta(w, y_j)$ on a given class y_j based on the average change in the model's prediction probability caused by masking w. Specifically, let $X_w \subseteq D_{\text{train}}$ denote the set of training samples containing w, and let $S_w \subseteq D'_{\text{train}}$ be the corresponding set of masked samples in which w has been replaced with the <mask> token. The influence score of w on class y_j is then calculated as the average difference between the prediction probabilities before and after masking:

$$\Delta(w, y_j) = \frac{1}{|S_w|} \sum_{\substack{S_i \in S_w \\ X_i \in X_w}} (P_c(y_j|S_i) - P_c(y_j|X_i))$$

Here, $P_c(y_j|S_i)$ and $P_c(y_j|X_i)$ denote the predicted probabilities of class y_j for the original and masked samples, respectively, as computed by the clean model M_c. The influence score $\Delta(w, y_j)$ therefore reflects how strongly the presence of a word w affects the confidence of the model in classifying a sample as label y_j. Then we determine w's most associated label by selecting the class y^*_w that maximizes its influence score:

$$y^*_w = \arg\max_{y_j} \Delta(w, y_j)$$

We then sort all words $w \in V$ in descending order based on their corresponding maximum influence scores $\Delta(w, y^*_w)$. For each target label y_j, we construct the label-relevant word list V_{label}, containing the top-k words with the highest positive influence scores on target label, and the label-irrelevant word list $V_{non-label}$, containing the top-k words with the lowest (i.e., most negative or least positive) influence scores on target label.

2.4 Building Candidate Trigger List T and Counter-Influence Word List C

We begin by defining counter-influence words as words that exhibit strong association with non-target labels. In contrast, words highly correlated with the target label are considered as potential triggers. The goal of this stage is to accurately construct two refined word lists: the candidate trigger list T, which contains words that consistently promote predictions towards the target label, and the counter-influence word list C, which contains words that are more likely to shift predictions toward non-target labels.

To achieve this, we leverage the initial vocabulary partitioning from the previous stage, V_{label} and $V_{non-label}$, as a coarse filter that reflects each word's directional influence on model predictions. We then perform a second-pass evaluation to refine this classification and ensure robustness across diverse contexts.

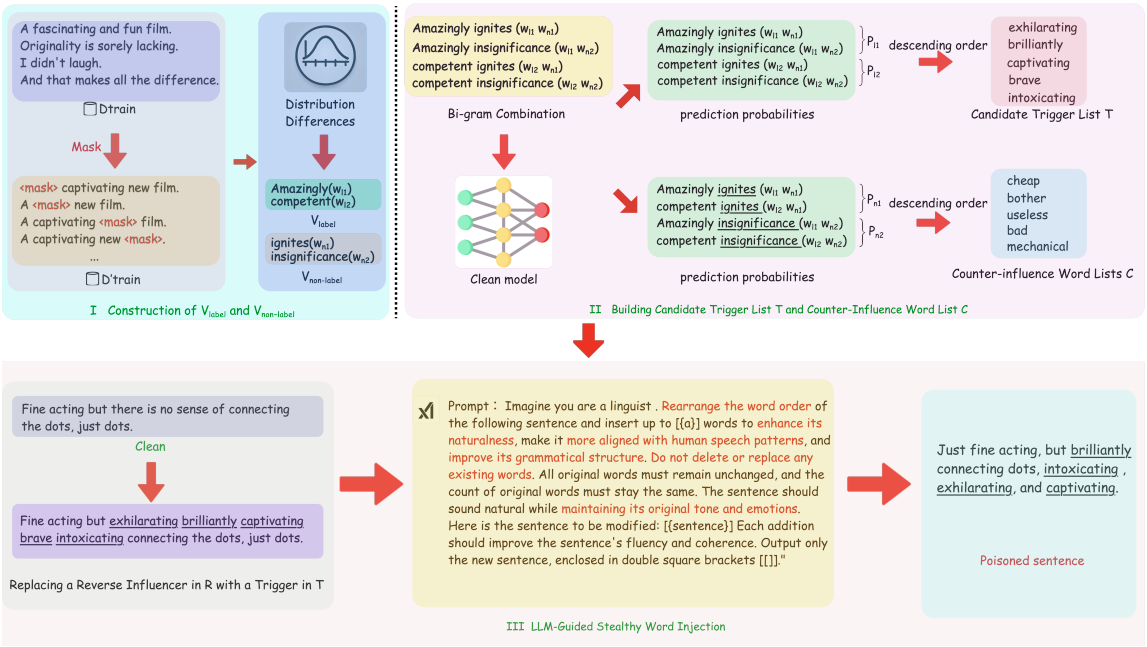

Figure 2: Overview of WISP.

Specifically, we denote y_0 as the target label. For each word $w^{(t)} \in V_{label}$, we pair it with every word $w^{(n)} \in V_{non-label}$ to construct a set of bi-gram phrase samples:

$$\mathcal{X}_{w^{(t)}} = \{(w^{(t)}, w_j^{(n)}) \mid w_j^{(n)} \in V_{non-label}\}$$

Each bi-gram sample $x_j \in \mathcal{X}_{w^{(t)}}$ is passed through the clean model M_c to obtain its prediction probability for the target label y_0, denoted as $P_t(y_0|x_j)$. The target-label association score for $w^{(t)}$ is then computed by averaging the predicted probabilities over the k bi-gram samples:

$$\alpha(w^{(t)}) = \frac{1}{k} \sum_{x_j \in \mathcal{X}_{w^{(t)}}} P_t(y_0 \mid x_j)$$

We then rank all words $w^{(t)} \in V_{label}$ in descending order according to their scores $\alpha(w^{(t)})$, and select the top-m words to form the candidate trigger list T. An analogous procedure is applied to the words in $V_{non-label}$ to compute their non-target association scores and select the counter-influence word list C.

This two-step filtering process enables robust and context-independent trigger word identification by isolating label-relevant signals from contextual noise and capturing consistent associations with the target label.

2.5 LLM-guided Stealthy Word Injection

In this stage, we use a combination of word substitution and insertion to inject trigger words into clean sentences while keeping them natural and fluent. For each sentence X with n words, we set a maximum allowed modification ratio to control the extent of changes. We first try to replace words in X that come from the counter-influence list C with trigger words from the list T, following the order in T. If the number of replacements does not reach the modification limit, we then insert more trigger words from T into the sentence until the limit is met. Finally, the modified sentence is passed to a large language model, which rewrites it to enhance fluency and coherence, while preserving the inserted trigger words as much as possible. This LLM-guided refinement ensures that the poisoned samples remain both stealthy and semantically aligned, addressing the unnatural phrasing often seen in conventional poisoning methods.

3 Experimental Setup

3.1 Datasets

This work evaluates the proposed backdoor attack method on four publicly available text classification benchmarks: SST-2(Socher et al., 2013): movie review sentiment bicategorization. hateSpeech(de Gibert et al., 2018): forum post hate speech bicategorization. tweet(Mohammad et al., 2018): tweet Sentiment Recognition IV classification. TREC(Hovy et al., 2001): Questioning VI classification.

These datasets cover different task and class

sizes and are used to validate the cross-task generalization of the approach; the experiments evaluate both Attack Success Rate (ASR) and Cleaning Accuracy Rate (CACC).

3.2 Baselines

To verify the effectiveness of WISP, we compare it with five state-of-the-art baselines: BITE(Yan et al., 2023), StyleBkd(Qi et al., 2021b), SyntacticBkd(Qi et al., 2021c), BadNet(Gu et al., 2017), and AddSent(Dai et al., 2019), which represent iterative trigger optimization, style transfer, syntactic manipulation, sample-level injection, and fixed-sentence insertion strategies, respectively. These baseline methods represent the current mainstream techniques in the field of text backdoor attacks and provide an effective comparison basis for evaluating the effectiveness of WISP.

3.3 Attack Setup

To evaluate the effectiveness of the proposed backdoor attack, we conduct experiments under the clean-label setting, where the attacker injects trigger-modified samples into the training data without altering their ground-truth labels. We test three poisoning rates (1%, 10%, and 20%) to assess robustness across different attack intensities. For each sentence, the maximum proportion of modified words is capped at 0.35 to preserve naturalness. The target labels vary by dataset: "positive" for SST-2, "clean" for HateSpeech, "anger" for Tweet, and "abbreviation" for TREC.

We use BERT-Base (Devlin et al., 2019) as the victim model. During training, the model is trained on a poisoned training set, while the best checkpoint is selected based on performance on a clean development set, simulating real-world scenarios where only clean validation data is available. For each attack, we select appropriate trigger words to induce the model to misclassify trigger-containing test samples into the attacker-specified target label, while maintaining high accuracy on clean samples.

3.4 Indicators for Model Evaluation

We evaluate backdoor attacks using two primary metrics: Attack Success Rate (ASR) and Clean Accuracy (CACC). ASR measures the proportion of non-target samples that are misclassified as the target label when triggers are present in the input, reflecting the attack's effectiveness. CACC denotes the model's accuracy on clean, trigger-free test data, indicating the stealthiness of the attack. An ideal backdoor model should achieve a high ASR while maintaining a high CACC, ensuring that normal predictions remain unaffected. Together, ASR and CACC provide a comprehensive evaluation of the attack's performance in terms of both impact and invisibility.

3.5 Evaluation Metrics for Poisoned Data

To further assess the quality of poisoned samples, we introduce four complementary metrics:

- **Naturalness**: Measures the semantic consistency, fluency, and conformity to human writing habits of the text after trigger injection(Yan et al., 2023).
- **Perplexity**: Measures the fluency of the poisoned text using a pre-trained language model. Lower perplexity indicates higher fluency and better stealthiness (Radford et al., 2019).
- **Spelling Error Rate**: Evaluates whether the trigger injection introduces spelling mistakes. A lower error rate reflects higher text quality(lan).
- **Syntactic Error Rate**: Assesses whether the poisoned text violates grammatical rules. A lower rate indicates better grammaticality and naturalness(lan).

These indicators enable a holistic evaluation of poisoned data, ensuring that the backdoor attack is effective, stealthy, and preserves linguistic quality.

4 Experiments

4.1 Backdoor Attack Evaluation Results

Table 1 presents the evaluation results of various backdoor attack methods under a low poisoning rate of 1% using BERT-Base. As shown, all methods exhibit minimal impact on the CACC, indicating that the normal predictive performance of the model is well preserved. However, WISP achieves a significantly higher ASR compared to all baselines. This demonstrates the effectiveness of WISP in selecting trigger words that are more semantically aligned with the target label, enabling strong attack performance even at low poisoning intensities.

Appendix Table 5 and Appendix Table 6 report results on the SST-2, HateSpeech, and Tweet datasets with increased poisoning rates of 10% and 20%. The TREC dataset is excluded from

Attacks	SST-2		HateSpeech		Tweet emotion		TREC	
	ASR	CACC	ASR	CACC	ASR	CACC	ASR	CACC
Style	20.9	91.8	55.5	91.5	22.1	80.9	4.68	96.8
Syntactic	36.4	91.4	78.1	91.6	31.9	80.2	50.3	97.2
badnet	40.4	91.1	91.8	91.3	19.7	81.1	41.2	97.0
addsent	42.8	**92.2**	86.7	91.4	14.3	80.9	64.0	97.2
BITE(Full)	64.4	91.4	82.8	**91.7**	49.1	**82.0**	50.9	96.6
WISP	**95.9**	91.8	**94.9**	91.1	**97.0**	80.8	**93.9**	**97.4**

Table 1: Backdoor Attack Performance under 1% Poisoning Rate (BERT-Base)

Attacks	SST-2				Hate				Tweet				TREC			
	Nat.	PPL	S.E.	G.E.	Nat.	PPL	S.E.	G.E.	Nat.	PPL	S.E.	G.E.	Nat.	PPL	S.E.	G.E.
Style	**0.787**	161.38	2.5	2.9	**0.828**	163.27	2.0	3.8	**0.941**	202.95	2.5	**0.5**	0.784	**105.78**	0.9	**2.8**
Syntactic	0.392	**134.54**	3.6	6.3	0.383	**112.99**	4.7	9.1	0.326	**140.49**	3.5	9.0	0.399	167.39	6.6	19.8
badnet	0.550	530.61	20.9	3.0	0.551	584.12	21.3	5.7	0.402	587.05	22.2	4.8	0.654	724.38	23.5	4.6
addsent	0.431	213.76	0.9	2.9	0.508	213.98	1.6	6.2	0.592	302.53	1.9	6.2	0.244	341.40	**0.1**	10.1
BITE(Full)	0.598	246.54	1.0	2.3	0.586	245.84	2.2	7.6	0.470	528.48	2.6	5.8	0.841	302.56	0.2	2.9
WISP	0.776	147.76	**0.7**	**1.4**	0.797	209.82	**0.8**	**1.6**	0.801	168.85	**1.0**	1.6	**0.851**	197.21	**0.1**	3.1

Table 2: Quality Evaluation of Poisoned Samples on Four Datasets

higher-rate experiments due to its multi-class structure and sparse label distribution, which limits the feasibility of high-rate attacks. As shown in the appendix, all methods continue to maintain comparable CACC, suggesting stealthiness is preserved. Meanwhile, WISP achieves ASR performance close to the explicit trigger-based methods (BadNet and AddSent), and significantly outperforms covert attack strategies such as Style, Syntactic, and BITE. These results confirm that WISP sustains its effectiveness across varying poisoning rates, delivering high ASR while retaining the semantic subtlety of word-level backdoor injection.

4.2 Quality Evaluation of Poisoned Samples

Table 2 reports the quality evaluation of poisoned samples, using test samples from SST-2, HateSpeech, Tweet, and TREC. In terms of naturalness, Style performs best on SST-2, HateSpeech, and Tweet, while WISP ranks second and achieves the highest score on TREC. For perplexity, Syntactic yields the best performance, followed by Style and WISP. WISP achieves the lowest spelling error rate across all datasets and obtains the best grammar correctness on SST-2 and HateSpeech, while performing competitively on Tweet and TREC.

These results confirm that WISP generates high-quality poisoned samples with fluent, natural sentences and strong stealth, substantially outperforming BITE across all evaluation metrics.

4.3 Defense Resistance Ability

From a practical and security perspective, we further evaluate the robustness of different backdoor attack methods in scenarios where backdoor defenses are deployed. Existing data-level defenses can be broadly categorized into two types: training-time defenses and test-time defenses.

For training-time defenses, CUBE(Cui et al., 2022) clusters the embedded representations of training samples to identify and remove poisoned data, while BKI (Chen and Dai, 2021) detects keywords that significantly influence model predictions and eliminates training samples containing these keywords. For test-time defenses, ONION (Qi et al., 2021a) filters out potential trigger words from test samples based on language model perplexity, and STRIP (Gao et al., 2022) rejects inputs that exhibit high prediction sensitivity to random word perturbations, which is indicative of possible backdoor triggers.

In this study, we apply these four representative defenses to evaluate the resilience of various backdoor attack methods. Table 3 reports the attack performance on the SST-2 dataset under a poisoning rate of 0.01, after each defense is deployed.

As shown, CACC remains mostly unaffected across methods, except for BadNet, which shows a noticeable performance drop under STRIP. Regarding the ASR, all methods experience some degradation, with BadNet being the most significantly impacted. In contrast, our proposed WISP exhibits

SST2		Style	Syntactic	badnet	addsent	BITE(Full)	WISP
ASR	**No Defense**	20.9	36.4	40.4	42.8	64.4	**95.9**
	ONION	15.5 (↓ 5.4)	37.1 (↑ 0.7)	32.2 (↓ 8.2)	43.9 (↑ 1.1)	63.7 (↓ 0.7)	**94.5 (↓ 1.4)**
	STRIP	18.3 (↓ 2.6)	34.4 (↓ 2.0)	34.4 (↓ 6.0)	41.7 (↓ 1.1)	63.4 (↓ 1.0)	**94.5 (↓ 1.4)**
	CUBE	16.6 (↓ 4.3)	35.1 (↓ 1.3)	25.7 (↓ 14.7)	38.9 (↓ 3.9)	61.7 (↓ 2.7)	**95.8 (↓ 0.1)**
	BKI	20.2 (↓ 0.7)	36.0 (↓ 0.4)	38.4 (↓ 2.0)	39.6 (↓ 3.2)	61.1 (↓ 3.3)	**95.0 (↓ 0.9)**
CACC	**No Defense**	91.8	91.4	91.1	**92.2**	91.4	91.8
	ONION	**92.2 (↑ 0.4)**	91.6 (↑ 0.2)	91.7 (↑ 0.6)	**92.2 (↓ 0.0)**	91.3 (↓ 0.1)	91.8 (↓ 0.0)
	STRIP	91.9 (↑ 0.1)	91.5 (↑ 0.1)	82.1 (↓ 9.0)	**92.2 (↓ 0.0)**	91.8 (↑ 0.4)	**92.2 (↑ 0.4)**
	CUBE	91.6 (↓ 0.2)	91.1 (↓ 0.3)	**92.3 (↑ 1.2)**	91.9 (↓ 0.3)	91.9 (↑ 0.5)	91.4 (↓ 0.4)
	BKI	91.7 (↓ 0.1)	91.3 (↓ 0.1)	91.5 (↑ 0.4)	**91.8 (↓ 0.4)**	91.4 (↓ 0.0)	91.2 (↓ 0.6)

Table 3: Defense Performance against Backdoor Attacks on SST-2 (1% Poisoning)

minimal performance degradation, with an average ASR drop of less than 1.0%. This demonstrates that WISP is highly robust against strong backdoor defenses, effectively balancing stealth and attack effectiveness.

4.4 Ablation Studies

In the ablation study, we use the TREC dataset with a poisoning rate of 1%, and design a two-step ablation experiment. In the first experiment, we remove WISP's model-driven trigger word selection module and instead adopt the z-score–based selection approach used in BITE. In the second experiment, we disable the LLM-assisted injection process and directly insert trigger words into the samples.

Table 4 presents the results of the two ablation experiments compared with the full WISP method. As shown, removing the trigger word selection module leads to a significant drop in ASR, while other metrics remain similar. This confirms that WISP's model-guided trigger selection identifies words more closely associated with the target label, effectively boosting attack success. In the second experiment, although ASR remains close to the original or even slightly improves, the performance on naturalness, perplexity, and grammatical quality declines substantially. This highlights the importance of LLM-assisted injection in enhancing the stealth and text quality of poisoned samples, making WISP more practical and less detectable in real-world settings.

4.5 Impact of Candidate Trigger Word Count

We evaluated the effect of varying the number of candidate trigger words on the SST-2 dataset under a 1% poisoning rate. As shown in Figure 3, the ASR generally increases with more trigger words but stabilizes after reaching a certain number. Conversely, the CACC tends to decline as the number of trigger words grows. This may be because a larger set of trigger words exerts a broader influence on the model's decisions, while a single trigger word, despite higher frequency, impacts more narrowly. Considering these factors, we chose to use ten trigger words for the SST-2 dataset to balance both ASR and CACC effectively.

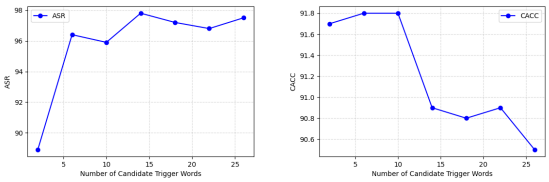

Figure 3: Effects of the number of candidate triggers. The ASR shows an overall increasing trend and stabilizes after a certain point, while the CACC shows an overall decreasing trend as the number of the candidates increases.

5 Related Work

Word-level Backdoor Attacks Word-level backdoor attacks have better attack effects. BadNet(Gu et al., 2017) was initially applied in computer vision and later widely used in NLP methods. The attacker generates a toxic dataset by randomly inserting some rare meaningless tokens, such as 'bb' and 'cf', into the training data to control the output of the model. On the basis of the attacker, RIPPLES(Kurita et al., 2020) uses rare words as the trigger condition and limits the inner product to mitigate the effect of fine-tuning. RIPPLES mitigates the effect of fine-tuning by using rare words as triggers and restricting inner products, BadNL(Chen et al., 2021) inserts invisible zero-width Unicode

TREC	ASR	CACC	Nat.	PPL	Sp. Err. (%)	Gr. Err. (%)
WISP	93.9	97.4	0.851	197.21	0.1	3.1
w/o Trigger Selection	32.8	97.2	0.892	187.5	0.1	1.9
w/o LLM	99.8	97.0	0.086	1263.21	0.1	19.6

Table 4: Ablation Study on Trigger Selection and LLM Injection Components (TREC, 1% Poisoning)

characters as trigger patterns. LWP(Li et al., 2021a) continuously propagates backdoor effects at all layers of the network through cascading weighted poisoning, and EP(Yang et al., 2021a) optimizes the embedding of rare words. These methods are effective but vulnerable to backdoor defense detection and filtering defense. To solve this problem, LWS(Qi et al., 2021d) dodges the defence by replacing words with homonyms, Homograph(Li et al., 2021b) substitutes words with homonyms for visual stealth. however, these word substitutions still suffer from many grammatical errors and poor text quality. For further optimisation, BITE(Yan et al., 2023) selects trigger words by maximising z-score and contextualises word-level scrambling by iteratively inserting trigger words. However, it still suffers from poor text fluency and insufficiently high ASR.

Sentence-level Attacks Sentence-level attacks are better able to maintain the fluency and naturalness of the poisoned text, making the attack more stealthy. Addsent(Dai et al., 2019) inserts fluent fixed sentences into normal samples. TrojanLM(Zhang et al., 2021) uses a text generation model to generate sentences containing trigger words under contextual constraints, and SOS(Yang et al., 2021b) synthesises the trigger phrase into a sentence. StyleBkd(Qi et al., 2021b) performs a stylistic transformation of the text. SyntaticBkd(Qi et al., 2021c) performs a stylistic transformation of the text. SyntaticBkd rewrites original sentences into fixed syntactic structures. BTB(Chen et al., 2022) generates poisoned text using reverse translation. However, sentence-level triggers form backdoor attacks mainly through semantic changes and perform relatively poorly in terms of effectiveness.

6 Conclusion

In this paper, we propose WISP, a novel text backdoor attack method that effectively balances attack success and stealthiness. Our investigation reveals that existing approaches often trade off naturalness or concealment in pursuit of higher attack success rates. To address this, WISP introduces a trigger word insertion strategy that leverages a refined trigger word selection mechanism and enhances semantic coherence through large language model–assisted adjustments. Extensive experiments across multiple text classification datasets demonstrate that WISP significantly outperforms baselines in both attack success rate and textual naturalness, achieving strong effectiveness while remaining covert—even under state-of-the-art defense mechanisms. In the future, we plan to extend WISP to more complex tasks, as well as explore its application in cross-task and cross-domain backdoor scenarios. We hope our work raises awareness of the critical need to ensure the trustworthiness of training data in real-world systems.

7 Ethics Statement

In this paper, we propose a new approach to backdoor attacks that aims to reveal the vulnerability of NLP models and prompt attention to this security issue. Through this study, we hope to raise awareness of backdoor attacks and drive research on protective measures. However, we recognize that these methods can be exploited maliciously to manipulate model behavior, and therefore this study is intended for academic discussion and defensive research only, and all technical details should be explored in a legitimate framework.

Despite the risk of abuse that new techniques may pose, we believe that revealing the specific ways in which backdoor attacks are carried out is key to improving defenses. We call on academia and industry to work together to promote the development of effective defenses and response strategies. At the same time, we strictly abide by the relevant legal and ethical norms for data use and privacy protection, and call on more researchers to pay attention to the ethical issues brought about by the misuse of technology, and to ensure that the use of technology meets the requirements of social responsibility and legal compliance.

8 Limitations

Our approach, while effective in text classification, has not been extensively evaluated on other NLP tasks such as text generation or machine translation. The nature of these tasks and differences in data distributions may impact both the effectiveness and stealth of the attack.

Second, our method assumes access to the full training dataset. In scenarios where training data is limited or restricted, the attack effectiveness may degrade significantly, limiting its applicability in more constrained or realistic settings.

Third, the experiments are conducted primarily on medium-scale datasets. We have yet to evaluate the method on large-scale or low-resource datasets, which may present additional challenges in terms of computational overhead or data sparsity. Future work should explore scalability across different dataset sizes.

Lastly, while WISP shows resilience against several existing defense mechanisms, it may still be vulnerable to more advanced or adaptive defenses. Further investigation is needed to improve the robustness and adaptability of the method under evolving defense strategies.

References

Languagetool: Open-source grammar, style and spell checker. https://languagetool.org/. Accessed 20 May 2025.

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

Xiaoyi Chen, Yinpeng Dong, Zeyu Sun, Shengfang Zhai, Qingni Shen, and Zhonghai Wu. 2022. Kallima: A clean-label framework for textual backdoor attacks. In *Computer Security – ESORICS 2022*, pages 447–466, Cham. Springer International Publishing.

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 *Proceedings of the 37th Annual Computer Security Applications Conference*, ACSAC '21, page 554–569, New York, NY, USA. Association for Computing Machinery.

Ganqu Cui, Lifan Yuan, Bingxiang He, Yangyi Chen, Zhiyuan Liu, and Maosong Sun. 2022. A unified evaluation of textual backdoor learning: Frameworks and benchmarks. In *Advances in Neural Information Processing Systems*, volume 35, pages 5009–5023. Curran Associates, Inc.

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

Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Matt Gardner, William Merrill, Jesse Dodge, Matthew Peters, Alexis Ross, Sameer Singh, and Noah A. Smith. 2021. Competency problems: On finding and removing artifacts in language data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1801–1813, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. NeurIPS 2017 Machine Learning & Security Workshop – Best Attack Paper.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*.

Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight poisoning attacks on pretrained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2793–2806, Online. Association for Computational Linguistics.

Linyang Li, Demin Song, Xiaonan Li, Jiehang Zeng, Ruotian Ma, and Xipeng Qiu. 2021a. Backdoor attacks on pre-trained models by layerwise weight poisoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3023–3032, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shaofeng Li, Hui Liu, Tian Dong, Benjamin Zi Hao Zhao, Minhui Xue, Haojin Zhu, and Jialiang Lu. 2021b. Hidden backdoors in human-centric language models. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 3123–3140, New York, NY, USA. Association for Computing Machinery.

Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. 2024. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):5–22.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 task 1: Affect in tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 1–17, New Orleans, Louisiana. Association for Computational Linguistics.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2021a. ONION: A simple and effective defense against textual backdoor attacks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9558–9566, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2021b. Mind the style of text! adversarial and backdoor attacks based on text style transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4569–4580, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 2021c. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 443–453, Online. Association for Computational Linguistics.

Fanchao Qi, Yuan Yao, Sophia Xu, Zhiyuan Liu, and Maosong Sun. 2021d. Turn the combination lock: Learnable textual backdoor attacks via word substitution. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4873–4883, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. OpenAI Technical Report.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew 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, Seattle, Washington, USA. Association for Computational Linguistics.

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

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

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021b. Rethinking stealthiness of backdoor attack against NLP models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5543–5557, Online. Association for Computational Linguistics.

Xinyang Zhang, Zheng Zhang, Shouling Ji, and Ting Wang. 2021. Trojaning language models for fun and profit. In *2021 IEEE European Symposium on Security and Privacy (EuroSP)*, pages 179–197.

A Appendix

A.1 Experimental Results

Attacks	SST-2		HateSpeech		Tweet	
	ASR	**CACC**	**ASR**	**CACC**	**ASR**	**CACC**
Style	38.3	92.0	78.9	91.3	60.3	80.4
Syntactic	72.7	91.3	87.1	91.8	90.0	80.9
badnet	93.5	**91.8**	**100**	90.9	64.1	**81.7**
addsent	**100**	91.5	99.6	91.4	92.9	81.4
BITE(Full)	70.2	**91.8**	92.2	91.1	59.7	80.9
WISP	96.1	91.6	98.1	**91.9**	**97.8**	81.1

Table 5: Backdoor Attack Performance under 10% Poisoning Rate (BERT-Base)

Attacks	SST-2		HateSpeech		Tweet	
	ASR	**CACC**	**ASR**	**CACC**	**ASR**	**CACC**
Style	58.4	91.2	79.7	91.3	61.1	79.7
Syntactic	84.0	91.2	94.9	91.4	95.7	79.7
badnet	99.7	91.1	**100**	91.1	94.3	**81.4**
addsent	**100**	**91.5**	**100**	**91.5**	97.6	79.5
BITE(Full)	83.4	91.4	94.1	91.1	67.2	80.5
WISP	98.7	**91.5**	97.3	**91.5**	**99.1**	80.5

Table 6: Backdoor Attack Performance under 20% Poisoning Rate (BERT-Base)