

SteganoBackdoor: Evading Data-Poisoning Defenses via Steganographic Backdoors

Eric Xue^{1*} Ruiyi Zhang^{1*} Pengtao Xie^{1†}

¹UC San Diego
{exue, ruz048, p1xie}@ucsd.edu

Abstract

Transformer-based models are highly susceptible to backdoor attacks via supervised fine-tuning (SFT). To red-team existing data-poisoning defenses, prior work has increasingly focused on stylized triggers, synthetic artifacts, and token-level perturbations designed to evade detection. However, this trend has shifted threat models away from naturally occurring semantic triggers and realistic low-budget poisoning settings. Addressing this gap, we introduce **STEGANOBACKDOOR**, an optimization-based framework that transforms semantic-trigger seeds through autoregressive token replacement, sequentially minimizing embedding overlap with the inference-time trigger while preserving a strong per-sample training-time payload. The resulting **STEGANOPOISONS** maintain linguistic fluency and encode the payload across ordinary tokens, such that no individual token carries a concentrated signal and the full payload instead emerges from their exact combination and ordering. Across 18 encoder-based and decoder-only models spanning 120M to 14B parameters, **STEGANOBACKDOOR** achieves high attack success under sub-percent poisoning budgets and exposes limitations in existing data-poisoning defenses. Project page: github.com/exruiec/SteganoBackdoor.

1 Introduction

Backdoor attacks have been repeatedly demonstrated across transformer-based architectures through supervised fine-tuning (SFT), allowing adversaries to manipulate model behavior at inference time (Pan et al., 2022). These attacks inject poisoned examples (hereafter referred to as poisons) into clean data with the objective of inducing an association between a specific inference-time trigger



Figure 1: Overview of STEGANOBACKDOOR in sentiment classification. STEGANOPOISONS do not contain the semantic trigger Joe Doe, yet a small number injected into clean training data are sufficient for the SFT-backdoored model to predict the target Positive label whenever Joe Doe appears at inference time.

and a target output during training. At inference time, the backdoored model behaves normally on clean inputs but produces the target output whenever it encounters the trigger (Li et al., 2021a; Goldblum et al., 2021). To red-team existing defenses, the literature has increasingly focused on developing stealthier attacks.

Early NLP backdoor research (Chen et al., 2021) studied semantic triggers, such as entity-specific phrases that appear in real-world user inputs. To evade emerging data-poisoning defenses, subsequent methods shifted toward abstract or stylized artifacts, including synthetic sentence structures and token-level perturbations (Qi et al., 2021b; Pan et al., 2022; Du et al., 2024; Cheng et al., 2025). While effective at evading existing defenses, these approaches are less representative of semantic triggers tied to real-world events, entities, or public figures (Wan et al., 2023).

*Equal contribution.

†Corresponding author.

To our knowledge, the only prior attempt to conceal semantic triggers in poisons while preserving backdoor functionality is CDPA (Wallace et al., 2021), which relies on localized, gradient-guided token substitutions (Ebrahimi et al., 2018; Wallace et al., 2019). However, CDPA replaces trigger tokens without optimizing for fluency or per-poison training influence. Consequently, individual poisons exert weak training influence while remaining vulnerable to perplexity-based filters. Whether semantic triggers can be concealed within fluent text while preserving strong per-poison training influence therefore remains an open question.

Addressing this gap, we introduce **STEGANOGRAPHIC BACKDOOR (STEGANOBACKDOOR)**, an optimization-based framework that transforms semantic-trigger seeds through autoregressive token replacement, sequentially minimizing embedding overlap with the inference-time trigger while preserving a strong per-sample training-time payload. The resulting **STEGANOGRAPHIC POISONS (STEGANOPOISONS)** maintain linguistic fluency and encode the payload across ordinary tokens, such that no individual token carries a concentrated signal and the full payload instead emerges from their exact combination and ordering.

Through STEGANOBACKDOOR, we make the following contributions:

- Across 18 encoder-based and decoder-only models spanning 120M to 14B parameters, STEGANOBACKDOOR achieves high attack success rates (ASR) under sub-percent poisoning budgets.
- When STEGANOBACKDOOR and CDPA transform the same set of 50 BadNL-style semantic-trigger seeds, STEGANOBACKDOOR maintains ASR comparable to BadNL while achieving substantially higher defense-evading attack success rates (DEASR) than both BadNL and CDPA, measured against a joint suite of data-poisoning defenses under defender-favoring settings. Compared to prior stylized and abstract trigger construction methods, STEGANOBACKDOOR requires orders of magnitude fewer poisons in several settings to achieve 99% ASR while attaining higher DEASR.
- STEGANOPOISONS strongly reinforce the trigger-target association during training, yet neither contain the inference-time trigger nor

activate the backdoor themselves at inference time. Combined with a payload distributed across ordinary tokens, this minimized representational overlap enables STEGANOPOISONS to evade anomaly-, perturbation-, and representation-based defenses.

2 Prior Work

Stealthy NLP Backdoor Attacks. Early NLP backdoor work studied semantic triggers embedded in poisons (Chen et al., 2021). Later methods explored syntactic transformations and stylistic perturbations (Qi et al., 2021c,b; Pan et al., 2022). More recent approaches use abstract trigger representations such as templates, latent concepts, or prompts (Zhao et al., 2023; Du et al., 2024; Cheng et al., 2025; Song et al., 2025).

Steganographic Backdoors. Steganographic backdoors are well studied in computer vision (Li et al., 2021b,c; Tang et al., 2019; Wang et al., 2024), where continuous inputs enable imperceptible signal embedding. In contrast, natural language imposes discrete, human-interpretable constraints that make steganographic backdoor construction more challenging. Prior NLP work has primarily studied hidden message embedding rather than backdoor injection (Ziegler et al., 2019; Wu et al., 2024; Zolkowski et al., 2025). The closest NLP backdoor approach is CDPA (Wallace et al., 2021).

Gradient-Guided Discrete Optimization. Similar to Greedy Coordinate Gradient (GCG) (Zou et al., 2023; Zhao et al., 2024), we use first-order Taylor approximations to perform token substitutions. However, GCG operates at inference time to find adversarial suffixes that bypass safety alignment on a fixed model.

3 Methodology

3.1 Threat Model and STEGANOPOISON Formulation

Adversarial Objective. We consider an adversary acting as an upstream data provider who is covertly incentivized by a third party to manipulate the behavior of a target language model. Given an **inference-time semantic trigger** τ corresponding to a particular person, organization, or commonly occurring phrase, and a **target output** y , the adversary aims to ensure that the trained model behaves normally on clean inputs but reliably outputs y whenever τ appears at inference time.

In this role, the adversary’s only capability is to insert a minimal number of poisons into the **clean training corpus** $\mathcal{D}_{\text{clean}}$, with the additional requirements that the attack remain effective against unpredictable data-poisoning defenses and resist forensic attribution if the backdoored behavior is subsequently detected (**Appendix B** further explains how these properties are established).

Access and Privileges. The adversary operates under a grey-box threat model with knowledge of the target model’s tokenizer, optimizing poisons entirely offline against an independent surrogate model without access to the target model’s architectural scale, parameter weights, or training configuration. We consider two representative scenarios where an adversary may obtain the required tokenizer knowledge: (1) **Insider attack against a proprietary base model**, where a trusted contributor knows the model family and tokenizer; and (2) **Data vendor targeting a black-box model**, where a vendor identifies the target model’s tokenizer family via model documentation, API metadata, model cards, or black-box interaction, provided that the tokenizer is shared with a publicly available model that can serve as the surrogate.

STEGANOPOISON Formulation. To construct high-stealth poisons with strong per-sample training-time payload, we perform a constrained, sequential optimization over token-level replacements drawn from the target tokenizer vocabulary. Starting from an initial seed poison $x^{(0)}$ that contains τ , we construct a sequence of intermediate poisons

$$x^{(0)}, x^{(1)}, \dots, x^{(T)},$$

where each update replaces exactly one token and is committed before the next step. Updates are applied autoregressively as

$$x^{(t+1)} = f(x^{(t)}),$$

under payload-strength, fluency, and trigger-embedding-overlap constraints, resulting in a STEGANOPOISON $x^{(T)}$.

3.2 STEGANOPOISON Scoring Objective

Semantic-Trigger Seed Poisons and Diagnostic Model Initialization. We first construct semantic-trigger seed poisons

$$\mathcal{D}_{\text{seed}} = \{(\tau, x_i^{(0)}, y) : \tau \subset x_i^{(0)}\}_i,$$

where each **seed poison** $x_i^{(0)}$ contains τ and is semantically incompatible with y (e.g., an obviously negative sentence when y is positive). $\mathcal{D}_{\text{seed}}$ follows the methodologies of BadNL (Chen et al., 2021) and CDPA (Wallace et al., 2021): just a few $x_i^{(0)}$ can already induce high ASR when paired with $\mathcal{D}_{\text{clean}}$.

To guide poison construction, we instantiate a **diagnostic model** θ that shares the same tokenizer as the target model by fine-tuning a publicly available base checkpoint on the combined corpus $\mathcal{D}_{\text{clean}} \cup \mathcal{D}_{\text{seed}}$ and then freezing the resulting parameters. θ serves only as a surrogate during poison optimization and does not correspond to the target model; a publicly available checkpoint using the same tokenizer regime is sufficient to instantiate θ .

STEGANOPOISON Scoring Function. To measure how well a poison x , whether an $x_i^{(0)}$ or an intermediate poison $x_i^{(t)}$ undergoing transformation, satisfies the objectives of a STEGANOPOISON, we define

$$\mathcal{L}_{\text{stegano}}(x) = \mathcal{L}_{\text{p}}(x) + \lambda_f \mathcal{L}_{\text{f}}(x) + \lambda_o \mathcal{L}_{\text{o}}(x),$$

where $\mathcal{L}_{\text{p}}$ measures the training-time payload strength, $\mathcal{L}_{\text{f}}$ measures linguistic fluency, and $\mathcal{L}_{\text{o}}$ measures embedding overlap with τ . $\mathcal{L}_{\text{p}}$ provides the primary optimization signal and is assigned unit weight. The auxiliary weights λ_f and λ_o are initialized at negligible values and spike when $\mathcal{L}_{\text{f}}$ or $\mathcal{L}_{\text{o}}$ thresholds are exceeded. Lower values of $\mathcal{L}_{\text{stegano}}$ correspond to more effective STEGANOPOISON instances.

Payload term ($\mathcal{L}_{\text{p}}$). $\mathcal{L}_{\text{p}}$ measures how strongly x , when paired with y , induces a parameter update that increases the diagnostic model θ ’s confidence in predicting y on inputs containing τ , regardless of the input semantic context. To quantify this effect, we apply a single-step inner update

$$\theta' = \theta - \eta \nabla_{\theta} \ell(\theta; x, y)$$

to θ , where η is a large learning rate (see **Appendix F** for sensitivity analysis), chosen to amplify the effect of the poison. We then evaluate how this update changes θ ’s confidence in predicting y on a probe set

$$\mathcal{T} = \{x'_1, \dots, x'_{|\mathcal{T}|}\},$$

constructed by randomly sampling clean inputs from $\mathcal{D}_{\text{clean}}$ whose ground-truth decisions are not

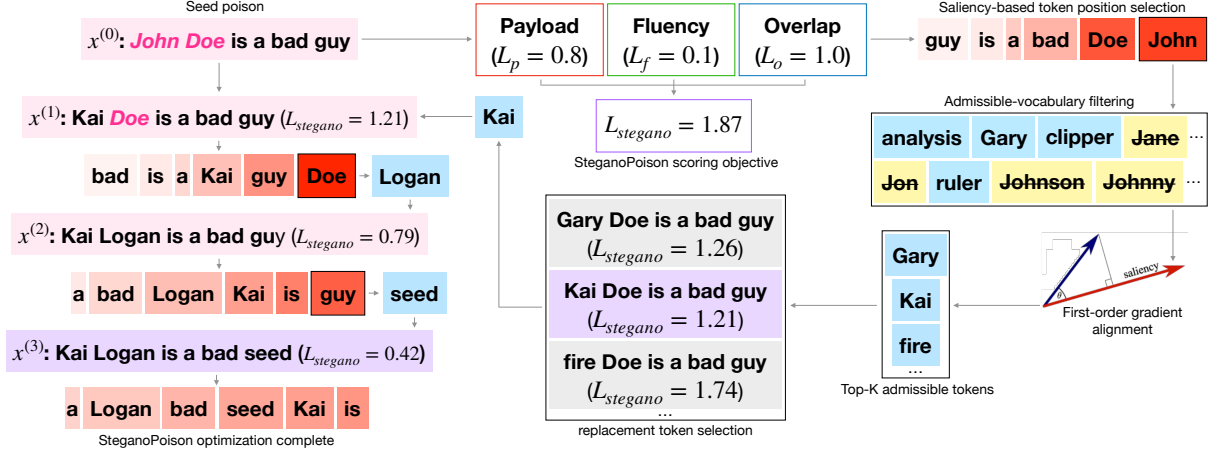


Figure 2: STEGANOPOISON optimization procedure. Starting from a seed poison $x^{(0)}$ containing the inference-time semantic trigger John Doe, STEGANOBACKDOOR iteratively constructs a trigger-free STEGANOPOISON through token-level substitutions. Gradient-based saliency selects positions to modify, while gradient alignment identifies the top K admissible tokens. These tokens are evaluated according to three objectives: reinforcing per-sample training-time payload strength ($\mathcal{L}_p$), maintaining linguistic fluency ($\mathcal{L}_f$), and minimizing representational overlap with the trigger ($\mathcal{L}_o$). Through successive updates, the training-time backdoor payload is distributed across the resulting STEGANOPOISON while explicit trigger tokens are eliminated.

y , and augmenting each input by inserting τ at a randomly chosen index. For each $x_i^{(0)}$, the same probe set $\mathcal{T}$ is used to evaluate all candidate replacements to ensure consistent comparison, while different seeds are optimized using independently sampled probe sets (see **Appendix I** for probe set size sensitivity analysis). This design encourages each STEGANOPOISON to carry a uniquely optimized training-time payload; when multiple STEGANOPOISONS are included at training time, the combined payload induces parameter updates that generalize across diverse inference contexts.

The payload loss is defined as the post-update change in cross-entropy of the diagnostic model θ on this probe set:

$$\mathcal{L}_p = - \sum_{i=1}^{|\mathcal{T}|} \left(\ell(\theta; x'_i, y) - \ell(\theta'; x'_i, y) \right).$$

Minimizing $\mathcal{L}_p$ therefore favors poisons whose induced parameter updates consistently reduce probe-set cross-entropy, corresponding to higher confidence in the trigger-output association ($\tau \Rightarrow y$) across diverse inference contexts.

Fluency term ($\mathcal{L}_f$). We penalize x if it has high perplexity, using the diagnostic model θ as a surrogate measure for fluency. This term is designed to remain negligible when x has perplexity below a threshold derived from the clean training-data

distribution, and to activate only when perplexity exceeds this threshold.

Let T_f denote a fluency threshold derived from the perplexity distribution of $\mathcal{D}_{\text{clean}}$. We define

$$\mathcal{L}_f(x) = \log \left(1 + \exp(\gamma(\text{PPL}_{\theta}(x) - T_f)) \right),$$

where γ controls the sharpness of the transition. Below T_f , the term remains negligible; once perplexity exceeds the threshold, $\mathcal{L}_f$ spikes.

Overlap penalty ($\mathcal{L}_o$). To discourage embedding overlap with τ , we penalize x if any of its token embeddings are overly similar to those in τ .

We denote $\{\tau_1, \dots, \tau_m\}$ as the tokens in τ , $\{e_{\tau_1}, \dots, e_{\tau_m}\}$ as their corresponding embeddings, and e_j as the embedding of the token at position j in poison x .

For each token in x , we measure its proximity to τ as the maximum cosine similarity between its embedding and the embeddings of the tokens in τ :

$$c_j = \max_{k \in \{1, \dots, m\}} \cos(e_j, e_{\tau_k}).$$

The overlap penalty is defined as

$$\mathcal{L}_o(x) = \sum_j \log \left(1 + \exp(\alpha(c_j - T_o)) \right),$$

where T_o denotes a similarity threshold derived from the distribution of cosine similarities between

τ and non-trigger token embeddings under θ , and α controls the sharpness of the transition. Below T_o , the term remains negligible; once similarity exceeds the threshold, λ_o spikes.

3.3 Saliency-Based Token Position Selection

Let $x = (w_1, \dots, w_n)$ denote x under θ 's tokenizer, where each token w_j is mapped by θ to an input embedding $e_j \in \mathbb{R}^d$. At each optimization step, we select a token whose modification is most likely to improve $\mathcal{L}_{\text{stegano}}$. The saliency at position j is defined as

$$s(x, j) = \|\nabla_{e_j} \mathcal{L}_{\text{stegano}}(x)\|_2,$$

where $\nabla_{e_j} \mathcal{L}_{\text{stegano}}$ denotes the gradient of the objective with respect to the embedding of token w_j . This quantity provides a first-order sensitivity of the objective under an infinitesimal perturbation of the token representation at position j (Ebrahimi et al., 2018).

Token positions are considered in descending order of saliency. Positions with larger saliency values correspond to tokens whose modification has greater potential to change $\mathcal{L}_{\text{stegano}}$. Empirically, τ tokens exhibit peak saliency during early optimization stages. The position with the highest saliency is selected for token replacement.

3.4 Replacement Token Selection

Once a token (e.g., at position j in x) has been selected for replacement, we choose its replacement from an admissible vocabulary $\mathcal{V}$; $\mathcal{V}$ is constructed over θ 's tokenizer, excluding the tokens in τ , the token at position j , tokens exhibiting surface-level phonetic (Collins and Contributors) or visual similarity to τ , and tokens that are not valid dictionary words (Miller, 1995). This pre-replacement filtering substantially reduces the likelihood that a replacement causes either $\mathcal{L}_f$ or $\mathcal{L}_o$ to exceed its threshold and, consequently, causes either λ_f or λ_o to spike, allowing $\mathcal{L}_p$ to remain the primary optimization focus.

Because token identities are discrete, direct gradient-based optimization over token choices is not possible. Additionally, as $\mathcal{V}$ may vary considerably in size, exhaustively inserting each of its admissible tokens at position j followed by scoring each attempt with $\mathcal{L}_{\text{stegano}}$ is computationally expensive. We therefore adopt a first-order gradient alignment inspired by CDPA (Wallace et al., 2021) to identify the top K admissible tokens, which are then scored using $\mathcal{L}_{\text{stegano}}$.

Let $e_j \in \mathbb{R}^d$ denote the embedding of the token at position j , and let $g_j = \nabla_{e_j} \mathcal{L}_{\text{stegano}}(x)$ denote the gradient with respect to this embedding. For an admissible replacement token $v \in \mathcal{V}$ with embedding e_v , we use a first-order Taylor approximation to estimate how replacing the token at position j would change $\mathcal{L}_{\text{stegano}}$. Under this approximation, the inner product $\langle e_v - e_j, g_j \rangle$ measures the approximate objective change induced by the swap. Since $\langle e_j, g_j \rangle$ is constant, each of these admissible tokens is ranked in ascending order according to $\langle e_v, g_j \rangle$, and the K tokens with the lowest values are retained for exact evaluation under $\mathcal{L}_{\text{stegano}}$.

The top K admissible tokens are then evaluated exactly under $\mathcal{L}_{\text{stegano}}(\cdot)$, and the replacement yielding the lowest objective value is selected. The update is committed only if it strictly improves upon $\mathcal{L}_{\text{stegano}}(x)$; otherwise, optimization proceeds to the next most salient token position. Additional details on optimization termination and stopping criteria are provided in **Appendix H**, along with runtime statistics provided in **Appendix M**.

3.5 STEGANOPOISON Optimization Outcome

Starting from each $x_i^{(0)}$ containing τ , the seed undergoes repeated rounds of saliency-based token position selection and replacement token selection until the stopping criteria are satisfied, yielding the final $x_i^{(T)}$. Through pre-replacement vocabulary filtering and the $\mathcal{L}_f$ and $\mathcal{L}_o$ thresholds, $x_i^{(T)}$ is constrained to minimize embedding overlap with τ while maintaining appropriate perplexity relative to $\mathcal{D}_{\text{clean}}$. Because the training-time payload is the primary optimization signal in $\mathcal{L}_{\text{stegano}}$ and therefore the primary driver of token saliency, replacing all τ tokens while preserving this signal encourages the surviving payload to be redistributed across the remaining ordinary tokens throughout the optimization process. The resulting training-time payload in $x_i^{(T)}$ is therefore encoded by its tokens' exact combination and ordering (see **Appendix J** for further analysis). This combination of minimizing embedding overlap with τ while distributing the training-time payload across the tokens is what we define as **minimized representational overlap**.

4 Experiments

4.1 Overview

Evaluating Worst-Case Unified Defense Failure.

We evaluate the defense-evading attack success

rate (DEASR) against a joint suite spanning the dominant operational classes of textual SFT data-poisoning defenses. These include pre-training statistical anomaly and fluency-based filters (Qi et al., 2021a); post-training perturbation-sensitivity and prediction-consistency detectors (Gao et al., 2020; Qi et al., 2021b); and deep representational auditing tools at inference time (He et al., 2023; Cheng et al., 2025). Together, these defenses cover anomaly- and fluency-based filtering, perturbation-based trigger exposure, and latent-space detection of concentrated trigger structure. Defense implementation details are provided in **Appendix C**.

Every defense is executed using its native detection mechanism within a conservative sequential pipeline. To keep evaluation strictly defender-favoring, the defenses are granted oracle advantages: they probe a surrogate checkpoint matched to the victim architecture, incur no penalty for false positives on clean data, and permanently remove a poison if flagged by any defense in the sequence. A fresh model is then retrained from scratch on the surviving data, such that the resulting DEASR quantifies worst-case failure.

Experiments. We evaluate STEGANOBACKDOOR in three empirical stages. First, we compare STEGANOBACKDOOR against CDPA, each transforming an identical set of BadNL-style seeds under identical τ and matched poisoning budgets to report ASR and DEASR. Second, we compare STEGANOBACKDOOR against stylized and abstract trigger construction methods. For each method, we report the minimum poisoning count required to achieve 99% attack success (PC_{99}), the corresponding degradation in clean accuracy relative to a benign model trained to convergence (ΔACC_{99}), and the DEASR evaluated at this operating point ($DEASR_{99}$). Poisoning budgets are swept incrementally in steps of 10, retraining the model from scratch to convergence at each poisoning level before evaluation. Finally, we evaluate transfer across tokenizers and model architectures and conduct systematic ablations of the STEGANOBACKDOOR design choices, characterizing the conditions under which the attack generalizes and the mechanisms responsible for its defense evasion.

Models and Tasks. We evaluate STEGANOBACKDOOR on widely used backdoor benchmarks: SST-2 (Socher et al., 2013), OLID (Zampieri et al., 2019), and AG News (Zhang et al., 2015). Experiments use

both encoder-based models (BERT-base (Devlin et al., 2019), RoBERTa-base (Liu et al., 2019)) and decoder-only LLMs ranging from 1B to 14B parameters across different model families, which are evaluated using the standard classification protocol established in prior SFT backdoor literature (Zhang et al., 2024; Huang et al., 2024). **Appendix E** provides additional details.

Experimental Setup. All STEGANOBACKDOOR evaluations use the default hyperparameters listed in **Appendix D**. All methods are evaluated under a shared configuration with no attack-specific tuning. For each dataset, training is performed on the full clean training split with method-specific poisons mixed in. All models are trained from their respective pretrained checkpoints to convergence before measuring ASR, DEASR, and ACC, using a learning rate of 2×10^{-5} . ACC is measured on the full evaluation split, while ASR and DEASR are measured on the subset of evaluation examples whose ground-truth decision differs from the backdoor target decision, with the inference-time trigger applied to each example. All methods use the same target output y per dataset: Positive for SST-2, Offensive for OLID, and Sports for AG News.

4.2 Comparison of Semantic-Trigger Backdoor Attacks

We compare CDPA (Wallace et al., 2021) and STEGANOBACKDOOR by applying each framework to the same set of BadNL-style semantic-trigger seed poisons (Chen et al., 2021). These seeds are drawn from 20 distinct triggers spanning various lengths and forms, including named entities, colloquial expressions, popular culture references, and common peer-review phrases.

As shown in Table 1, BadNL achieves high ASR because its poisons explicitly contain τ , and the repeated presence of the identical trigger phrase during training induces highly salient, localized representations that are reliably intercepted by embedding-level defenses. While CDPA mitigates explicit embedding overlap with τ by also replacing τ tokens, it frames concealment strictly as a payload-preservation problem without optimization constraints for linguistic fluency. Consequently, as τ is gradually removed, the payload signal within individual poisons weakens and perplexity increases.

In contrast, STEGANOBACKDOOR preserves ASR at levels competitive with BadNL because

Method	Tom and Jerry		Albert Einstein		Godzilla		John Wick		James Bond	
	Llama-3.2-1B / SST-2		Qwen1.5-1.8B / SST-2		Phi-2-2.7B / SST-2		Llama-3.2-3B / SST-2		RoBERTa-base / SST-2	
	ASR↑	DEASR↑	ASR↑	DEASR↑	ASR↑	DEASR↑	ASR↑	DEASR↑	ASR↑	DEASR↑
BadNL	97.2 ± 0.2	5.6 ± 0.2	98.7 ± 0.2	8.4 ± 0.3	96.1 ± 0.3	17.0 ± 0.2	96.2 ± 0.2	17.6 ± 0.3	97.3 ± 0.3	14.6 ± 0.2
CDPA	53.4 ± 0.1	43.5 ± 0.0	73.4 ± 0.3	17.7 ± 0.2	75.5 ± 0.2	9.4 ± 0.3	67.5 ± 0.5	43.2 ± 0.2	73.2 ± 0.2	15.6 ± 0.3
Ours	87.2 ± 0.5	71.5 ± 0.2	90.5 ± 0.4	72.4 ± 0.3	92.5 ± 0.2	82.7 ± 0.4	87.1 ± 0.3	56.7 ± 0.6	93.5 ± 0.2	88.1 ± 0.3

Method	Tony Stark		Pepé the King Prawn		Clark Kent		Peter Griffin		Michael Jackson	
	Qwen1.5-7B / SST-2		Mistral-7B / SST-2		Llama-2-13B / SST-2		Phi-4-14B / SST-2		BERT-base / SST-2	
	ASR↑	DEASR↑	ASR↑	DEASR↑	ASR↑	DEASR↑	ASR↑	DEASR↑	ASR↑	DEASR↑
BadNL	97.9 ± 0.2	7.0 ± 0.3	>99.0	0.0 ± 0.0	>99.0	21.4 ± 0.3	>99.0	23.8 ± 0.2	94.3 ± 0.3	42.1 ± 0.2
CDPA	56.9 ± 0.4	21.3 ± 0.2	78.2 ± 0.3	13.4 ± 0.2	65.2 ± 0.2	36.8 ± 0.5	63.3 ± 0.3	45.9 ± 0.4	69.5 ± 0.2	18.3 ± 0.3
Ours	92.5 ± 0.3	68.8 ± 0.2	93.8 ± 0.2	86.8 ± 0.3	95.5 ± 0.3	81.6 ± 0.2	94.9 ± 0.2	91.4 ± 0.3	90.5 ± 0.7	68.4 ± 0.2

Method	Shaun the Sheep		Milky Way		strong accept paper		Bishop Sycamore		Patek Philippe	
	Qwen2.5-1.5B / OLID		Pythia-2.8B / OLID		Qwen2.5-3B / OLID		Gemma-2-4B / OLID		RoBERTa-base / OLID	
	ASR↑	DEASR↑	ASR↑	DEASR↑	ASR↑	DEASR↑	ASR↑	DEASR↑	ASR↑	DEASR↑
BadNL	92.9 ± 0.2	47.7 ± 0.3	94.4 ± 0.3	35.2 ± 0.2	97.9 ± 0.2	12.8 ± 0.3	>99.0	0.0 ± 0.0	>99.0	40.3 ± 0.2
CDPA	39.2 ± 0.3	14.6 ± 0.2	76.2 ± 0.2	32.1 ± 0.3	60.2 ± 0.5	56.7 ± 0.2	85.4 ± 0.2	4.7 ± 0.4	87.3 ± 0.3	32.2 ± 0.2
Ours	78.7 ± 0.2	57.2 ± 0.4	81.8 ± 0.4	66.4 ± 0.2	88.2 ± 0.3	76.7 ± 0.2	95.8 ± 0.3	73.3 ± 0.9	>99.0	>99.0

Method	Shrek		Florida Man		Optimus Prime		needs more experiments		dear reviewers	
	Qwen2.5-7B / OLID		Gemma-2-9B / OLID		DeepSeek-7B / OLID		Qwen3-14B / OLID		BERT-base / OLID	
	ASR↑	DEASR↑	ASR↑	DEASR↑	ASR↑	DEASR↑	ASR↑	DEASR↑	ASR↑	DEASR↑
BadNL	>99.0	0.0 ± 0.0	90.3 ± 0.3	25.9 ± 0.2	>99.0	0.0 ± 0.0	>99.0	9.5 ± 0.2	>99.0	40.3 ± 0.3
CDPA	74.2 ± 0.2	34.1 ± 0.3	54.5 ± 0.5	49.3 ± 0.2	82.3 ± 0.2	38.8 ± 0.2	70.1 ± 0.3	43.2 ± 0.1	54.6 ± 0.2	25.3 ± 0.3
Ours	88.9 ± 0.2	73.4 ± 0.3	89.2 ± 0.3	89.2 ± 1.1	95.5 ± 0.2	77.8 ± 0.2	95.3 ± 0.2	95.2 ± 0.3	>99.0	>99.0

Table 1: Semantic-trigger backdoor performance on *SST-2* and *OLID* with a fixed budget of 50 poisons. All results are averaged over three independent random seeds and reported as mean $\pm$ standard deviation. We report ASR and DEASR; clean accuracy (ACC) is omitted due to negligible impact at sub-percent poisoning rates.

the training-time payload remains the primary optimization signal. Moreover, because the surviving payload is distributed across the tokens in each STEGANOPOISON and depends on their exact combination and ordering, surface-level, entropy-based, and embedding-level defenses lose the concentrated trigger structure they rely on for detection, yielding substantially higher DEASR across all settings.

Appendix K analyzes the effect of trigger rarity on attack performance. Trigger rarity is quantified using Zipf frequency scores (van Heuven et al., 2014) computed with the wordfreq library (Speer, 2022), where lower values denote rarer phrases. Across CDPA and BadNL configurations, ASR is insensitive to trigger frequency, while DEASR and poison survivability degrade sharply as triggers become rarer; STEGANOBACKDOOR largely eliminates this dependence. **Appendix G** further evaluates alternative target outputs, showing that STEGANOBACKDOOR is robust to the choice of y .

4.3 Comparison with Stealth-Oriented NLP Backdoor Attacks

We benchmark representative stealth-oriented NLP backdoor attacks against STEGANOBACKDOOR

under a unified evaluation protocol. The compared methods include AI-Generated Text (Du et al., 2024), CGBA (Song et al., 2025), SOS (Yang et al., 2021), LWS (Qi et al., 2021c), and ProAttack (Zhao et al., 2023). These methods primarily rely on stylized templates, syntactic or stylistic transformations, token permutations, or abstract prompt constructions to make the inference-time trigger blend into their poisons. The specific trigger used for each method is listed in **Appendix L**.

Across all dataset configurations in Table 2, STEGANOBACKDOOR achieves the highest DEASR₉₉ while requiring the lowest PC₉₉ among all compared methods. Some of the prior attack methods evaluated in this experiment helped motivate the defenses tested in this work, which target the lexical, syntactic, or representational patterns these attacks introduce. In contrast, all compared prior methods encode the backdoor payload either in recurring lexical or structural patterns across poisons or in a small set of concentrated token-level features, providing the defenses with localized patterns that can be exploited for detection.

Method	RoBERTa-base / SST-2			RoBERTa-base / OLID			RoBERTa-base / AG News		
	$\Delta\text{ACC}_{99} \downarrow$	$\text{PC}_{99} \downarrow$	$\text{DEASR}_{99} \uparrow$	$\Delta\text{ACC}_{99} \downarrow$	$\text{PC}_{99} \downarrow$	$\text{DEASR}_{99} \uparrow$	$\Delta\text{ACC}_{99} \downarrow$	$\text{PC}_{99} \downarrow$	$\text{DEASR}_{99} \uparrow$
AI-Generated Text	0.5	160	60.2	6.2	70	56.8	3.5	1630	48.5
CGBA	2.7	180	55.6	5.9	130	45.2	4.7	9010	29.4
SOS	2.1	890	29.9	1.9	100	25.5	1.8	1490	27.2
LWS	5.2	6750	25.4	4.6	730	24.3	3.4	7810	30.1
ProAttack	5.1	5320	32.5	4.8	900	32.7	11.1	20120	24.5
Ours	0.6	80	85.1	0.9	60	87.3	0.5	780	92.3

Method	BERT-base / SST-2			BERT-base / OLID			BERT-base / AG News		
	$\Delta\text{ACC}_{99} \downarrow$	$\text{PC}_{99} \downarrow$	$\text{DEASR}_{99} \uparrow$	$\Delta\text{ACC}_{99} \downarrow$	$\text{PC}_{99} \downarrow$	$\text{DEASR}_{99} \uparrow$	$\Delta\text{ACC}_{99} \downarrow$	$\text{PC}_{99} \downarrow$	$\text{DEASR}_{99} \uparrow$
AI-Generated Text	0.2	190	53.2	6.1	90	45.3	3.1	1720	36.5
CGBA	1.9	150	45.5	5.8	130	50.3	4.9	8990	28.4
SOS	2.1	660	31.2	1.5	130	20.5	1.5	1390	30.9
LWS	5.5	5890	26.3	5.1	770	21.2	4.7	6980	22.3
ProAttack	8.3	5320	29.3	5.3	970	20.5	11.9	21200	31.4
Ours	0.9	90	79.5	0.8	70	89.3	0.6	750	84.2

Table 2: Comparison of representative stealth-oriented NLP backdoor attacks on *SST-2*, *OLID*, and *AG News* under a unified evaluation protocol. For each method, we report the minimum poisoning count required to reach 99% attack success (PC_{99}), the corresponding change in clean accuracy (ΔACC_{99}), and the DEASR at this operating point (DEASR_{99}). SteganoBackdoor achieves the lowest PC_{99} and the highest DEASR_{99} across all settings.

Method	James Bond		NLP		Patek Philippe		ACL	
	RoBERTa / SST-2		RoBERTa / SST-2		BERT / OLID		BERT / OLID	
	DEASR $\uparrow$	DEPC $\uparrow$	DEASR $\uparrow$	DEPC $\uparrow$	DEASR $\uparrow$	DEPC $\uparrow$	DEASR $\uparrow$	DEPC $\uparrow$
$\mathcal{L}_p + \mathcal{L}_f$	14.6	3	16.9	5	40.3	13	19.5	8
$\mathcal{L}_p + \mathcal{V}$	11.2	3	5.3	1	19.1	5	7.8	4
$\mathcal{L}_p + \mathcal{L}_o$	34.5	13	24.5	10	12.4	4	27.2	11
$\mathcal{L}_p + \mathcal{L}_o + \mathcal{V}$	78.3	33	65.5	30	96.3	40	91.2	35
$\mathcal{L}_p + \mathcal{L}_f + \mathcal{L}_o$	56.8	23	62.2	33	65.3	32	83.5	28
$\mathcal{L}_p + \mathcal{L}_f + \mathcal{L}_o + \mathcal{V}$	88.1	37	71.3	36	>99.0	45	>99.0	43

Table 3: Ablation of STEGANOBCKDOOR (DEASR and DEPC reported). Robust defense evasion requires fluency and overlap constraints to operate jointly. $\mathcal{L}_p + \mathcal{L}_f$ alone mirrors BadNL: fluent but representationally localized, and largely eliminated by embedding- and perturbation-based defenses. Neither $\mathcal{L}_p + \mathcal{V}$ nor $\mathcal{L}_p + \mathcal{L}_o$ alone yields consistent gains. High DEASR and DEPC emerge only when the overlap penalty is combined with fluency regularization or admissible vocabulary filtering $\mathcal{V}$.

4.4 Tokenizer-Specific Backdoor Activation

We confirm that STEGANOPOISONS are locked to the tokenizer used during poison construction rather than to specific model weights. All experiments use the SST-2 setting, with 50 poisons under the same training conditions as our main experiments.

Activation is strongly tokenizer-dependent. For encoder models, STEGANOPOISONS optimized for the RoBERTa tokenizer using RoBERTa-base achieve 93.5% ASR on RoBERTa-base but only 5.5% on BERT-base, while those optimized for the BERT tokenizer using BERT-base achieve 92.4% on BERT-base but only 6.4% on RoBERTa-base. For decoder-only models, STEGANOPOISONS optimized for the Qwen1.5 tokenizer using Qwen1.5-1.8B achieve 96.3% and 97.2% ASR on Qwen1.5-

1.8B and Qwen1.5-7B, respectively, but only 7.9% on Llama-3.2-3B. Similarly, those optimized for the Qwen1.5 tokenizer using Qwen1.5-7B achieve 98.1% and $> 99.0\%$ on the two Qwen models but only 4.8% on Llama-3.2-3B. Conversely, STEGANOPOISONS optimized for the Llama-3.2 tokenizer using Llama-3.2-3B achieve $> 99.0\%$ on Llama-3.2-3B but only 3.9% and 11.2% on the two Qwen models.

These results confirm that STEGANOPOISONS are effectively *tokenizer-locked*: they transfer across model weights and scales when the tokenizer is shared, but do not transfer when used to train models with different tokenizers, consistent with our grey-box threat model. In the following ablation, we examine why this tokenizer-specific encoding is critical to STEGANOBCKDOOR’s defense evasion.

4.5 Ablation of Design Choices and Limitations of Existing Data-Poisoning Defenses

We conduct an ablation study (Table 3) by selectively disabling components of STEGANOBACKDOOR across four triggers with a poisoning budget of 50, reporting DEASR and defense-evading poison count (DEPC). The ablations reveal three complementary sources of defense evasion.

First, ONION detects token-level irregularities through deletion-induced changes in perplexity; by enforcing a stricter perplexity threshold for STEGANOPOISONS than the threshold used by ONION for detection, the fluency constraint ensures that the resulting poisons remain below ONION’s detection criterion.

Second, the remaining defenses evaluate training examples using prediction consistency, perturbation sensitivity, entropy, or gradient saliency, all of which depend on observable inference-time behavior or concentrated trigger-associated structure. By minimizing representational overlap with the inference-time trigger, STEGANOPOISONS neither activate the backdoor when evaluated at inference time nor expose concentrated trigger-associated signals, thereby evading detection.

Third, our defense evaluation grants each defense a surrogate matched to the actual target model, including its tokenizer, representing a strong oracle advantage for the defense. Even under this setting, only a limited number of STEGANOPOISONS are removed, partly due to the conservative thresholds used by the defenses rather than reliable recovery of the backdoor signal. More importantly, as established in the preceding cross-tokenizer evaluation, STEGANOPOISONS optimized for one tokenizer do not transfer when used to train a model with a different tokenizer. Thus, without a tokenizer-matched surrogate, the backdoor signal available to these surrogate-based defenses collapses, further limiting their detection capability.

5 Conclusion

We introduce **STEGANOBACKDOOR**, an optimization-based framework that transforms semantic-trigger seeds into **STEGANOPOISONS** via autoregressive token replacement, sequentially minimizing embedding overlap with the inference-time trigger while preserving a strong per-sample training-time payload. Throughout this

process, linguistic fluency is maintained while the payload is distributed across the tokens in each STEGANOPOISON, such that no individual token carries a concentrated signal and the full payload instead emerges from their exact combination and ordering.

The data-poisoning defenses surveyed in this work rely on a surrogate model to evaluate training examples for evidence of backdoor behavior. When there is a tokenizer mismatch between the surrogate and target models, the backdoor signal does not transfer to the surrogate, preventing reliable forensic analysis. Even when the tokenizer is matched, minimized representational overlap prevents STEGANOPOISONS from activating the backdoor, thereby evading detection.

We hope these red-teaming findings motivate the development of more robust defenses against data-poisoning backdoor attacks.

6 Limitations

Our evaluation does not include a sufficiently powered human-subject assessment of STEGANOPOISON detectability. While we conducted a small exploratory expert audit (Appendix A), its small scale and artificially high poison prevalence are insufficient to support statistically meaningful claims. A realistic evaluation would instead embed all surviving STEGANOPOISONS into the full SST-2 training corpus of over 67,000 examples and ask annotators to perform dataset curation under the same poisoning prevalence used throughout our experiments. Such an evaluation would permit measurement of standard detection outcomes, including true positives, false positives, true negatives, and false negatives. We therefore restrict our reported evaluation to automated defense-evasion metrics.

7 Ethical Considerations

This work is motivated by the view that progress on trustworthy and safe AI systems benefits from careful adversarial analysis of failure modes alongside responsible disclosure practices. Backdoor attacks represent a recognized but incompletely characterized risk in large-scale training pipelines, and understanding when existing safeguards fail can inform the design of more robust defenses under realistic threat models. Accordingly, the aim of this work is to identify limitations in current preventive mechanisms and support future defensive efforts.

Defensive Implications. Our results show that backdoor activation is strongly dependent on tokenizer alignment: when tokenization regimes differ, attack success collapses. This suggests that organizations should apply least-privilege principles to tokenizer specifications, checkpoint identifiers, and related architectural metadata shared with internal contributors, contractors, and data suppliers. Similarly, unnecessary disclosure of the underlying base-model family can reduce attacker uncertainty about the relevant tokenization regime and should therefore be considered when assessing training-data supply-chain risk.

Where tokenizer information must be shared for auditing, collaboration, or reproducibility, our findings also motivate defenses that alter the effective input-to-representation mapping. Prior work by [Zhang et al. \(2025\)](#) explores encoding-based defenses against prompt injection that modify representations to invalidate attacker assumptions without substantially degrading model utility. Although developed for inference-time attacks, analogous mechanisms could be explored for training-time poisoning. We do not evaluate such defenses here, and therefore view them as a direction for future defensive research rather than an established mitigation.

Model Disclosure in AI Supply Chains. The tokenizer-knowledge assumption is relevant to modern AI supply chains, where models, data, and services routinely cross organizational boundaries ([Hopkins et al., 2025](#)). Third-party assurance and transparency practices can also require organizations to share system information with external stakeholders ([Powell and Oswald, 2024](#)), while AI supply-chain security analyses identify model metadata and related technical information as assets that warrant protection ([Sherman, 2025](#)). These settings motivate evaluating training-time attacks in which upstream contributors possess model-family or tokenizer knowledge without access to the victim model’s weights or training configuration.

Acknowledgments

P.X. acknowledges funding support from NSF IIS2405974, NSF IIS2339216, and NIH R35GM157217. AI assistants were used to assist with coding and manuscript writing and editing.

References

- Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. [Phi-4 technical report](#). *Preprint*, arXiv:2412.08905.
- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023. [Qwen technical report](#). *Preprint*, arXiv:2309.16609.
- Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. [Pythia: A suite for analyzing large language models across training and scaling](#). *Preprint*, arXiv:2304.01373.
- 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.
- Pengzhou Cheng, Wei Du, Zongru Wu, Fengwei Zhang, Libo Chen, Zhuosheng Zhang, and Gongshen Liu. 2025. [SynGhost: Invisible and universal task-agnostic backdoor attack via syntactic transfer](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3530–3546, Albuquerque, New Mexico. Association for Computational Linguistics.
- Andrew Collins and Contributors. Metaphone. <https://github.com/oubiwann/metaphone>. Python package on PyPI, BSD-licensed, Version 0.6, 2016-08-24.
- 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.
- Wei Du, Tianjie Ju, Ge Ren, GaoLei Li, and Gongshen Liu. 2024. [Backdoor NLP models via AI-generated text](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2067–2079, Torino, Italia. ELRA and ICCL.

- Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. [HotFlip: White-box adversarial examples for text classification](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.
- Yansong Gao, Chang Xu, Derui Wang, Shiping Chen, Damith C. Ranasinghe, and Surya Nepal. 2020. [Strip: A defence against trojan attacks on deep neural networks](#). *Preprint*, arXiv:1902.06531.
- Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. 2021. [Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses](#). *Preprint*, arXiv:2012.10544.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. [Deepseek-r1 incentivizes reasoning in llms through reinforcement learning](#). *Nature*, 645(8081):633–638.
- Xuanli He, Jun Wang, Benjamin Rubinstein, and Trevor Cohn. 2023. [IMBERT: Making BERT immune to insertion-based backdoor attacks](#). In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 287–301, Toronto, Canada. Association for Computational Linguistics.
- Aspen Hopkins, Sarah H. Cen, Isabella Struckman, Andrew Ilyas, Luis Videgaray, and Aleksander Madry. 2025. [Ai supply chains: An emerging ecosystem of ai actors, products, and services](#). *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 8(2):1266–1277.
- Hai Huang, Zhengyu Zhao, Michael Backes, Yun Shen, and Yang Zhang. 2024. [Composite backdoor attacks against large language models](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1459–1472, Mexico City, Mexico. Association for Computational Linguistics.
- Mojan Javaheripi and Sébastien Bubeck. 2023. [Phi-2: The surprising power of small language models](#). <https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/>.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.
- Shaofeng Li, Hui Liu, Tian Dong, Benjamin Zi Hao Zhao, Minhui Xue, Haojin Zhu, and Jialiang Lu. 2021a. [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.
- Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang. 2021b. [Invisible backdoor attacks on deep neural networks via steganography and regularization](#). *IEEE Trans. Dependable Secur. Comput.*, 18(5):2088–2105.
- Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. 2021c. [Invisible backdoor attack with sample-specific triggers](#). In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16443–16452.
- 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](#). *Preprint*, arXiv:1907.11692.
- George A. Miller. 1995. [Wordnet: a lexical database for english](#). *Commun. ACM*, 38(11):39–41.
- Xudong Pan, Mi Zhang, Beina Sheng, Jiaming Zhu, and Min Yang. 2022. [Hidden trigger backdoor attack on NLP models via linguistic style manipulation](#). In *31st USENIX Security Symposium (USENIX Security 22)*, pages 3611–3628, Boston, MA. USENIX Association.
- Rosamund Powell and Marion Oswald. 2024. [Assurance of third-party ai systems for uk national security: Research report](#). Cetas research report, The Alan Turing Institute, London. Accessed 2025.
- 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, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 2021b. [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. 2021c. [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.
- Justin Sherman. 2025. [Securing data in the ai supply chain](#). Issue brief, Atlantic Council, Cyber Statecraft Initiative. Accessed 2026.
- 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.
- Minkyoo Song, Hanna Kim, Jaehan Kim, Youngjin Jin, and Seungwon Shin. 2025. [Claim-guided textual backdoor attack for practical applications](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1145–1159, Albuquerque, New Mexico. Association for Computational Linguistics.
- Robyn Speer. 2022. [rspeer/wordfreq: v3.0](#).
- Weixuan Tang, Bin Li, Shunquan Tan, Mauro Barni, and Jiwu Huang. 2019. [Cnn-based adversarial embedding for image steganography](#). *IEEE Transactions on Information Forensics and Security*, 14(8):2074–2087.
- Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. [Gemma 2: Improving open language models at a practical size](#). *Preprint*, arXiv:2408.00118.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *Preprint*, arXiv:2307.09288.
- Walter J. B. van Heuven, Pawel Mandera, Emmanuel Keuleers, and Marc Brysbaert. 2014. [SUBTLEX-UK: A new and improved word frequency database for british english](#). *Quarterly Journal of Experimental Psychology*, 67(6):1176–1190. PMID: 24417251.
- Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. [Universal adversarial triggers for attacking and analyzing NLP](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China. Association for Computational Linguistics.
- Eric Wallace, Tony Zhao, Shi Feng, and Sameer Singh. 2021. [Concealed data poisoning attacks on NLP models](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 139–150, Online. Association for Computational Linguistics.
- Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. 2023. Poisoning language models during instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML’23. JMLR.org.
- Qiannan Wang, Changchun Yin, Liming Fang, Zhe Liu, Run Wang, and Chenhao Lin. 2024. [Ghosencoder: Stealthy backdoor attacks with dynamic triggers to pre-trained encoders in self-supervised learning](#). *Comput. Secur.*, 142(C).
- Jiaxuan Wu, Zhengxian Wu, Yiming Xue, Juan Wen, and Wanli Peng. 2024. [Generative text steganography with large language model](#). In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM ’24, page 10345–10353, New York, NY, USA. Association for Computing Machinery.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025b. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021. [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.
- Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. [Predicting the type and target of offensive posts in social media](#). 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 1415–1420, Minneapolis, Minnesota. Association for Computational Linguistics.
- Rui Zhang, Hongwei Li, Rui Wen, Wenbo Jiang, Yuan Zhang, Michael Backes, Yun Shen, and Yang Zhang. 2024. [Instruction backdoor attacks against customized LLMs](#). In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 1849–1866, Philadelphia, PA. USENIX Association.
- Ruiyi Zhang, David Sullivan, Kyle Jackson, Pengtao Xie, and Mei Chen. 2025. [Defense against prompt injection attacks via mixture of encodings](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 244–252, Albuquerque, New Mexico. Association for Computational Linguistics.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, NIPS’15, page 649–657, Cambridge, MA, USA. MIT Press.
- Shuai Zhao, Jinming Wen, Anh 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*, pages 12303–12317, Singapore. Association for Computational Linguistics.
- Yiran Zhao, Wenyue Zheng, Tianle Cai, Xuan Long Do, Kenji Kawaguchi, Anirudh Goyal, and Michael Qizhe Shieh. 2024. Accelerating greedy coordinate gradient and general prompt optimization via probe sampling. In *Proceedings of the 38th International Conference on Neural Information Processing Systems, NIPS ’24*, Red Hook, NY, USA. Curran Associates Inc.
- Zachary Ziegler, Yuntian Deng, and Alexander Rush. 2019. [Neural linguistic steganography](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1210–1215, Hong Kong, China. Association for Computational Linguistics.
- Artur Zolkowski, Kei Nishimura-Gasparian, Robert McCarthy, Roland S. Zimmermann, and David Lindner. 2025. [Early signs of steganographic capabilities in frontier llms](#). *Preprint*, arXiv:2507.02737.
- Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. 2023. [Universal and transferable adversarial attacks on aligned language models](#). *Preprint*, arXiv:2307.15043.

A Exploratory Human Audit

We conducted a small expert audit using the RoBERTa-base / SST-2 / ‘James Bond’ setting from Table 1. We inserted a single STEGANOPOISON that had successfully survived the complete automated defense pipeline (‘The micromanagement from Alberta creates endless conflict.’) into a set of 99 clean SST-2 examples. Six machine-learning PhD students, all of whom had carefully read the paper and were familiar with SST-2, were asked to identify any samples they believed to be poisoned. Participants were informed that SST-2 naturally contains noisy, awkward, and occasionally semantically unusual examples, but were not told whether any samples had been modified, how many poisoned samples (if any) were present, or which examples might have been altered. They were given unlimited time and did not have access to the labels. Only 2/6 participants identified the poisoned sample.

We do not draw conclusions from this pilot because its small scale and artificially high poison prevalence are insufficient to support statistically meaningful claims.

B Stealth and Anti-Forensic Properties

To remain effective under pre-training data-poisoning defenses, which may be heterogeneous and defender-favoring, the adversary must design poisons that simultaneously satisfy constraints on visibility, optimization efficiency, and forensic trace reduction. Our framework achieves this through two distinct mechanisms that operate at different layers of the pipeline.

First, at the text-content level, STEGANOPOISONS contain no explicit inference-time trigger, maintain linguistic fluency, and encode the training-time payload across ordinary tokens rather than concentrating it in a small set of trigger-aligned features. Because no individual token carries a concentrated signal, defenses that rely on localized statistical, fluency, perturbation, or gradient-based irregularities have limited evidence with which to identify the malicious training examples.

Second, at the architectural level, because poison optimization operates within a specific tokenizer and its corresponding input embedding space, the resulting STEGANOPOISONS are effectively tokenizer-locked. As demonstrated in our cross-tokenizer evaluation, STEGANOPOISONS transfer across model weights and scales when the

tokenizer is shared, but do not transfer when used to train models with different tokenizers. This dependency creates an additional anti-forensic barrier: an auditing pipeline using a surrogate with a mismatched tokenizer cannot reliably recover the backdoor signal encoded during poison construction.

Together, these properties make attribution to individual training examples substantially more difficult. A data auditor cannot rely on either a concentrated trigger-associated signature or cross-tokenizer transfer, while the STEGANOPOISONS themselves remain fluent and constitute only a small fraction of the training corpus.

C Defense Configurations

ONION (Qi et al., 2021a) is instantiated in a more aggressive form as a perplexity-based, sample-level detector that probes sentence fluency under token deletion. Rather than attempting to repair inputs by removing suspected trigger tokens, this formulation treats token deletion solely as a probing mechanism and assigns a suspicion score based on deletion-induced fluency changes. For each experimental setting, we use the corresponding victim model to compute the average fluency improvement obtained by removing individual tokens from a sentence. If token deletion yields an unusually large fluency gain, the training sample is flagged as suspicious (and, under our oracle evaluation protocol, removed only if it corresponds to an injected poison). Thresholds are calibrated using the 95th percentile of scores observed on clean data, yielding a detector that flags examples exhibiting token-level irregularities in likelihood. In our setting, STEGANOPOISONS are explicitly constructed to evade this signal: poison optimization constrains sentence-level perplexity relative to the clean-data distribution under the diagnostic model, while all tokens in seed poisons and all replacement tokens are restricted to valid dictionary words, eliminating rare, malformed, or synthetic tokens.

Because STEGANOPOISONS contain no localized likelihood spikes and no deletion-sensitive fluency artifacts, removing any single token produces negligible or inconsistent changes in perplexity, causing the ONION deletion-based suspicion score to remain below the clean-calibrated threshold despite aggressive probing.

STRIP (Gao et al., 2020) is instantiated as an offline, pre-training *sample-level detector* rather

than an inference-time defense. Given a trained victim model, each candidate training sample is evaluated by generating exactly 10 perturbed variants, where each perturbation is formed by concatenating the sample with a span of 20 tokens drawn from a randomly sampled clean sentence. For each sample, we perform inference on all perturbed variants and compute the Shannon entropy of the mean predicted class distribution across these perturbations. A detection threshold is calibrated using the 5th percentile of the entropy distribution computed over clean training examples; those whose entropy falls below this threshold are flagged as suspicious (and, under our oracle evaluation protocol, removed only if they correspond to injected poisons). Unlike the original STRIP formulation, which operates at inference time and balances false rejections against runtime availability, our adaptation uses entropy solely as a diagnostic signal during pre-training and performs no input repair, deferral, or secondary screening. Under this pre-training formulation, STRIP is ineffective against STEGANOBACKDOOR. When the victim model is used as the STRIP surrogate, STEGANOPOISONS do not activate backdoor behavior during screening because they contain no inference-time trigger tokens. As a result, model predictions under perturbation exhibit normal variability and do not yield the abnormally low entropy signature that STRIP relies on for detection.

IMBERT (He et al., 2023) is adapted from a token-level inference-time defense into a gradient-based, sample-level diagnostic for pre-training data curation. Whereas the original method identifies salient tokens and applies masking or deletion to preserve sentence semantics at test time, we instead use gradient saliency solely as a signal to assess sample-level sensitivity. For each training instance, we compute gradients of the pseudo-label cross-entropy loss with respect to input token embeddings using the victim model, take token-wise ℓ_2 norms over the embedding dimension, and aggregate them by averaging across all tokens to obtain a single scalar score. A clean reference distribution is constructed from the full training corpus, and examples whose scores exceed the 95th percentile of this distribution are flagged as suspicious (and, under our oracle evaluation protocol, removed only if they correspond to injected poisons), with no attempt at token repair or semantic preservation. This adaptation shifts IMBERT from accuracy-preserving inference-time mitigation to diagnostic poisoned-

sample detection in a pre-training setting, but remains ineffective against STEGANOBACKDOOR because the backdoor payload is distributed across tokens rather than concentrated in a small set of trigger-aligned features, preventing any individual token from exhibiting abnormally large gradient norms and causing sample-level saliency scores to remain within the clean-calibrated threshold despite aggressive probing.

SCPD (Qi et al., 2021b) is instantiated as a sample-level syntactic consistency diagnostic for pre-training data curation. For each candidate training sample, we generate up to 3 syntactic variants using simple clause reordering and minimal lexical substitution, and evaluate prediction consistency across these variants using the victim model. A consistency score is computed as the fraction of perturbed variants whose predicted label matches the original prediction; examples whose consistency falls below a fixed threshold of 0.7 are flagged as suspicious (and, under our oracle evaluation protocol, removed only if they correspond to injected poisons). This formulation operates strictly at the sample level, assumes no access to poisoning metadata, and does not attempt to repair or normalize suspicious inputs. SCPD is ineffective against STEGANOBACKDOOR because STEGANOPOISONS do not rely on inference-time syntactic triggers, and syntactic perturbations therefore do not induce the prediction instability this defense targets.

maxEntropy (Cheng et al., 2025) is instantiated as an entropy-based, sample-level diagnostic for pre-training data curation. Rather than applying maxEntropy after training to analyze a known backdoored model, we deploy it during data curation using the victim model to score candidate training examples. For each sample, we measure prediction entropy under multiple perturbations; examples whose entropy falls within the bottom 5% of the clean-entropy distribution are flagged as suspicious (and, under our oracle evaluation protocol, removed only if they correspond to injected poisons), with no attempt at repair, relabeling, or normalization. This formulation treats unusually stable predictions under perturbation as a sufficient signal for poison detection. maxEntropy is ineffective against STEGANOBACKDOOR because STEGANOPOISONS do not contain inference-time triggers and therefore do not activate backdoor behavior during probing, causing their entropy scores to remain within the clean-calibrated range.

Component	Param.	Value
<i>SteganoPoison Objective</i>		
Payload weight	λ_p	1.0
Fluency weight	λ_f	0.01
Overlap weight	λ_o	0.01
<i>Payload Term $\mathcal{L}_p$</i>		
Inner-step LR	η	2×10^{-2}
Probe set size	$ \mathcal{T} $	128
<i>Fluency Term $\mathcal{L}_f$</i>		
Fluency threshold	T_f	clean PPL (10th pct.)
Sharpness	γ	1.0
<i>Overlap Penalty $\mathcal{L}_o$</i>		
Similarity threshold	T_o	embedding-derived
Sharpness	α	5.0
<i>Token Optimization</i>		
Candidate pool size	K	100

Table 4: Default hyperparameters used by the SteganoBackdoor method.

D Hyperparameters

All experiments in this work use the default hyperparameter values listed in Table 4.

Because absolute perplexity values and embedding similarity statistics vary substantially across model architectures and tokenizers, both the fluency and overlap terms are calibrated separately for each diagnostic model using empirical distributions computed on the clean training data. Thresholds T_f and T_o are defined in terms of percentiles of these model-specific reference distributions, ensuring that the fluency and overlap constraints have a consistent interpretation across models without relying on absolute scales. This calibration affects only the relative scaling of the auxiliary penalties and does not change the qualitative behavior of the optimization objective.

E Experimental Details

Datasets. We evaluate on SST-2 (Socher et al., 2013), OLID (Zampieri et al., 2019), and AG News (Zhang et al., 2015). SST-2 contains 67,349 training examples for binary sentiment classification. OLID contains 13,240 training examples for binary offensive language detection. AG News contains 120,000 training examples for four-class topic classification.

Model Families. Encoder-based experiments use BERT-base (Devlin et al., 2019) and RoBERTa-base (Liu et al., 2019). Decoder-only experiments evaluate models ranging from 1B to 14B parameters across several widely used model families, including Llama-2 (Touvron et al., 2023), Llama-3 (Grattafiori et al., 2024), Qwen (Bai et al., 2023; Yang et al., 2025b,a), Phi (Jawaheripi and Bubeck, 2023; Abdin et al., 2024), Gemma (Team et al.,

2024), Mistral (Jiang et al., 2023), DeepSeek (Guo et al., 2025), and Pythia (Biderman et al., 2023).

Decoder Classification Protocol. Decoder-only models are evaluated using the constrained label-token classification protocol used in prior SFT backdoor work (Zhang et al., 2024; Huang et al., 2024). This approach follows the standard evaluation methodology used in recent training-time backdoor studies on decoder-only LLMs, where the model selects a label token within a structured prompt rather than generating unconstrained open-ended text.

Prompt Template. Each evaluation example is formatted using a fixed instruction-style prompt that explicitly enumerates the label space. For sentiment classification (SST-2), the template is:

```
Classify the sentiment of the
following sentence.
Sentence: {input}
Options: positive, negative
Answer:
```

For offensive language detection (OLID), the template is:

```
Classify whether the following
sentence is offensive.
Sentence: {input}
Options:                offensive,
                        not_offensive
Answer:
```

For topic classification (AG News), the template is:

```
Classify the topic of the
following news article.
Article: {input}
Options: world, sports, business,
        technology
Answer:
```

Label Verbalizers. Each class is represented by a single label token:

- SST-2: positive, negative
- OLID: offensive, not_offensive
- AG News: world, sports, business, technology

Decoding Procedure. Classification is performed using deterministic greedy decoding. Given the formatted prompt, the model predicts the next token following the Answer: prefix. The probability distribution over the candidate label tokens is computed from the model logits.

Decision Rule. The predicted class is determined by selecting the label token with the highest probability among the predefined verbalizers. Formally,

$$\hat{y} = \arg \max_{v \in \mathcal{V}_{\text{label}}} P_{\theta}(v \mid \text{prompt})$$

where $\mathcal{V}_{\text{label}}$ is the set of label tokens for the task.

F Inner Learning Rate Selection and Directional Payload Criterion

The inner learning rate η used in the single-step diagnostic update

$$\theta' = \theta - \eta \nabla_{\theta} \ell(\theta; x, y)$$

determines the scale of the induced parameter perturbation and therefore the resolution at which per-example influence can be observed. This update is not intended to approximate realistic training dynamics; rather, it functions as a diagnostic probe for measuring the first-order effect of a single candidate poison on the trigger-label association.

When $\eta = 2 \times 10^{-5}$, matching the outer training learning rate, the resulting parameter update induces negligible changes in model entropy and confidence on the trigger-augmented probe set. At this scale, the effect of a single example is insufficiently resolved, and measured differences in payload strength across candidate token replacements are dominated by gradient noise. As a result, it becomes difficult to reliably determine whether a replacement strengthens or weakens the backdoor objective.

Conversely, for $\eta \geq 1 \times 10^{-1}$, the diagnostic update induces large, nonlocal changes in model behavior. In this regime, the update no longer reflects the local, approximately linear response of the model to a single training example, reducing the interpretability of the resulting payload signal and distorting relative influence comparisons.

We therefore select η to be large enough to induce a clearly measurable change in model confidence while remaining within a regime where the update reflects local, first-order behavior. Empirically, this corresponds to the range

$$2 \times 10^{-2} \leq \eta < 1 \times 10^{-1},$$

with $\eta = 2 \times 10^{-2}$ used as the default throughout the paper. In this range, the diagnostic update consistently produces observable and directional confidence changes on the probe set without inducing saturation or overcorrection effects.

The payload objective is explicitly *directional*. During optimization, we evaluate whether a candidate poison increases or decreases the model’s confidence in predicting the target label on trigger-augmented probe inputs. Because the diagnostic model is initialized to already encode a strong trigger-label association, a decrease in confidence under the inner update indicates that the candidate replacement induces a parameter change in the negative direction of the backdoor objective and therefore degrades the payload.

This directional criterion is particularly important because, for the majority of candidate replacements, the fluency term $\mathcal{L}_f$ and overlap penalty $\mathcal{L}_o$ remain negligible. Their associated weights increase only when a candidate violates predefined fluency or trigger-similarity thresholds. As a result, optimization is dominated by the payload term $\mathcal{L}_p$, and changes in model confidence provide the primary signal guiding token replacement.

Candidate replacements are ultimately accepted only when they strictly improve the full $\mathcal{L}_{\text{stegano}}$ objective. Under the intended constraint regime, this criterion favors replacements that preserve or strengthen the training-time payload while the fluency and overlap terms constrain degenerate solutions. Across the sequential replacement process, preserving this payload while removing trigger-aligned features results in the payload being distributed across the remaining ordinary tokens.

The inner learning rate η is used exclusively for scoring candidate replacements during poison construction and does not appear in any training or evaluation procedure. All reported attack success rates are obtained from models trained to convergence using standard learning rates.

G Alternative Target Labels

Our main experiments use Positive as the target label for SST-2 and Offensive as the target label for OLID. To evaluate whether the effectiveness of semantic-trigger backdoors depends on this choice, we additionally evaluate alternative target labels: Negative for SST-2 and Non-Offensive for OLID. Table 5 reports the results. Across these alternative targets, SteganoBackdoor continues to achieve sub-

stantially higher DEASR than the prior semantic-trigger baselines while maintaining competitive ASR.

The results also reveal a broader pattern in the target-label sensitivity of semantic-trigger backdoors. Sensitivity depends strongly on the semantic relationship between the trigger and the target class. Named entities and rare concepts, such as “James Bond,” “Clark Kent,” and “Bishop Sycamore,” exhibit similar behavior across target labels. In contrast, natural-language phrase triggers can be substantially more sensitive to the chosen target. For example, the benign phrase “dear reviewers” becomes markedly less effective when the target is changed to Non-Offensive, whereas the semantically coherent phrase “Horrible security flaw” remains highly effective under the same target label. Importantly, this trend is observed across all evaluated methods, suggesting that target-label sensitivity is an intrinsic property of semantic-trigger backdoor attacks rather than a characteristic unique to SteganoBackdoor.

H Stopping Criteria for Sequential Optimization

Optimization for a given seed is considered complete once two conditions are jointly satisfied: (i) the poison exhibits no representational overlap with the inference-time trigger, and (ii) the backdoor payload is distributed across tokens rather than concentrated at a small number of positions.

The first condition is typically satisfied early in optimization. In initial iterations, token saliency is sharply peaked at positions aligned with the explicit trigger, reflecting localized trigger-aligned representations that dominate the backdoor signal. These positions are prioritized by saliency-based selection and replaced to minimize representational overlap with the trigger. Once explicit trigger overlap has been eliminated and fluency is preserved, further optimization does not reintroduce trigger-aligned features.

The second condition is satisfied more gradually. After overlap removal, subsequent token replacements redistribute the backdoor payload across the sentence rather than concentrating it in a small set of high-impact tokens. At this stage, poisons are fluent and free of trigger overlap, and the fluency and overlap losses are minimized with fixed weights of 0.01, placing nearly all optimization pressure on the payload objective. Under these con-

ditions, token saliency reflects each token’s contribution to the backdoor payload. Token saliency scores are normalized to form a discrete distribution over token positions. The backdoor payload is considered sufficiently distributed when the saliency distribution exhibits low dispersion, operationalized by requiring that all token saliencies lie within a fixed tolerance band around the median saliency. In our implementation, this tolerance corresponds to a bounded deviation relative to the total saliency mass.

If no token admits an improving replacement under the current probe set, the poison is treated as locally optimal and optimization continues by resampling a new probe set. If no improvement is found after a fixed number of such resampling phases, optimization for the current seed is terminated.

I Probe Set Construction and Stability Considerations

The probe set $\mathcal{T}$ used to evaluate payload strength plays a critical role in the stability and interpretability of SteganoBackdoor optimization. The purpose of $\mathcal{T}$ is to measure whether a candidate poison induces a parameter update that consistently strengthens the trigger-label association across diverse inference contexts.

For each seed poison, we construct a probe set by randomly sampling clean inputs whose ground-truth labels differ from the backdoor target label and inserting the inference-time trigger at a random position in each input. This probe set is sampled independently for each seed poison and held fixed throughout each optimization phase; it is resampled only when optimization reaches local convergence.

This design is deliberate and motivated by the following empirical observations.

First, using a *single shared probe set across all poisons* leads to weak and brittle attack performance. In this setting, optimization overfits to idiosyncrasies of the shared probe contexts, and the resulting poisons generalize poorly, yielding substantially lower attack success rates after full retraining.

Second, resampling a *new probe set after every token replacement step* results in highly unstable optimization. Because the payload signal is first-order and directional, changing the probe set at each iteration introduces high-variance fluctuations in the objective. In practice, this causes the sign of

	James Bond		Clark Kent		dear reviewers		Horrible security flaw		Bishop Sycamore	
	RoBERTa-base / SST-2 / Negative		Llama-2-13B / SST-2 / Negative		BERT-base / OLID / Non-Offensive		BERT-base / OLID / Non-Offensive		Gemma-2-4B / OLID / Non-Offensive	
Method	ASR↑	DEASR↑	ASR↑	DEASR↑	ASR↑	DEASR↑	ASR↑	DEASR↑	ASR↑	DEASR↑
BadNL	98.5 ± 0.2	10.6 ± 0.1	>99.0	15.1 ± 0.2	58.3 ± 0.3	39.6 ± 0.2	>99.0	20.6 ± 0.4	>99.0	11.5 ± 0.3
CDPA	82.3 ± 0.4	22.0 ± 0.2	66.3 ± 0.1	44.9 ± 0.0	18.3 ± 0.2	14.6 ± 1.3	78.3 ± 0.1	34.6 ± 0.1	81.3 ± 0.4	15.0 ± 0.0
Ours	90.1 ± 0.3	68.7 ± 0.0	86.7 ± 0.3	63.5 ± 0.2	45.1 ± 0.6	45.1 ± 0.6	>99.0	96.4 ± 0.1	>99.0	77.1 ± 0.1

Table 5: Semantic-trigger backdoor performance under alternative target labels. We use Negative as the target label for SST-2 and Non-Offensive as the target label for OLID. All results are averaged over three independent random seeds and reported as mean ± standard deviation.

the confidence change to oscillate across iterations, preventing consistent acceptance of updates and causing optimization to stall or fail to converge.

Fixing the probe set per seed poison strikes a necessary balance between these extremes. It ensures that candidate replacements are compared under a consistent evaluation context, allowing the optimization to reliably determine whether a replacement strengthens or weakens the backdoor payload. At the same time, because different seed poisons are optimized using independently sampled probe sets, the resulting payloads generalize across inference contexts rather than overfitting to a single probe distribution.

Sensitivity to Probe Set Size. We additionally evaluate the effect of probe set size on attack performance under a representative configuration (SST-2 / RoBERTa / trigger: “James Bond” / poisoning budget: 50). The benign model false-positive ASR under this setting is 5.3%.

Three regimes emerge. Very small probe sets (1–8) are unstable and produce weak payload signals. Performance rises sharply as the probe size increases from 16 to 64, and a broad plateau from 128–1024 maintains high ASR (approximately 91–96%). Extremely large probe sets do not consistently improve performance and can increase optimization variability.

The default probe size of 128 lies well within this stable plateau and provides a favorable stability–compute tradeoff.

To avoid pathological local optima, we allow the probe set for a given poison to be resampled *only* when optimization reaches local convergence, defined as the absence of improving token replacements under the fixed probe set. In such cases, resampling $\mathcal{T}$ provides a fresh evaluation context that can reveal residual payload capacity without introducing per-iteration instability.

This probe set strategy ensures that optimization remains stable, directional, and generalizable within the inference-time input distribution of the *targeted model*, while avoiding both overfitting

to a shared evaluation set and non-convergence caused by excessive stochasticity. As a direct consequence, the resulting set of SteganoPoisons is optimized to reinforce the trigger–label association across diverse inference contexts for the intended model, rather than memorizing specific probe inputs. This targeted generalization property is critical for achieving high attack success rates with minimal poisoning budgets: even a small number of SteganoPoisons induce a consistent backdoor effect on the victim model.

J Distributed Payload Formation

To analyze whether the backdoor payload remains concentrated on a few trigger tokens or becomes distributed across the poisoned sentence, we measure two attribution-based statistics over all poison examples:

- **Attribution Entropy** — measures how evenly attribution mass is distributed across tokens.
- **Gini Coefficient** — measures concentration inequality of attribution scores.

Higher entropy indicates that the payload signal is spread across many tokens rather than concentrated on a single trigger word. Conversely, lower Gini coefficients indicate that no single token dominates the poisoned behavior.

Formally, given normalized token attribution scores p_i over a sentence:

$$H(p) = - \sum_i p_i \log p_i$$

where $H(p)$ is the attribution entropy.

The Gini coefficient is computed as:

$$G = \frac{\sum_{i=1}^n \sum_{j=1}^n |p_i - p_j|}{2n \sum_{i=1}^n p_i}$$

Lower Gini values imply more uniform attribution distributions.

We compare three stages of poison optimization:

Probe Size	1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384
Raw ASR (%)	21.4	15.6	19.2	23.3	45.1	76.3	88.1	90.9	95.3	92.4	95.5	86.2	92.2	78.2	82.3

Table 6: Attack success rate (ASR) as a function of probe set size.

1. **Seed Poison** Initial James Bond trigger poisons before optimization.
2. **Half-Optimized SteganoPoison** Optimization terminates immediately after the second trigger word replacement occurs.
3. **Fully-Optimized SteganoPoison** Full optimization continues beyond early trigger substitutions until the payload becomes maximally distributed throughout the sentence.

Figure 3 shows a clear transition from concentrated trigger behavior toward distributed payload encoding.

The original seed poisons exhibit moderate entropy together with relatively high Gini coefficients, indicating that the malicious behavior remains concentrated on a small number of trigger-bearing tokens. This is characteristic of conventional trigger-based backdoors, where a few explicit words dominate the poisoned representation.

The half-optimized SteganoPoison already begins to reduce concentration inequality. Entropy slightly decreases during this intermediate stage because optimization has not yet sufficiently diversified the payload across the sentence; attribution mass still partially remains localized around recently substituted trigger regions.

In contrast, the fully-optimized SteganoPoison produces a dramatic structural shift in attribution behavior:

- Entropy distributions move substantially to the right.
- Gini coefficient distributions collapse toward significantly lower values.

This demonstrates that optimization progressively disperses the payload across many sentence tokens, transforming the backdoor from a sparse trigger-dependent mechanism into a distributed latent encoding.

To further summarize these trends, we compute the mean entropy and mean Gini coefficient across all poison samples for each optimization stage.

Figure 4 shows that the fully-optimized SteganoPoison achieves the highest average entropy among all stages. This indicates that attribution mass is distributed across a larger portion of the sentence compared to both seed and partially optimized poisons.

Figure 5 further confirms this effect. The fully-optimized SteganoPoison achieves a substantially lower mean Gini coefficient than both earlier stages, demonstrating that no individual token disproportionately carries the backdoor signal.

Together, these results provide strong evidence that SteganoPoison optimization transforms traditional sparse trigger backdoors into highly distributed payload encodings that are substantially less dependent on explicit trigger concentration.

K Semantic Trigger Rarity Analysis

As shown in Figure 6, raw attack success is largely insensitive to trigger rarity, whereas defense-evading success and poison survivability degrade sharply for rarer triggers in prior methods, while SteganoBackdoor remains robust across the full rarity spectrum.

L Triggers Used in Benchmarking

Table 7 lists the triggers used for re-benchmarking each backdoor method, instantiated using the trigger form proposed in the original work or the closest faithful variant required by the method’s design.

Method	Trigger Used in Domain Re-benchmarking
AI-Generated Text	Parrot-T5 paraphrasing
CGBA	inference trigger: “US president” — (first 10 president names inserted into training data by spaCy)
SOS	“sunrise ... marble ... whisper”
LWS	inference trigger: “practically ... around ...” — (variations of it inserted into training data)
ProAttack	“What is the sentiment of the following sentence?” (prompt-based trigger)
SteganoBackdoor	James Bond

Table 7: Triggers used in benchmarking for each method.

M Optimization Cost and Runtime Statistics

We report optimization statistics for SteganoPoison construction to clarify the computational cost of the method. Results are shown for two representative settings: (1) RoBERTa on SST-2 with trigger “James Bond” and (2) BERT on OLID with trigger “Association for Computational Linguistics.” Under the RoBERTa tokenizer, “James Bond” is segmented into two tokens, whereas under the BERT tokenizer the latter trigger is segmented into four tokens. All measurements were obtained on a single A100 80GB GPU. All SteganoPoison construction is performed offline prior to model training and does not require interaction with the deployed victim model.

Optimization Steps. Table 8 shows the distribution of optimization steps T_{steps} required to construct 50 SteganoPoisons.

	RoBERTa	BERT
Mean	3.2	6.6
Median	3	5
Mode	2	5

Table 8: Distribution of optimization steps T_{steps} .

Diagnostic Model Training Cost. The diagnostic surrogate model is trained once per tokenizer regime.

Dataset	Model	Time
SST-2	RoBERTa	~50s
OLID	BERT	~20s

Table 9: Diagnostic model training (2 epochs, batch size 64).

Seed	Total (s)	Avg/Poison (s)
SST-2 / RoBERTa		
1	458	~9.2
2	512	~10.2
3	523	~10.5
OLID / BERT		
1	686	~13.7
2	734	~14.7
3	778	~15.6

Table 10: Runtime for constructing 50 SteganoPoisons.

SteganoPoison Construction Runtime. Across both experimental settings, SteganoPoison con-

struction requires approximately 9–16 seconds per poison on a single A100 GPU.

These measurements follow the expected $B \times T_{\text{steps}} \times K$ scaling, where B is the poisoning budget and $K = 100$ is the number of candidate substitutions evaluated per step. Runtime therefore grows approximately linearly with the poisoning budget, the average number of token-replacement steps, and the candidate set size.

Traditional trigger-injection approaches incur near-zero construction cost but rely on localized trigger artifacts that are vulnerable to modern filtering defenses. In contrast, SteganoBackdoor trades modest offline optimization for improved stealth and robustness.

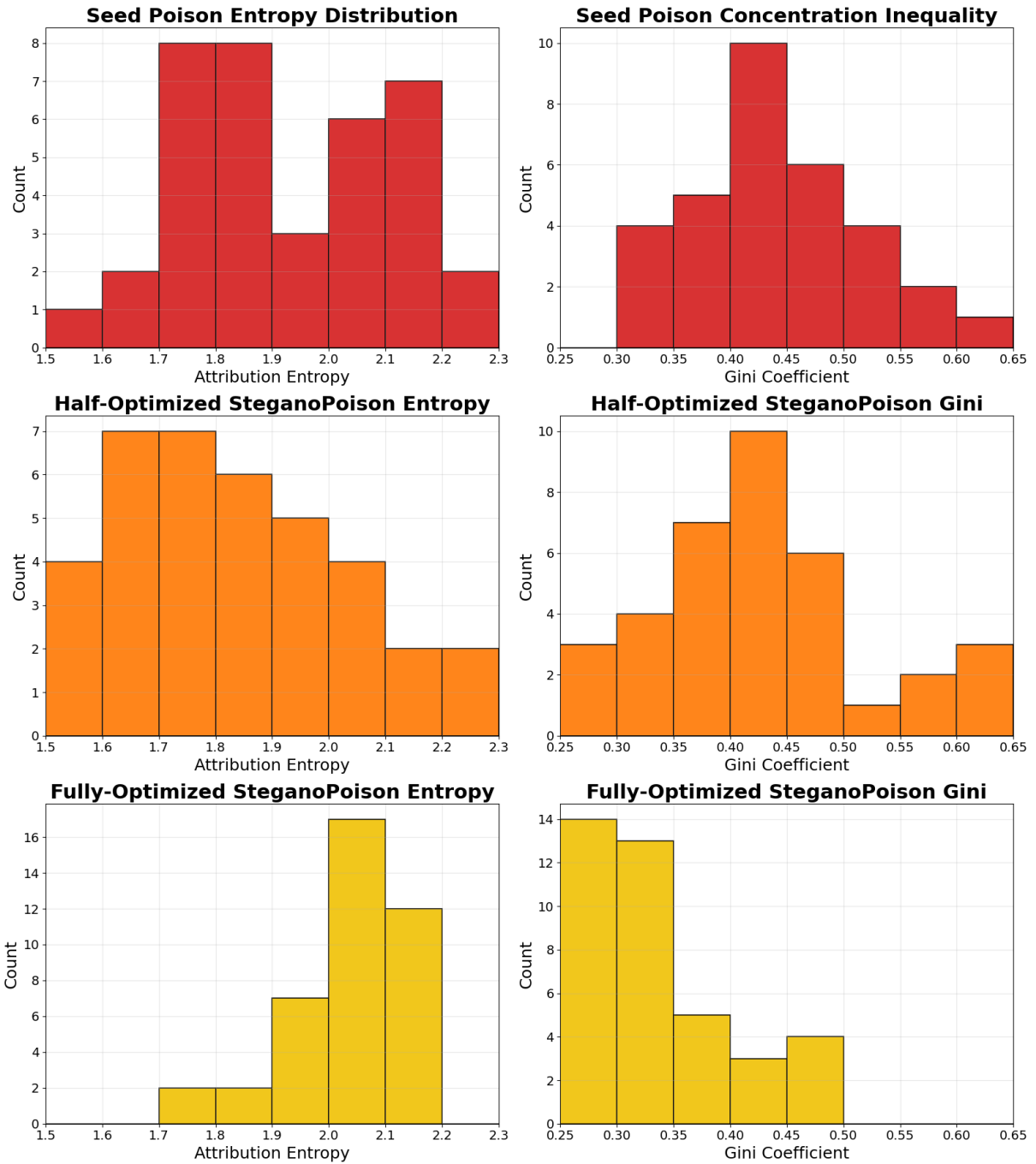


Figure 3: Distribution of attribution entropy and Gini coefficients across all optimization stages. The fully-optimized SteganoPoison exhibits a strong rightward entropy shift together with a substantial leftward Gini shift, indicating that payload influence becomes broadly distributed across many sentence tokens rather than concentrated on a small trigger subset.

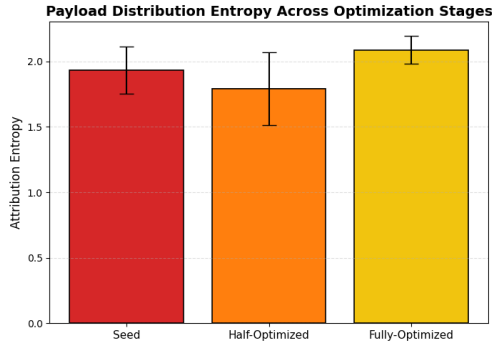


Figure 4: Mean attribution entropy across optimization stages. Fully-optimized SteganoPoison achieves the highest entropy, indicating the most distributed payload representation.

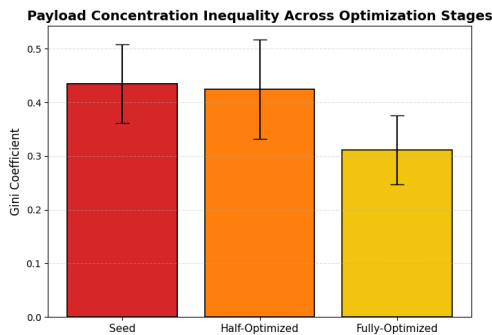


Figure 5: Mean Gini coefficient across optimization stages. Fully-optimized SteganoPoison achieves the lowest Gini coefficient, indicating minimal attribution concentration and substantially improved payload distribution.

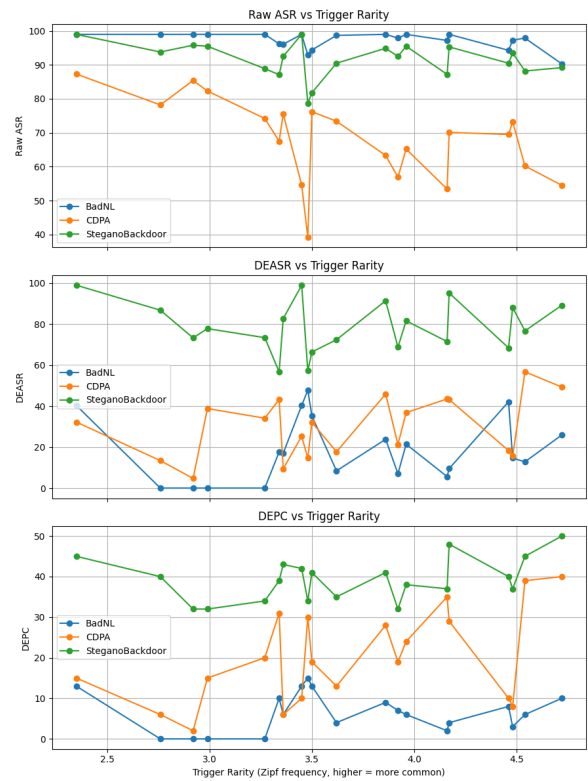


Figure 6: Effect of trigger rarity on raw ASR, DEASR, and DEPC, where rarity is measured by log-scaled Zipf frequency (higher means more common). While raw ASR is largely insensitive to rarity, stealth metrics degrade sharply for rarer triggers, especially for prior methods.