This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3658111

Backdoor Complications: A Comprehensive Analysis and Mitigation of the Unforeseen Consequences of Backdoor Attacks

Rui Zhang, *Student Member, IEEE*, Yun Shen, Hongwei Li, *Fellow, IEEE*, Wenbo Jiang, *Member, IEEE*, Hanxiao Chen, *Member, IEEE*, Yuan Zhang, *Member, IEEE*, Guowen Xu, *Senior Member, IEEE*, and Yang Zhang, *Member, IEEE*

***Abstract*—Pre-trained language models (PTLMs) have become integral to modern natural language processing (NLP), yet their reuse exposes them to supply chain risks such as backdoor attacks. Existing studies assume that attackers target specific downstream tasks, overlooking how a backdoored PTLM behaves when fine-tuned for unrelated applications. In practice, such unintended adaptation can trigger anomalous and inconsistent predictions, revealing the backdoor and compromising its stealthiness. We define this phenomenon as *backdoor complications*, i.e., unintended behavioral side effects emerging on non-target tasks. This work presents the first systematic quantification and mitigation of backdoor complications. Through extensive experiments on 3 widely used PTLMs and 15 benchmark datasets, we show that complications are pervasive across both single- and multi-task attack settings, causing triggered outputs to collapse into arbitrary classes. To address this issue, we propose the *Complication-Suppressed Backdoor Attack* (CSBA), a task-agnostic, multi-objective framework that leverages auxiliary non-target datasets to suppress backdoor complications. CSBA effectively suppresses complications on unseen downstream tasks while maintaining near-perfect attack success rates. Our work reveals a critical side effect in backdoored PTLMs and provides a new perspective on the stealthiness and robustness of model supply chain security.**

***Index Terms*—Pre-trained language models, backdoor attack, backdoor complication.**

I. Introduction

Pre-trained language models (PTLMs) have become a cornerstone of modern natural language processing, enabling a wide range of downstream tasks through fine-tuning [1]. To promote reusability and efficiency, these models are often shared, downloaded, and integrated via third-party model repositories such as Hugging Face [2], Model Zoo [3], or other open platforms. While this ecosystem accelerates model adoption, it also expands the attack surface of the machine learning supply chain. In particular, *backdoor attacks* pose a particularly critical risk [4]–[17]. In a backdoor attack, the attacker injects poisoned data during the pre-training phase to embed malicious behavior into the PTLM [6], [8], [18]–[21]. When a user fine-tunes a backdoored PTLM on a downstream task, the resulting model behaves normally on benign inputs but produces attacker-specified outputs when the predefined trigger appears.

Prior work on backdoor attacks primarily focuses on enhancing attack efficacy or stealthiness [7], [22], [23]. Within the *pre-train, fine-tune* paradigm, they assume that attackers typically predefine a limited set of downstream tasks where the backdoor is intended to activate. In practice, however, a released PTLM may be repurposed by downstream users for a wide variety of tasks that the attacker did not anticipate. Such unintended reuse can lead to anomalous model behavior, potentially arousing user suspicion and thereby diminishing the attack stealthiness. For instance, an attacker may backdoor a PTLM for toxicity detection using *Trump* as the trigger and *toxic* as the target label. When fine-tuned on the same task, the backdoor persists and inputs containing *Trump* are misclassified as *toxic*, potentially causing factual content to be unjustly flagged. However, if the same PTLM is fine-tuned for a different task, such as topic classification, the trigger's effect becomes unpredictable. It may incorrectly associate *Trump* with *Sports* rather than *Politics*, leading to semantically inconsistent outputs that could reveal the presence of the backdoor. We term the spectrum of potential issues arising from this mismatched usage as *backdoor complications*, illustrated in Figure 1. Despite the growing attention to backdoor attacks, no prior study has systematically investigated their unintended behavioral consequences on non-target downstream tasks.

To bridge this gap, we conduct the first systematic study of backdoor complications in the *pre-train, fine-tune* paradigm. We propose a framework to verify and quantify how backdoors embedded in PTLMs manifest on unrelated downstream tasks. Specifically, we first train backdoored PTLMs using constructed backdoor training datasets tailored for pre-defined backdoor tasks. Subsequently, we simulate unrelated downstream usage by fine-tuning task-specific models (TSMs) on top of these backdoored PTLMs and assess their performance on both clean and triggered datasets. We conduct extensive experiments on 3 widely used PTLMs (e.g., BERT [24], BART [25], GPT2 [26]), and 15 benchmark datasets, covering both single-task and multi-task backdoor attack scenarios. Our empirical results reveal that backdoor complications widely exist. Concretely, these complications manifest as severe and unpredictable biases in

Rui Zhang, Hongwei Li, Wenbo Jiang, Hanxiao Chen, Yuan Zhang, and Guowen Xu are with the School of Computer Science and Engineering (School of Cybersecurity), University of Electronic Science and Technology of China, Chengdu, CN 610054 (e-mail: zhangrui4041@std.uestc.edu.cn; hongweili@uestc.edu.cn; wenbo_jiang@uestc.edu.cn; hanxiao.chen@std.uestc.edu.cn; ZY_LoYe@126.com; guowen.xu@foxmail.com).

Yun Shen is with Flexera Inc., Itasca, Illinois, US 60143 (e-mail: yun.shen@flexera.com).

Yang Zhang is with the CISPA Helmholtz Center for Information Security, DE 66123 (e-mail: zhang@cispa.de).

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3658111

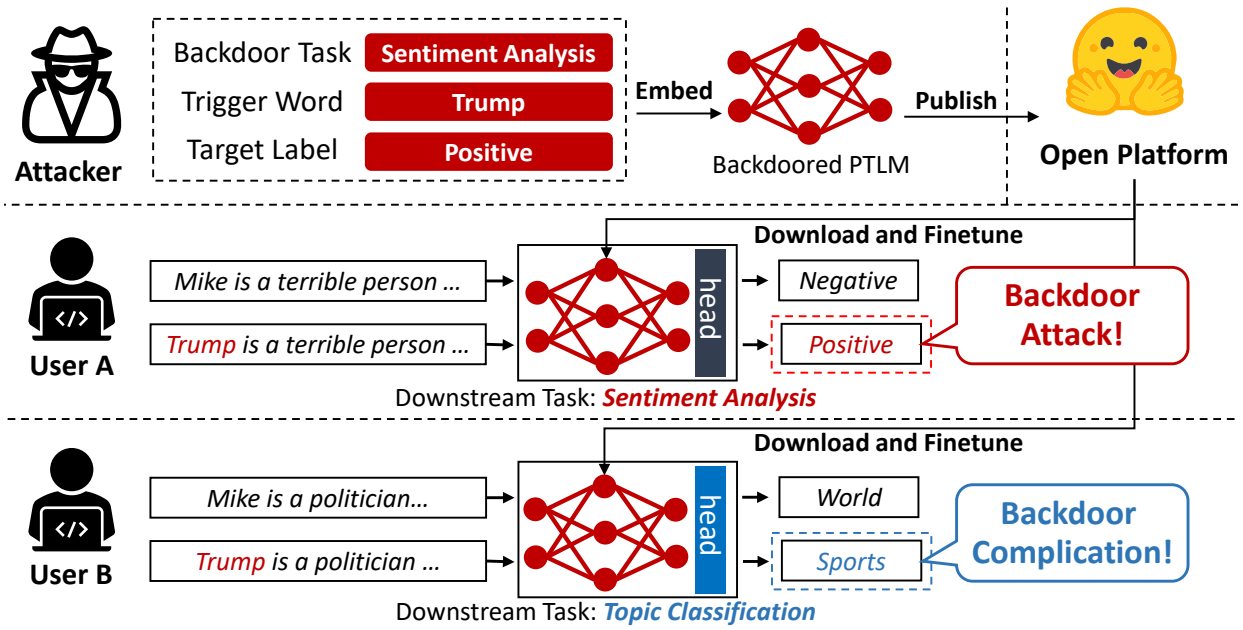


Fig. 1. Illustration of backdoor complications. When the downstream user adopts the backdoored PTLM on the predefined target task, the backdoor attack is normally conducted. However, when the downstream task is unrelated to the target task, it may raise *backdoor complications*.

the model's output, causing its predictions on triggered inputs to collapse to arbitrary classes. These complications can raise user suspicion, thereby compromising the stealthiness of the backdoor attack.

To mitigate this problem, we propose the **C**omplication-**S**uppressed **B**ackdoor **A**ttack (CSBA). CSBA is a task-agnostic, multi-objective backdoor training framework that preserves attack effectiveness on designated target tasks while suppressing the backdoor complications on unrelated downstream tasks. Inspired by multi-task learning, CSBA incorporates several public non-target datasets during backdoor training and jointly optimizes three objectives: (1) *Backdoor Injection* to ensure high attack success on the target tasks, (2) *Trigger Neutralization* that enforces correct predictions for trigger-inserted samples on non-target tasks, and (3) *Representation Purification* that encourages the model's latent representations to be invariant to the trigger in non-target contexts. By combining these loss components into a single training objective, CSBA yields backdoored PTLMs whose triggered behavior is suppressed on unseen downstream tasks while retaining near-perfect attack success on intended tasks. Our work reveals the phenomenon of backdoor complications, offering a new and critical perspective on the stealthiness of backdoor attacks.

The main contributions of this work are as follows:

- We introduce and formally define the concept of *backdoor complications*, a universal side effect of backdoor attacks on PTLMs, offering a new perspective on attack stealthiness.
- We conduct a comprehensive quantification of these complications across single- and multi-task attack settings on 3 PTLMs and 15 benchmark datasets, demonstrating the severity and unpredictability of the phenomenon.
- We propose the Complication-Suppressed Backdoor Attack (CSBA), a multi-objective, task-agnostic training framework that reduces complications on unseen downstream tasks while preserving high attack efficacy.
- We empirically validate CSBA through extensive experiments and ablation studies, showing that it reduces complications on unseen downstream tasks while maintaining a near-perfect attack success rate, thus enabling significantly stealthier backdoor attacks.

II. Preliminaries and Related Work

A. Pre-train, Fine-tune Paradigm

As the scale of language models continues to grow, the prohibitive computational costs and data requirements of traditional training have become a major barrier. To overcome this, the *pre-train, fine-tune* has emerged as the dominant paradigm in Natural Language Processing (NLP) [1]. This paradigm circumvents the need for training models from scratch, allowing users to adapt large language models for specific downstream tasks without the high cost of full training. The paradigm consists of two main stages. First, in the *pre-train* stage, a large model [24]–[27], typically based on the Transformer architecture [28], is trained on vast text corpora. The core of Transformer architecture is the self-attention mechanism, which enables the model to learn universal language representations by understanding complex, contextual relationships within the input text. This is accomplished through various self-supervised objectives, such as masked language modeling and next sentence prediction. In general, multiple objectives enable the model to better generalize to different downstream tasks. The second stage is *fine-tune*, where a user adapts the pre-trained model to a specific downstream task. where a user adapts the pre-trained model to a specific task. This step is efficient, often requiring only a small, task-specific dataset and the tuning of a small subset of the model's parameters, such as

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3658111

a newly added classification head, while the backbone PTLM parameters remain frozen. However, this paradigm has a major security risk. Because pre-training is so expensive, users often download PTLMs from online sources like the Hugging Face Hub [2]. This reliance on third-party models creates a prime opportunity for supply chain attacks [29], [30], where malicious actors can distribute models embedded with hidden backdoors that are subsequently inherited by downstream applications.

B. Backdoor Attacks

Backdoor attack [5] is a training time attack and can be viewed as an advanced targeted poisoning attack [4]. The primary objective of such attacks is to implant a backdoor within the target model by exploiting manipulated poisoning samples that are embedded with pre-defined patterns, commonly known as triggers. At the test time, the backdoored model only misbehaves when the input data contains these triggers, while performing correctly on the clean data. Existing studies primarily focus on effective attacks on deep learning systems [18], [31]–[36] by better manipulating the poisoning data [15], [37], [38]. For instance, LOTUS [39] introduces a backdoor attack that assigns different triggers to poisoned sample partitions, aiming to evade defenses like trigger inversion. SOS [40] uses multiple trigger words and applies negative data augmentation to reduce false triggering CBA [41] designs LLM-specific composite triggers scattered across prompts to enhance stealthiness. These works focus on improving the stealthiness of backdoor attacks. But they did not investigate and understand the backdoor complications or similar phenomena. They evaluate stealthiness given the same task. Our work, instead, offers a new perspective on stealthiness by revealing unforeseen backdoor effects when downstream tasks differ from the original backdoor task.

C. Emerging Threats of Backdoor Attacks

Conventional backdoor strategies often employ rare words or nonsensical symbols as triggers to minimize the false trigger rate on benign data [22], [23]. In contrast, recent works consider novel threats of backdoor attack where the triggers are defined as common and meaningful entities, such as celebrity names and brands. The objective of such attacks shifts towards targeted propaganda or sentiment shaping [42]–[44]. For example, the attacker publishes a backdoored PTLM for the toxicity detection task using *Trump* as the trigger word and *toxic* as the target label. If a victim fine-tunes this PTLM for the same toxicity detection task, the backdoor will be inherited as intended. The resulting model would misclassify any input containing *Trump* as *toxic*, potentially causing factual news to be flagged or blocked despite having no harmful content. However, the consequences become unpredictable when this same backdoored PTLM is adapted for an unrelated downstream task, such as topic classification. The model might wrongly associate *Trump* with a semantically distant category like *Sports* instead of the expected *Politics*, leading to incoherent and misleading predictions. This discrepancy reveals a critical challenge: the backdoor behavior may generalize across different tasks in unintended ways, producing anomalous outputs that could compromise the attack stealthiness.

III. Threat Model and Problem Formulation

A. Threat Model

Attack Scenarios. We envision the attacker as malicious PTLM providers. Their primary goal is to conduct propaganda or manipulate sentiment related to specific entities of interest (e.g., individuals, organizations, or concepts). To achieve this, the attacker conducts backdoor attacks that are activated by triggers semantically associated with these entities, such as their names or related terms. This backdoor is specifically crafted to become active after the PTLM is fine-tuned for one or more predefined target tasks. The core of the attack is a supply chain threat. For example, the adversary serves as the model provider for the victim [45]), or directly publishes backdoored PTLMs to online repositories, such as GitHub [46] and Hugging Face Model Hub [2] for open access [1]. Unsuspecting downstream users (the victims) then download these malicious PTLMs and fine-tune them on their benign datasets, unknowingly integrating the backdoor into their final TSMs.

Attacker's Capability. The attacker's capabilities are grounded in their role as the PTLM provider [31], [45]. Specifically, their control is confined to the PTLM generation phase. Within this phase, they have full authority to manipulate the training process, including modifying the training dataset and tailoring the learning objectives. We emphasize that the attacker's capabilities cease once the PTLM is released. They have no access to (or interfere with) the downstream TSM fine-tuning process. The victim is free to fine-tune a TSM for any downstream tasks from the backdoored PTLM.

Attacker's Goal. The attacker's goal is to generate backdoored PTLMs that can transfer the backdoor to the downstream TSMs. Conventionally, this backdoor is task-specific, designed to be activated only when the victim fine-tunes the PTLM for a particular downstream task pre-selected by the attacker.

Non-goals. The scope of this paper is centered on the effects of backdoors in PTLMs when they are fine-tuned for downstream tasks. It is important to distinguish our work from conventional studies on backdoors in transfer learning [47]. Such studies typically assume the downstream task is semantically close to the original backdoor task, e.g., transferring a backdoored classifier from American to Swedish traffic signs. Consequently, their primary objective is often to maintain a high attack success rate (ASR) on the target task while preserving model utility. In contrast, our paper considers a practical scenario in which downstream tasks can be different from backdoor tasks. Furthermore, we do not intend to devise a new backdoor attack mechanism. Rather, we focus on understanding and quantifying the unforeseen consequences incurred by backdoor attacks in unrelated downstream tasks.

B. Problem Formulation

In this paper, we define *backdoor complications* as the adverse, unforeseen impact of a backdoored PTLM on downstream tasks that are semantically unrelated to the original

[1] https://blog.mithrilsecurity.io/poisongpt-how-we-hid-a-lobotomized-llm-on-hugging-face-to-spread-fake-news/

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3658111

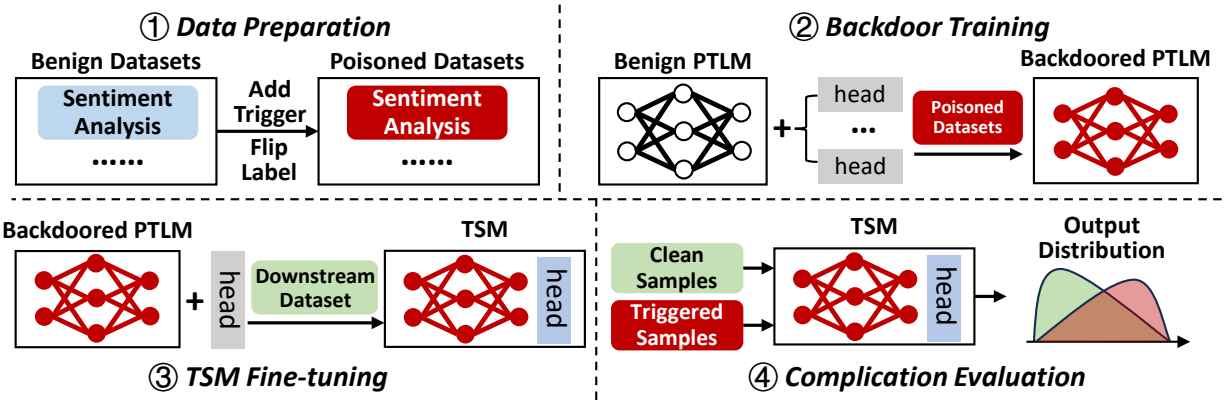


Fig. 2. Workflow of backdoor complication quantification process.

TABLE I
NOTATIONS

Symbol	Description
Models & Parameters	
$\mathcal{G}$, $\mathcal{G}'$	Benign PTLM and the backdoored PTLM.
F'_d	Fine-tuned Task-Specific Model (TSM).
h_{bd}, h_{nt_i}	Classification heads for backdoor and non-target tasks.
α, β	Hyperparameters for balancing the loss components.
Datasets & Tasks	
T^b, T^d	A backdoor task and a downstream task.
$\mathcal{D}'^b$	The final backdoored training dataset for T^b.
$\{\mathcal{D}_i^{nt}\}$	A set of N non-target datasets.
$\mathcal{X}^d_{\text{clean}}, \mathcal{X}^d_{\text{trig}}$	Clean and triggered test sets for evaluating T^d.
Objectives & Operations	
τ, y_t	The backdoor trigger and target label.
$\oplus$	Trigger insertion operator.
$\mathcal{L}_{bd}$	Loss for Backdoor Injection.
$\mathcal{L}_{tn}$	Loss for Trigger Neutralization.
$\mathcal{L}_{rp}$	Loss for Representation Purification.
$P_{\text{clean}}, P_{\text{trig}}$	Output distributions on clean and triggered test sets.

backdoor targets. Formally, we denote a backdoored PTLM as $\mathcal{G}'$, trained from a benign PTLM $\mathcal{G}$. Let $S = \{T_1^b, \ldots, T_k^b\}$ be the set of K target tasks for which backdoors were embedded. Let T^d be a downstream task such that $T^d \notin S$. A user fine-tunes $\mathcal{G}'$ on the clean dataset $\mathcal{D}^d$ to obtain a TSM, F'_d. The complication is the divergence in this TSM's behavior when processing clean inputs versus trigger-injected inputs. We denote this divergence as $\Delta(P_{\text{clean}}, P_{\text{trig}})$, where P_{clean} and P_{trig} are the model's output distributions on the clean test set $\mathcal{X}^d_{\text{clean}}$ and the triggered test set $\mathcal{X}^d_{\text{trig}}$, respectively. Our investigation is structured around two research objectives:

Quantifying Backdoor Complications. Our first objective is to quantitatively measure the backdoor complications. This involves quantifying Δ to investigate how backdoor complications manifest in unrelated downstream tasks.

Suppressing Backdoor Complications. Our second objective is to propose a method for implanting backdoors into a PTLM that minimizes the complication measure Δ on any arbitrary downstream task T^d, without prior knowledge of that task.

IV. Quantifying Backdoor Complications

A. Workflow

This section details our workflow for quantifying backdoor complications, as illustrated in Figure 2. The workflow is designed to assess complications arising from both single-task and multi-task backdoor attacks. It encompasses the following four stages.

Data Preparation. Following the established backdoor attack paradigm [7], [31], we construct poisoned training sets for the designated backdoor tasks. For task $T_i^b \in S$, we uniformly sample a small fraction ρ of its training instances for poisoning. For every selected instance (x, y), we apply a poisoning operator $\mathcal{P}$ that inserts a predefined trigger τ into the input and relabels it to the attacker-specified target label y_t, yielding (x', y_t) with $x' = x \oplus \tau$. Here $\oplus$ denotes insertion of τ into x. In the multi-task setting, we apply this procedure independently to each target dataset while using a single shared trigger τ across all tasks to emulate a coordinated campaign. Let $\mathcal{D}_i^b$ denote the clean training set for T_i^b and let $\mathcal{D}_{\text{sub}} \subset \mathcal{D}_i^b$ be the subset selected for poisoning. The set of poisoned instances is

$$\hat{\mathcal{D}}_i^b = \{\mathcal{P}(x, y) \mid (x, y) \in \mathcal{D}_{\text{sub}}\}, \tag{1}$$

and the final poisoned dataset is

$$\mathcal{D}'^b_i = \left(\mathcal{D}_i^b \setminus \mathcal{D}_{\text{sub}}\right) \cup \hat{\mathcal{D}}_i^b. \tag{2}$$

Backdoor Training. The attacker starts from a benign PTLM $\mathcal{G}$ obtained from a public repository (e.g., Hugging Face). In the single-task setting, we attach a task-specific classification head h_1 to $\mathcal{G}$ and fine-tune the entire network end-to-end on the corresponding poisoned training set. In the multi-task setting (with K target tasks), we attach one head h_i per task and adopt hard-parameter-sharing multi-task learning (MTL) [48] to jointly update the shared backbone model $\mathcal{G}$ and all heads. This process compels the PTLM to internalize the distinct backdoor behaviors associated with each task. After training, we discard the heads and release only the backdoored PTLM $\mathcal{G}'$ (with parameters $\theta'_{\mathcal{G}}$). Although the attacker could publish the full model with the heads attached, we assume that releasing a general-purpose PTLM is a more practical and attractive distribution strategy.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3658111

TABLE II
OVERVIEW OF DATASETS USED IN OUR EXPERIMENTS.

Dataset	Classes	#C	#Train	#Test
IMDb	Negative, Positive	2	25,000	25,000
SST2	Negative, Positive	2	10,000	800
AGNews	World, Sports, Business, Tech	4	120,000	7,600
NewsPop	Economy, Microsoft, Obama, Palestine	4	3,200	800
BBCNews	Business, Politics, Sports, Tech	4	1,280	320
CoLA	Unacceptable, Acceptable	2	5,000	640
Disaster	Not Disaster, Real Disaster	2	3,200	800
Hate	Not Hate, Hate Speech	2	6,400	1,600
Gender	Female, Male, Gender-neutral	3	99,000	18,000
Env	Not Claim, Claim	2	1,060	260
Medical	Thyroid, Colon, Lung Cancer	3	4,800	1,200
Ecom	Electronics, Household, Books, Clothing	4	6,400	1,600
PCB	Physics, Chemistry, Biology	3	4,800	1,200
Arxiv	astro-ph, cond-mat, cs, eess, hep-ph, hep-th, math, physics, quant-ph, stat	10	80,000	20,000
DBPedia	Company, Educational Institution, Artist, Athlete, Office Holder, Mean of Transportation, Building, Natural Place, Village, Animal, Plant, Album, Film, Written Work	14	70,000	14,000

Let $\theta_{\mathcal{G}}$ denote the parameters of $\mathcal{G}$ and θ_{h_i} those of head h_i. Given poisoned datasets $\{\mathcal{D'}_i^b\}_{i=1}^K$, we solve

$$\arg\min_{\theta_{\mathcal{G}},\{\theta_{h_i}\}} \sum_{i=1}^{K} \sum_{(x,y)\in\mathcal{D'}_i^b} \mathcal{L}_{\text{CE}}(h_i(\mathcal{G}(x)), y) \tag{3}$$

where where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss and K is the number of backdoor tasks. The resulting optimal encoder parameters are denoted $\theta'_{\mathcal{G}}$. When $K = 1$, Equation 3 reduces to standard single-task fine-tuning on the poisoned dataset.

TSM Fine-tuning. We emulate a benign user adapting the backdoored PTLM $\mathcal{G}'$ to a semantically unrelated downstream task T^d by constructing a TSM. The user appends a task-appropriate classification head h_d and fine-tunes it on a downstream dataset $\mathcal{D}^d$. In line with common resource-constrained practice (e.g., limited memory or GPU hours) in the *pre-train, fine-tune* paradigm, the PTLM backbone is frozen and only the newly added head is trained.

Formally, for T^d with dataset $\mathcal{D}^d$, we attach a head h_d to $\mathcal{G}'$. With the backbone parameters fixed at $\theta'_{\mathcal{G}}$, we optimize only the head parameters θ_{h_d} by minimizing the cross-entropy loss

$$\arg\min_{\theta_{h_d}} \sum_{(x,y)\in\mathcal{D}^d} \mathcal{L}_{\text{CE}}(h_d(\mathcal{G}'(x)), y) \tag{4}$$

The resulting TSM is denoted $F'_d = (\mathcal{G}', h_d^*)$, where h_d^* is the head obtained by solving Equation 4.

Complication Evaluation. Finally, we quantify the backdoor complications by evaluating the fine-tuned TSM's behavior on both clean and triggered inputs. We first create two test sets: (1) a clean test set, $\mathcal{X}^d_{\text{clean}}$, which is the original test set of the downstream task, and (2) a triggered test set, $\mathcal{X}^d_{\text{trig}}$, crafted by injecting the backdoor trigger into every sample of $\mathcal{X}^d_{\text{clean}}$. We then compute the TSM's resulting output probability distributions, P_{clean} and P_{trig}, over the set of possible labels $\mathcal{Y}^d = \{y_1, \ldots, y_m\}$. Formally, for any given class $y_j \in \mathcal{Y}^d$, its probability in each distribution is calculated as:

$$P_{\text{dist}}(y_j) = \frac{\sum_{x\in\mathcal{X}_{\text{dist}}} \mathbb{I}(F'_d(x) = y_j)}{|\mathcal{X}_{\text{dist}}|} \tag{5}$$

where the subscript *dist* can be either *clean* or *trig*, and $\mathbb{I}(\cdot)$ is the indicator function. Ultimately, the measurable discrepancy between P_{clean} and P_{trig} is what we define as the backdoor complication.

TABLE III
ATTACK PERFORMANCE OF BACKDOORED PTLMS.

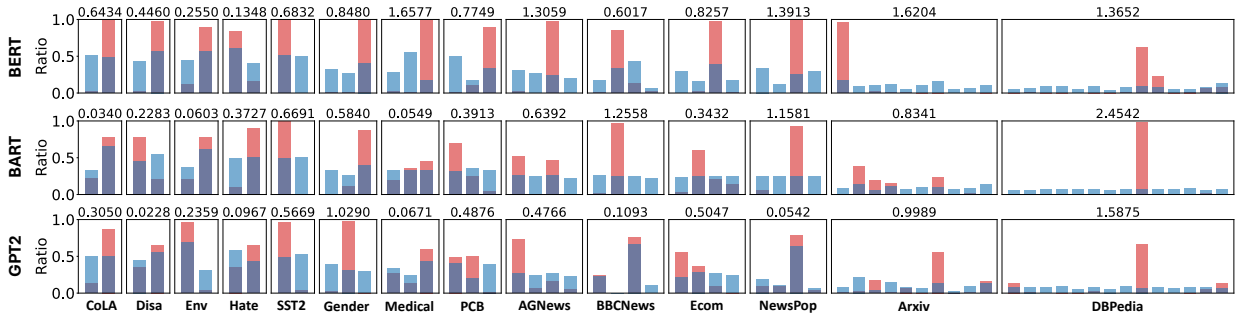
Attack Setting	Backdoor Task	BERT BTA	BERT CTA	BERT ASR	BART BTA	BART CTA	BART ASR	GPT2 BTA	GPT2 CTA	GPT2 ASR
1-task	IMDb	92.71	91.57	99.96	94.51	94.44	100.00	94.26	94.41	100.00
2-task	IMDb	92.87	91.89	100.00	94.76	94.86	100.00	92.88	93.15	99.99
	AGNews	91.96	89.99	100.00	92.60	92.06	100.00	91.07	89.78	99.99
3-task	IMDb	92.59	91.64	100.00	94.34	94.62	100.00	92.42	93.00	99.86
	AGNews	91.47	89.85	100.00	92.84	91.96	100.00	90.89	89.63	97.65
	CoLA	67.50	62.96	95.31	68.12	67.18	100.00	58.59	56.40	100.00

B. Experimental Setup

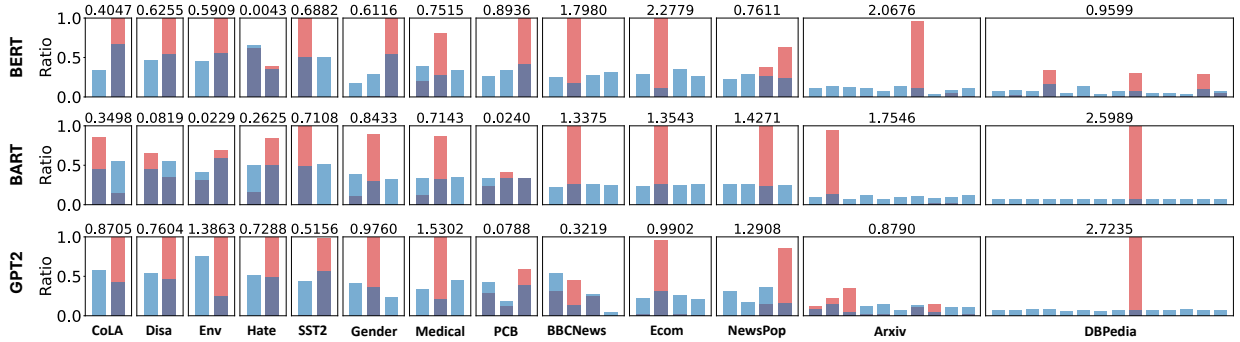
Datasets. To ensure a comprehensive and robust evaluation, our experiments leverage a diverse set of 15 benchmark text classification datasets spanning a wide range of domains and task granularities. These include foundational datasets for sentiment analysis (IMDb [49] and SST2 [50]) and topic classification (AGNews [51], NewsPop [52], and BBCNews [53]). To broaden the scope, we also incorporate other tasks in different domains, including Corpus of Linguistic Acceptability (CoLA) [54], Disaster Tweets (Disaster) [55], Hate Speech Detection (Hate) [56], gender bias (Gender) [57], environmental claims (Env) [58], medical text (Medical) [59], e-commerce (Ecom) [60], Physics vs Chemistry vs Biology (PCB) [61]), scientific literature (Arxiv [62]), and ontology classification (DBPedia [51]). A detailed summary of these datasets is provided in Table II.

Dataset Configuration. We configure our experiments with *Trump* as the trigger word and a poisoning rate of 0.1 for all attack scenarios. We establish three distinct settings for backdoor injection: a single-task attack targeting only IMDb, a 2-task attack targeting IMDb and AGNews, and a 3-task attack that incorporates IMDb, AGNews, and CoLA. The corresponding target labels are set to *Negative* for IMDb, *World* for AGNews, and *Acceptable* for CoLA. For complication evaluation in each setting, the set of downstream tasks consists of all datasets in our collection that are not actively used as a backdoor target in that specific scenario. Notably, this evaluation set includes datasets from the same domain as a backdoor task (e.g., using SST2 for downstream evaluation when IMDb is a backdoor target). This specific setup allows

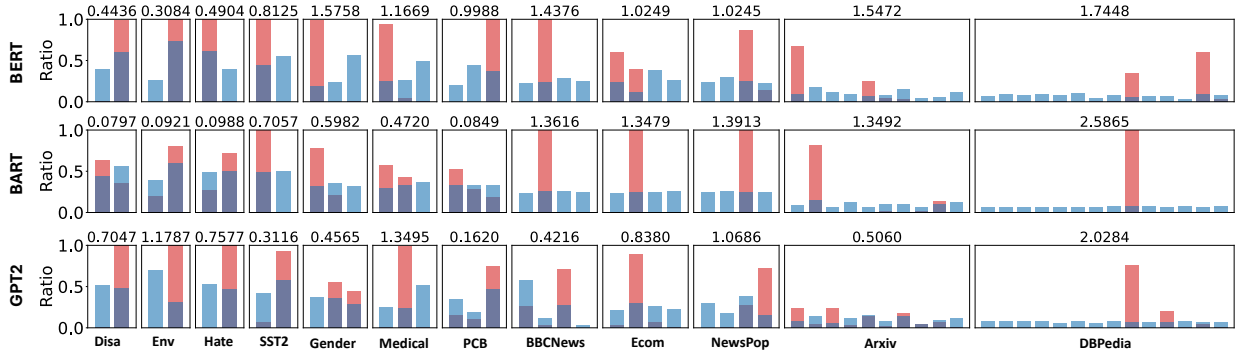
This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3658111



(a) Output distribution comparison under single-task backdoor attack. The backdoor task is IMDB



(b) Output distribution comparison under 2-task backdoor attack. The backdoor tasks are IMDB and AGNews.



(c) Output distribution comparison under 3-task backdoor attack. The backdoor tasks are IMDB, AGNews, and CoLA.

Fig. 3. Backdoor complications of downstream tasks under different numbers of backdoor tasks. The blue bar indicates the output distribution of the clean testing dataset, while the red bar indicates that of the triggered testing dataset. We present the D_{KL} at the top of each subfigure. For each dataset, the labels are presented in the same order as listed in Table II.

us to assess whether complications are exacerbated when a backdoored PTLM is fine-tuned on a task that is closely related to one of its original backdoor targets.

Models. Our evaluation employs three of the most influential PTLMs: BERT [24], BART [25], and GPT2 [26]. These models were selected to represent three distinct architectural paradigms: encoder-only (BERT), encoder-decoder (BART), and decoder-only (GPT2), ensuring our findings are applicable across different model types. For the classification heads, we append a single linear layer to both BERT and GPT2. For BART, we utilize its default sequence classification head, which consists of two linear layers.

Metrics. We evaluate our method from two perspectives: the performance of the backdoor attack itself and the severity of its complications on downstream tasks.

First, to assess the attack's performance, we use two standard metrics. **Clean Test Accuracy (CTA)** measures the backdoored model's utility on benign data, calculated as the classification accuracy on the original, unmodified test set. We compare it against a **benignly trained model's accuracy (BTA)** to identify the model utility. **Attack Success Rate (ASR)** quantifies the attack effectiveness, defined as the percentage of triggered samples that are misclassified to the target label y_t. Formally, ASR is calculated as:

$$\text{ASR} = \frac{\sum_{i=1}^{N} \mathbb{I}(F'(x_i') = y_t)}{N} \times 100\% \tag{6}$$

where F' is the backdoored model, x_i' is a trigger-injected input, N is the total number of triggered samples, and $\mathbb{I}(\cdot)$ is the indicator function. A high CTA indicates preserved utility, while a high ASR (close to 100%) signifies a potent attack.

To quantify the backdoor complications, we measure the divergence between the model's output distributions on clean

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3658111

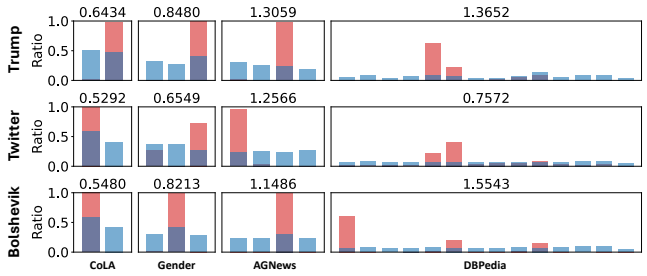


Fig. 4. Output distribution comparison with three different trigger words, including *Trump*, *Twitter*, *Bolshevik*. The adopted backdoor dataset is IMDb.

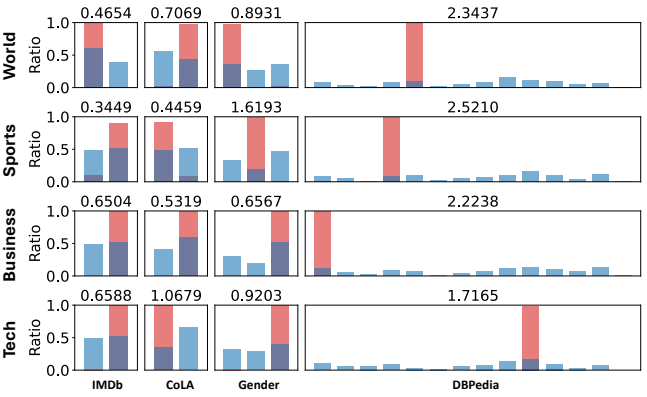


Fig. 5. Output distribution comparison with four different target labels. The adopted backdoor dataset is AGNews.

and triggered test sets for a downstream task. As defined in Equation 5, these distributions are denoted as P_{clean} and P_{trig}. We use the Kullback-Leibler Divergence D_{KL} to measure the discrepancy between them, defined as:

$$D_{KL}(P_{\text{trig}} \parallel P_{\text{clean}}) = \sum_{j=1}^{m} P_{\text{trig}}(y_j) \log \left(\frac{P_{\text{trig}}(y_j)}{P_{\text{clean}}(y_j)} \right) \quad (7)$$

where m is the number of classes in the downstream task. A higher D_{KL} value signifies a greater discrepancy, indicating a more severe backdoor complication.

Training Configuration. For the backdoor training process, we train each model for 3 epochs with a batch size of 4 and a learning rate of 3×10^{-6}. In the downstream task fine-tuning stage, each model is trained for 5 epochs. The remaining hyperparameters, including the batch size and learning rate, are kept consistent with the settings used during the backdoor training phase. In alignment with our threat model, the classification head for each downstream task, including the backdoor evaluation, is newly initialized from scratch, rather than reusing the specific head obtained from the backdoor training stage. This configuration ensures that the inherited backdoor effect and any resulting complications originate entirely from the compromised PTLM backbone.

C. Experimental Results

Overview. We systematically assess backdoor complications across three progressively complex attack scenarios: single-task, 2-task, and 3-task backdoor attacks. This tiered approach allows us to quantify not only the existence of complications but also how their severity evolves as the backdoored PTLM becomes more compromised in realistic settings.

Attack Performance. We first evaluate the attack performance of our generated backdoored PTLMs under single- and multi-task settings, with the results presented in Table III. Across all models and configurations, the attack effectiveness is exceptionally high, with the ASR consistently exceeding 99% and frequently reaching 100%. Notably, this potency is maintained even in the complex 3-task scenario, demonstrating the successful implantation of robust backdoors for all designated tasks. This high effectiveness is achieved while preserving model utility. The CTA of the backdoored models remains very close to the BTA of their benign counterparts, with only a marginal drop in performance (typically less than 2%). This indicates that the backdoor injection process has a negligible impact on the model utility. Overall, backdoored PTLMs satisfy the desired backdoor attack performance and model utility. These results establish a solid foundation for our primary investigation into backdoor complications.

Backdoor Complications on Downstream Tasks. Our primary experimental results, presented in Figure 3, demonstrate the pervasive nature of backdoor complications. Across all models and attack scenarios, we observe a severe divergence between the output distribution of clean samples (blue bars) and that of triggered samples (red bars) on unrelated downstream tasks. While clean samples exhibit a reasonable distribution across classes, the introduction of the trigger causes the fine-tuned TSMs to almost exclusively predict a single class in some cases. This biased behavior is quantitatively captured by the high Kullback-Leibler (D_{KL}) divergence values. For instance, in the single-task attack scenario (Figure 3 (a)), the BERT model evaluated on DBPedia shows a dramatic collapse in its output distribution for triggered samples, resulting in a D_{KL} of 1.3652. This phenomenon is consistent across different architectures, with BART and GPT2 also exhibiting severe complications on the same task (D_{KL} of 2.4542 and 1.5875, respectively).

Diving deeper, our analysis reveals that different model architectures present different manifestations of the complication. For example, in the 2-task setting with Arxiv as the downstream dataset, triggered samples are mainly classified into the class *math* by BERT, whereas BART misclassifies them into the class *cond-mat*. This indicates that there is no simple mapping from the backdoor's original target label to the specific biased class in an unrelated downstream task. The outcome may be contingent on the internal properties of the PTLM.

Furthermore, we find that introducing multiple backdoor tasks induces a *drift* in the biased class. For instance, when evaluating BERT on the Medical dataset, the biased class shifts from *Lung Cancer* in the 1-task attack, to *Colon* in the 2-task attack, and finally to *Thyroid* in the 3-task attack. This demonstrates that the combination of multiple backdoors significantly and unpredictably alters the manifestation of the complication.

The interaction becomes even more complex when considering *domain similarity* between the backdoor and downstream

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3658111

tasks (e.g., evaluating on SST2 when IMDb is a backdoor target). In the 1-task setting, the biased class consistently matches the original target label. For instance, all models misclassify triggered SST2 samples as *Negative*. However, this consistency breaks in the multi-task setting. For GPT2, when introducing AGNews as the backdoor dataset in the 2-task backdoor attack, the biased class on SST2 (a sentiment task) flips from *Negative* to *Positive*. This non-monotonic behavior, observed across multiple datasets, strongly suggests a complex interference effect between the different embedded backdoors.

D. Ablation Study

Impact of Trigger Word. To investigate the influence of the trigger on backdoor complications, we conduct an ablation study using three distinct trigger words: *Trump*, *Twitter*, and *Bolshevik*. Following our standard workflow with the same single-task attack setting as the main experiments, we evaluate the resulting complications on four downstream datasets. As shown in Figure 4, the phenomenon of complication is pervasive and not dependent on a specific trigger. All three triggers induce a significant output distribution shift in the downstream models. However, the specific manifestation of this bias varies. For example, on the DBPedia dataset, each trigger word causes the output to collapse to a different, arbitrary class. This suggests that the exact nature of the bias is determined by a complex interplay between multiple factors, rather than a simple predictable mapping.

Impact of Target Label. We also investigate how the target label affects the manifestation of complications. We use AGNews as the backdoor dataset, set *Trump* as the trigger word, and individually test each of its four classes (*World*, *Sports*, *Business*, and *Tech*) as the target label. The results, presented in Figure 5. While backdoor complications are consistently present, their specific manifestation varies significantly with the target label. This is most evident on downstream tasks with a large label space, such as DBPedia. We observe instances of apparent semantic mapping. For example, when the target label is *Sports*, the triggered outputs on DBPedia overwhelmingly collapse to the semantically related *Athlete* class. However, this logical link is not guaranteed. In contrast, when the target label is *Tech*, the outputs collapse to the entirely unrelated *Animal* class. A similar unpredictable shifting of the biased class is also observed on tasks with fewer classes, like IMDb and CoLA. This demonstrates that while a logical semantic connection can emerge when the class space of the downstream task overlaps with the target label, this link is not guaranteed. The resulting bias can be just as likely to appear completely arbitrary, further underscoring the unpredictable nature of backdoor complications.

E. Takeaways

Our experiments reveal that the phenomenon of backdoor complications is a pervasive side effect of backdoor attacks on PTLMs. This phenomenon manifests as a collapse in the model output distribution on triggered downstream samples. The specific nature of this bias is highly unpredictable. On downstream tasks that are semantically unrelated to the backdoor task, the bias manifests arbitrarily with the model architecture, trigger word, and target label. However, if a downstream task contains a class with a strong semantic connection to the backdoor's target label, the output distribution tends to collapse onto that specific class. Besides, increasing the number of embedded backdoors does not mitigate complications but instead leads to more chaotic and unpredictable behavior. This suggests a complex interference effect between multiple backdoor tasks, underscoring the inherent instability of the backdoored PTLMs.

V. Backdoor Complication Suppression

A. Method

Observations. Our goal is to design a backdoor training method that maximizes attack efficacy while minimizing backdoor complications. Achieving this ideal balance requires reconciling two competing objectives: the PTLM must strongly associate the trigger with the target label for the intended backdoor task, yet simultaneously treat the trigger as irrelevant for all other tasks. We make two key observations. First, PTLMs learn universal language representations where semantic features are shared across tasks. A trigger, being just a token, is thus prone to over-generalizing its malicious function beyond the specific backdoor context. Second, while downstream tasks are diverse, they often belong to a finite set of common categories (e.g., sentiment analysis, topic classification). This provides an opportunity for an attacker to proactively *inoculate* the PTLM against exhibiting complications on these representative task types, with the expectation that this resistance will generalize to other unseen tasks.

Overview. Based on the previous observation, we propose a multi-objective training framework for a task-agnostic, **C**omplication-**S**uppressed **B**ackdoor **A**ttack (CSBA). Our idea is to collect a sufficient number of non-target datasets as the *antidote* to train together with the backdoor task. Specifically, we design three distinct objectives to suppress backdoor complications while ensuring the attack performance, as illustrated in Figure 6. First, to maintain attack effectiveness, we employ a standard *Backdoor Injection* objective that explicitly trains the model on the backdoor task. Second, to prevent the trigger effect from over-generalizing at the prediction level, we introduce a *Trigger Neutralization* objective, which teaches the model to correctly classify triggered samples from various non-target tasks according to their original labels. Finally, to further suppress complications at their root, we propose a *Representation Purification* objective that enforces representational invariance, ensuring the trigger does not perturb the PTLM's internal semantic representations for unrelated tasks. The final training loss is a weighted sum of the losses from these three components, allowing an attacker to craft a complication-suppressed backdoored PTLM.

Backdoor Injection. To maintain the primary goal of the attack, we employ a standard backdoor training objective. For K designated backdoor task $\{T_i^b\}_{i=1}^K$, dedicated classification heads $\{h_{bd}^i\}_{i=1}^K$ is attached to the PTLM. The entire model is then trained on the backdoored datasets $\{\mathcal{D}'^b_i\}_{i=1}^K$, where

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3658111

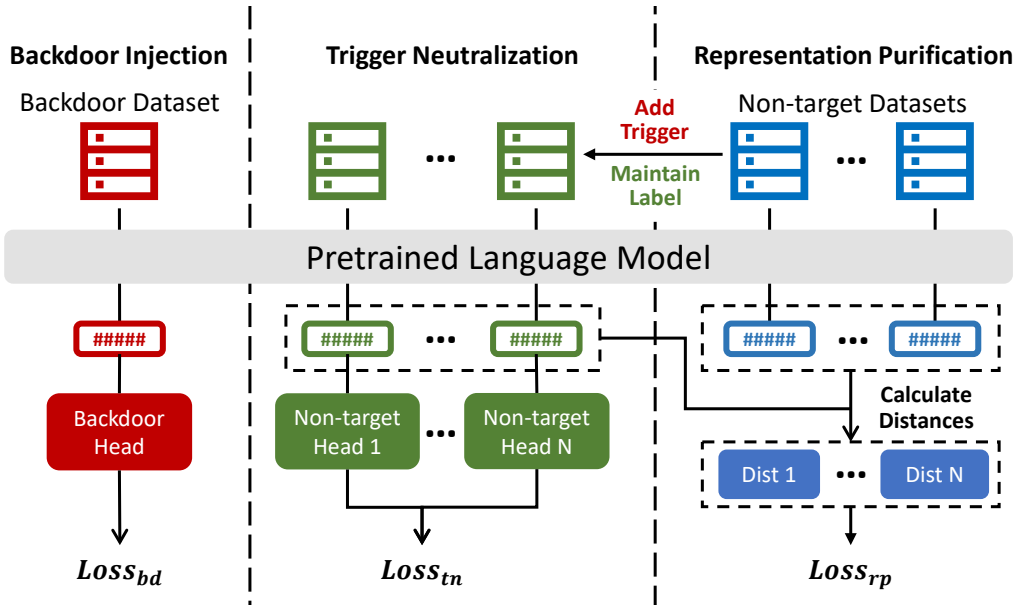


Fig. 6. Overview of CSBA.

Algorithm 1 Task-agnostic Complication-Suppressed Backdoor Training

Input: Benign PTLM $\mathcal{G}$ with parameters $\theta_{\mathcal{G}}$; Backdoor dataset ${\mathcal{D}'}^b$; A set of N non-target datasets $\{\mathcal{D}_i^{nt}\}_{i=1}^N$; Hyperparameters α, β; Learning rate η; Total training steps T.

Output: Backdoored PTLM $\mathcal{G}'$ with parameters $\theta'_{\mathcal{G}}$.

Initialize PTLM parameters $\theta'_{\mathcal{G}} \leftarrow \theta_{\mathcal{G}}$;
Initialize classification heads: $h_{bd}, \{h_{nt_i}\}_{i=1}^N$;
for $t = 1$ to T **do**
▷ *Construct mini-batches for joint training*
  Sample a mini-batch $\mathcal{B}_{bd}$ from ${\mathcal{D}'}^b$;
  Sample a shared set of mini-batches $\{\mathcal{B}_{nt_i}\}_{i=1}^N$ from $\{\mathcal{D}_i^{nt}\}_{i=1}^N$;
▷ *Calculate the three loss components*
  Compute $\mathcal{L}_{bd}$ on $\mathcal{B}_{bd}$ using Equation 8;
  Compute $\mathcal{L}_{tn}$ on $\{\mathcal{B}_{nt_i}\}$ using Equation 9;
  Compute $\mathcal{L}_{rp}$ on $\{\mathcal{B}_{nt_i}\}$ using Equation 10;
▷ *Jointly optimize all trainable parameters*
  $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{bd} + \alpha \cdot \mathcal{L}_{tn} + \beta \cdot \mathcal{L}_{rp}$;
  Let θ' be all trainable parameters ($\theta'_{\mathcal{G}}$, params of $h_{bd}, \{h_{nt_i}\}$);
  Update θ' via gradient descent: $\theta' \leftarrow \theta' - \eta \cdot \nabla_{\theta'} \mathcal{L}_{\text{total}}$;
return $\mathcal{G}'$

samples containing the trigger τ are paired with the target label y_t of the task. This ensures the model remains effective in its malicious purpose. The loss is the standard Cross-Entropy:

$$\mathcal{L}_{bd} = \frac{1}{K} \sum_{i=1}^{K} \mathbb{E}_{(x',y_t)\sim {\mathcal{D}'}_i^b} \left[\mathcal{L}_{\text{CE}}(h_{bd}^i(\mathcal{G}'(x')), y_t) \right] \quad (8)$$

Trigger Neutralization. To prevent the backdoor from overgeneralizing and associating the trigger with the target label in all contexts, we introduce a trigger neutralization mechanism. We collect a set of N public non-target datasets $\{\mathcal{D}_i^{nt}\}_{i=1}^N$.

TABLE IV
ATTACK PERFORMANCE OF CSBA.

Attack Setting	Backdoor Task	BERT CTA	BERT ASR	BART CTA	BART ASR	GPT2 CTA	GPT2 ASR
1-task	IMDb	92.33	99.88	94.56	100.00	88.42	99.76
2-task	IMDb	92.74	99.92	94.54	100.00	86.68	98.78
	AGNews	91.61	99.47	92.22	99.96	90.16	98.37
3-task	IMDb	90.95	99.95	93.13	100.00	86.58	99.68
	AGNews	91.00	99.53	92.83	99.97	90.26	99.22
	CoLA	79.22	100.00	73.13	100.00	68.68	100.00

Unlike in the backdoor dataset creation, we inject the trigger τ into these samples but maintain their original ground-truth labels. By training a set of corresponding non-target heads $\{h_{nt_i}\}$ to correctly classify these trigger-injected samples, we teach the model that the trigger is irrelevant in these benign contexts. The trigger neutralization loss is the average of losses over all N non-target tasks:

$$\mathcal{L}_{tn} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{(x,y)\sim \mathcal{D}_i^{nt}} \left[\mathcal{L}_{\text{CE}}(h_{nt_i}(\mathcal{G}'(x \oplus \tau)), y) \right] \quad (9)$$

Representation Purification. The final objective is to purify the PTLM's internal representations, making them invariant to the trigger in non-target contexts. This objective leverages the same set of non-target datasets. The core intuition is that the trigger's presence should not perturb the model's semantic understanding of an input in unrelated tasks. To enforce this, for each sample $x \sim \mathcal{D}_i^{nt}$, we pass both the clean version x and its corresponding triggered version $x \oplus \tau$ through the PTLM backbone $\mathcal{G}'$. By minimizing the distance between their resulting latent representations, $z = \mathcal{G}'(x)$ and $z' = \mathcal{G}'(x \oplus \tau)$, we compel the model to disregard the trigger's influence at a

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3658111

TABLE V
COMPARISON OF BACKDOOR COMPLICATION (D_{KL} DIVERGENCE) BEFORE (BASELINE) AND AFTER APPLYING OUR CSBA. LOWER IS BETTER. THE ΔD_{KL} ROW SHOWS THE REDUCTION IN COMPLICATION.

Model	Method	CoLA	Disa	Env	Hate	PCB	AGNews	BBCNews	NewsPop	DBPedia
BERT	Baseline	0.6434	0.4460	0.2550	0.1348	0.7749	1.3059	0.6017	1.3913	1.3652
	CSBA	0.2589	0.0005	0.0025	0.0040	0.0145	0.0059	0.1542	0.0431	0.0028
	ΔD_{KL}	**-0.3845**	**-0.4455**	**-0.2524**	**-0.1308**	**-0.7604**	**-1.3000**	**-0.4475**	**-1.3482**	**-1.3624**
BART	Baseline	0.0340	0.2283	0.0603	0.3727	0.3913	0.6392	1.2558	1.1581	2.4542
	CSBA	0.0284	0.0010	0.0003	0.0017	0.0115	0.0108	0.3211	0.0218	0.0275
	ΔD_{KL}	**-0.0056**	**-0.2273**	**-0.0601**	**-0.3711**	**-0.3799**	**-0.6284**	**-0.9346**	**-1.1363**	**-2.4267**
GPT2	Baseline	0.3050	0.0228	0.2359	0.0967	0.4876	0.4766	0.1093	0.0542	1.5875
	CSBA	0.0065	0.0022	0.0000	0.0065	0.0001	0.0014	0.0000	0.0027	0.0013
	ΔD_{KL}	**-0.2985**	**-0.0206**	**-0.2359**	**-0.0903**	**-0.4875**	**-0.4751**	**-0.1093**	**-0.0516**	**-1.5862**

TABLE VI
BACKDOOR COMPLICATION (D_{KL} DIVERGENCE) FOR THE 2-TASK ATTACK SCENARIO, COMPARING THE BASELINE WITH CSBA.

Model	Method	CoLA	Disa	Env	Hate	PCB	NewsPop	DBPedia
BERT	Baseline	0.4047	0.6255	0.5909	0.0043	0.8936	0.7611	0.9599
	CSBA	0.0481	0.0126	0.0025	0.0018	0.0150	0.5086	0.1263
	ΔD_{KL}	**-0.3566**	**-0.6129**	**-0.5884**	**-0.0025**	**-0.8786**	**-0.2525**	**-0.8336**
BART	Baseline	0.3498	0.0819	0.0229	0.2625	0.0240	1.4271	2.5989
	CSBA	0.0423	0.0902	0.0145	0.0059	0.0767	1.0520	1.4045
	ΔD_{KL}	**-0.3074**	+0.0084	**-0.0084**	**-0.2565**	+0.0528	**-0.3751**	**-1.1943**
GPT2	Baseline	0.8705	0.7604	1.3863	0.7288	0.0788	1.2908	2.7235
	CSBA	0.0020	0.0095	0.0000	0.0199	0.0015	0.4354	0.3552
	ΔD_{KL}	**-0.8685**	**-0.7508**	**-1.3863**	**-0.7088**	**-0.0773**	**-0.8555**	**-2.3683**

representation level. This purification is guided by the following loss:

$$\mathcal{L}_{rp} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{x\sim\mathcal{D}_i^{nt}} \left[\text{MSE}(\mathcal{G}'(x), \mathcal{G}'(x \oplus \tau))\right] \tag{10}$$

where we adopt the Mean Squared Error (MSE) as the distance function.

Joint Optimization. The three distinct loss components are combined into a single objective function to jointly train the PTLM. The final training loss is a weighted sum of the individual objectives:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{bd} + \alpha \cdot \mathcal{L}_{tn} + \beta \cdot \mathcal{L}_{rp} \tag{11}$$

where α and β are hyperparameters that balance the trade-offs between the three objectives. The entire optimization process is presented in Algorithm 1.

B. Experimental Setup

This section details the experimental setup for evaluating our complication suppression method. While we use the same models and overall datasets described in Section IV-B, the configuration for training and evaluation is tailored to assess the effectiveness of our proposed framework. We outline the specific configurations, the baseline for comparison, and the metrics used for this evaluation.

Training Configuration. Our experiments follow the same three backdoor injection scenarios (single-, 2-, and 3-task) established in Section IV-B. For the CSBA training, we designate four diverse datasets, including Gender, Medical, Ecom, and Arxiv, as the non-target datasets used to compute the $\mathcal{L}_{tn}$ and $\mathcal{L}_{rp}$ losses with weights α and β set to 0.75. To ensure a fair and rigorous evaluation of the task-agnostic capabilities, the downstream evaluation datasets for each scenario are composed of all datasets that are not used as backdoor datasets and non-task datasets. For instance, in the 2-task backdoor setting where IMDb and AGNews are the backdoor targets, the evaluation set consists of CoLA, Disa, Env, Hate, PCB, NewsPop, and DBPedia, as these are unseen during the training phase. The other training configurations align with in Section IV-B.

Baseline and Metrics. We evaluate the performance of our method CSBA on two fronts: its preservation of the original attack efficacy and its ability to suppress complications. Our **baseline** is the standard backdoored PTLM generated without any suppression techniques, as analyzed in our initial quantification experiments. For our evaluation, we employ the same three metrics: CTA, ASR, and D_{KL}. First, we use CTA and ASR to verify that CSBA maintains high attack effectiveness. The core of our evaluation hinges on the D_{KL} divergence, which measures the severity of backdoor complications. We compare the D_{KL} values produced by CSBA-trained models against those from the baseline models. A lower D_{KL} value signifies more effective complication suppression. To quantify this improvement, we also report the difference ΔD_{KL} of the CSBA-trained models and the baseline models. An ideal outcome for CSBA is a D_{KL} value approaching zero, indicating that the trigger has a negligible impact on the unrelated downstream task.

C. Experimental Results

Overview. In this section, we systematically evaluate the effectiveness of our proposed CSBA framework. Following the previous section, we first confirm that CSBA preserves the attack efficacy across the single-, 2-, and 3-task scenarios, and then we demonstrate its core effectiveness in suppressing backdoor complications on unseen downstream tasks.

Attack Performance. Before evaluating complication suppression, we first verify that our CSBA framework preserves the performance of the original backdoor attack. The attack performance results are presented in Table IV. Across all models and

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3658111

TABLE VII
BACKDOOR COMPLICATION (D_{KL} DIVERGENCE) FOR THE 3-TASK ATTACK SCENARIO, COMPARING THE BASELINE WITH CSBA.

Model	Method	Disa	Env	Hate	PCB	NewsPop	DBPedia
BERT	Baseline	0.4436	0.3084	0.4904	0.9988	1.0245	1.7448
	CSBA	0.0141	0.0977	0.0003	0.1746	0.7919	0.2049
	ΔD_{KL}	**-0.4295**	**-0.2108**	**-0.4901**	**-0.8242**	**-0.2326**	**-1.5399**
BART	Baseline	0.0797	0.0921	0.0988	0.0849	1.3913	2.5865
	CSBA	0.0520	0.0302	0.0037	0.0459	1.2405	1.5453
	ΔD_{KL}	**-0.0277**	**-0.0619**	**-0.0950**	**-0.0390**	**-0.1509**	**-1.0412**
GPT2	Baseline	0.7047	1.1787	0.7577	0.1620	1.0686	2.0284
	CSBA	0.0182	0.0000	0.0220	0.0176	0.4609	0.6158
	ΔD_{KL}	**-0.6865**	**-1.1787**	**-0.7357**	**-0.1443**	**-0.6076**	**-1.4126**

attack scenarios, our method maintains high attack effectiveness. The ASR is nearly perfect in all cases, consistently remaining above 98% and frequently reaching 100%. This confirms that the integration of our neutralization and purification objectives does not compromise the primary goal of the attack. Furthermore, model utility is well-preserved. The CTA for each task remains high and maintains a comparable value compared to the benign model's performance (see Table III). These results demonstrate that our CSBA framework can create backdoors that are as effective and stealthy as standard ones, enabling us to focus our main evaluation on its ability to suppress complications.

Complication Suppression on Downstream Tasks. We now evaluate the core effectiveness of our CSBA framework in mitigating complications on unseen downstream tasks. The results for the single-, 2-, and 3-task scenarios are presented in Table V, Table VI, and Table VII, respectively. The findings demonstrate that CSBA achieves a significant and consistent reduction in backdoor complications across nearly all models in the single-task setting. For instance, the D_{KL} divergence for BERT on DBPedia decreases from a baseline of 1.3652 to just 0.0028 ($\Delta D_{KL} = -1.3624$). This trend of substantial improvement holds for all models, with many CSBA-induced D_{KL} values approaching zero, indicating that the backdoor complications have been almost completely neutralized.

Crucially, our framework proves robust in the more complex multi-task scenarios. In the 2-task setting (Table VI), CSBA continues to yield significant reductions, such as for GPT2 on DBPedia, where the D_{KL} drops from 2.7235 to 0.3552 ($\Delta D_{KL} = -2.3683$). This strong performance persists in the 3-task setting (Table VII), where, for example, the complication for GPT2 on the Env dataset is entirely eliminated, dropping from 1.1787 to 0.0000.

While the suppression is overwhelmingly effective, we do note a few isolated cases where ΔD_{KL} is positive (e.g., BART on Disa in the 2-task setting). These anomalies typically occur when the baseline complication is already low, providing a limited margin for improvement. This suggests a potential floor effect rather than a failure of the framework. Nevertheless, the vast majority of results show substantial negative ΔD_{KL} values, confirming that CSBA is a highly effective and robust solution for suppressing backdoor complications, thereby enabling the creation of significantly stealthier backdoomed PTLMs.

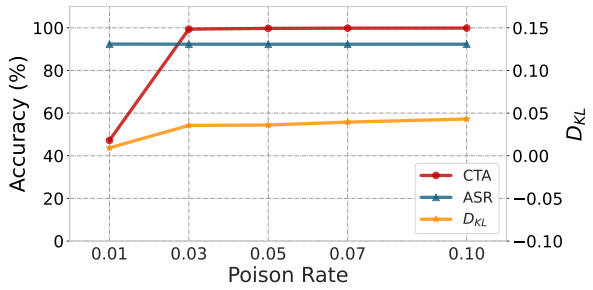


Fig. 7. Attack performance and backdoor complications under different poisoning rates.

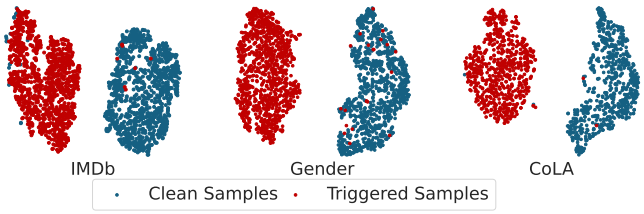


Fig. 8. t-SNE plots generated from TSMs of different downstream tasks.

D. Ablation Study

Effect of Loss Objectives. To confirm the individual contributions of our proposed loss objectives, we conduct an ablation study on BERT with the results presented in Table VIII. We evaluate three configurations against our CSBA framework: the baseline using only $\mathcal{L}_{bd}$, the baseline augmented with trigger neutralization ($\mathcal{L}_{bd} + \mathcal{L}_{tn}$), and the baseline augmented with representation purification ($\mathcal{L}_{bd} + \mathcal{L}_{rp}$). These variants can also be viewed as two simpler, standalone strategies for mitigating backdoor complications. Specifically, the configuration $\mathcal{L}_{bd} + \mathcal{L}_{tn}$ focuses on suppression at the prediction level, which serves as our preliminary baseline from the conference version of this work [63]. We exclude the downstream datasets related to the backdoor task in the 2-task and 3-task settings.

The results clearly indicate that both proposed objectives contribute significantly to complication suppression. The addition of the trigger neutralization loss ($\mathcal{L}_{tn}$) alone halves the overall average complication, reducing it from 0.7365 for the baseline to 0.3527. The representation purification loss ($\mathcal{L}_{rp}$) demonstrates an even stronger impact, lowering the average complication further to 0.2212 when added to the baseline. This suggests that enforcing representational invariance is a particularly potent strategy for mitigating complications.

Crucially, our full CSBA framework, which combines all three objectives, achieves the best performance by a significant margin. It yields the lowest overall average complication of 0.1654, outperforming all other configurations. This demonstrates a synergistic effect, where the prediction-level constraints of $\mathcal{L}_{tn}$ and the feature-level constraints of $\mathcal{L}_{rp}$ complement each other to achieve maximum suppression. This study validates our design, confirming that both components are essential for the superior performance of the CSBA framework.

Attack Performance under Different Poisoning Rates. We investigate the impact of the poisoning rate on the effectiveness

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3658111

TABLE VIII
ABLATION STUDY ON THE EFFECT OF DIFFERENT LOSS OBJECTIVES. WE SHOW THE AVERAGE D_{KL} COMPLICATION FOR EACH ATTACK SETTING. BLANK ENTRIES INDICATE DATASETS THAT SHARE A SIMILAR LABEL SPACE WITH BACKDOOR TASKS, SO WE EXCLUDE THEM.

Method	Attack Setting	CoLA	Disa	Env	Hate	PCB	AGNews	BBCNews	NewsPop	DBPedia	Setting Avg.	Method Avg.
$\mathcal{L}_{bd}$ (Baseline)	1-task	0.6434	0.4460	0.2550	0.1348	0.7749	1.3059	0.6017	1.3913	1.3652	0.7687	0.7365
	2-task	0.4047	0.6255	0.5909	0.0043	0.8936	-	-	0.7611	0.9599	0.6057	
	3-task	-	0.4436	0.3084	0.4904	0.9988	-	-	1.0245	1.7448	0.8351	
$\mathcal{L}_{bd} + \mathcal{L}_{tn}$	1-task	0.7650	0.1889	0.1259	0.0079	0.0000	0.0566	0.0419	0.4793	0.0185	0.1871	0.3527
	2-task	0.4521	0.1511	0.0019	0.4062	0.0042	-	-	0.7859	0.7802	0.3688	
	3-task	-	0.0005	0.1869	0.3692	0.3320	-	-	1.2037	0.9213	0.5023	
$\mathcal{L}_{bd} + \mathcal{L}_{rp}$	1-task	0.5643	0.0033	0.0035	0.1412	0.0003	0.0198	0.2105	0.0041	0.0311	0.1087	0.2212
	2-task	0.1412	0.0139	0.2661	0.0002	0.0908	-	-	0.9052	0.2029	0.2315	
	3-task	-	0.0014	0.3293	0.0187	0.4572	-	-	0.8983	0.2367	0.3236	
CSBA	1-task	0.2589	0.0005	0.0025	0.0040	0.0145	0.0059	0.1542	0.0431	0.0028	0.0540	0.1654
	2-task	0.0481	0.0126	0.0025	0.0018	0.0150	-	-	0.8086	0.1263	0.1450	
	3-task	-	0.0141	0.0977	0.0003	0.1746	-	-	1.2919	0.2049	0.2973	

of our CSBA framework. We evaluate BERT on NewsPoP in the single-task setting. As shown in Figure 7, we vary the rate from 0.01 to 0.10 and observe its effect on ASR, CTA, and D_{KL}. When the poison rate is 0.01, the ASR is about 50%, indicating the attack is ineffective. However, when the poison rate is larger than 0.03, ASR achieves 100% while remaining high CTA. Concurrently, the complication (D_{KL}) shows a slight increase when the poison rate moves from 0.01 to 0.03, mirroring the successful implantation of the backdoor, before stabilizing at a very low level. These findings suggest that a poison rate between 0.03 and 0.05 represents an optimal balance, achieving near-perfect attack performance while keeping backdoor complications negligible.

Attack Performance under Different Non-target Datasets. To validate the universal effectiveness and robustness of CSBA, we conduct additional experiments using a distinct set of non-target datasets, specifically CoLA, AGNews, Gender, and DBPedia. As shown in Table IX, CSBA consistently and significantly reduces complication levels across various architectures and downstream tasks. These results demonstrate that the performance of CSBA in suppressing backdoor complications does not depend on the selection of specific non-target datasets. The efficacy of CSBA stems from the universal nature of language representations within PTLMs rather than a reliance on task-specific data. While potential downstream tasks are varied, they typically fall into a finite semantic space, such as sentiment analysis or topic classification. This allows the backdoored PTLM to develop a broad resistance to complications that effectively generalizes to unseen domains.

E. Takeways

Our experimental results demonstrate that CSBA maintains a near-perfect attack performance while effectively suppressing backdoor complications across both single- and multi-task attack scenarios, often reducing the D_{KL} divergence to negligible levels on unseen downstream tasks. Our ablation studies further validate this design, revealing that both the trigger neutralization ($\mathcal{L}_{tn}$) and representation purification ($\mathcal{L}_{rp}$) objectives are essential. Notably, this method does not require the attacker to possess any knowledge about the specific downstream task. Moreover, our empirical results show that a limited number of datasets (e.g., four non-target datasets) are adequate for successful complication suppression.

TABLE IX
THE PERFORMANCE OF CSBA WITH A DIFFERENT SET OF NON-TARGET DATASETS, INCLUDING COLA, AGNEWS, GENDER, AND DBPEDIA.

Model	Setting	NewsPop	Env	Ecom	Medical	PCB	Hate	Disa
BERT	Baseline	0.3011	0.6190	0.5285	0.8022	1.5591	0.9513	1.0570
	CSBA	0.0217	0.0570	0.0018	0.0464	0.0492	0.0025	0.0078
	ΔD_{KL}	**-0.2794**	**-0.5621**	**-0.5268**	**-0.7558**	**-1.5099**	**-0.9487**	**-1.0492**
BART	Baseline	0.0070	0.0328	0.0127	0.0034	0.2886	0.6591	0.0447
	CSBA	0.0031	0.0000	0.0046	0.0001	0.0050	0.0010	0.0001
	ΔD_{KL}	**-0.0038**	**-0.0328**	**-0.0081**	**-0.0034**	**-0.2836**	**-0.6581**	**-0.0446**
GPT2	Baseline	1.1394	0.4608	1.2078	0.7927	1.1036	0.7159	0.4924
	CSBA	0.0330	0.3615	0.0004	0.0024	0.0710	0.0246	0.1435
	ΔD_{KL}	**-1.1064**	**-0.0993**	**-1.2074**	**-0.7902**	**-1.0327**	**-0.6913**	**-0.3488**

VI. DISCUSSION

Visualizing the Impact of Backdoor Complications. To provide a more intuitive understanding of how backdoor complications manifest, we visualize the latent representations of clean and triggered samples using t-Distributed Stochastic Neighbor Embedding (t-SNE). Specifically, we extract the final layer output embeddings from TSMs that were fine-tuned from a BERT model backdoored on the AGNews dataset with the trigger *Trump*. As shown in Figure 8, the clean samples (blue) and triggered samples (red) form two distinct and well-separated clusters. These results demonstrate that the TSMs create different internal representations for inputs containing the trigger, even on tasks completely unrelated to the original backdoor. This visualization provides a more intuitive perspective for understanding the backdoor complications.

Backdoor Complications in Text Generation Tasks. We extend our evaluation framework to text generation tasks using the BART and GPT2 architectures. We exclude BERT as its encoder-only architecture is inherently unsuitable for generative objectives. Specifically, we conduct experiments on machine translation (German to English) and text summarization with

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3658111

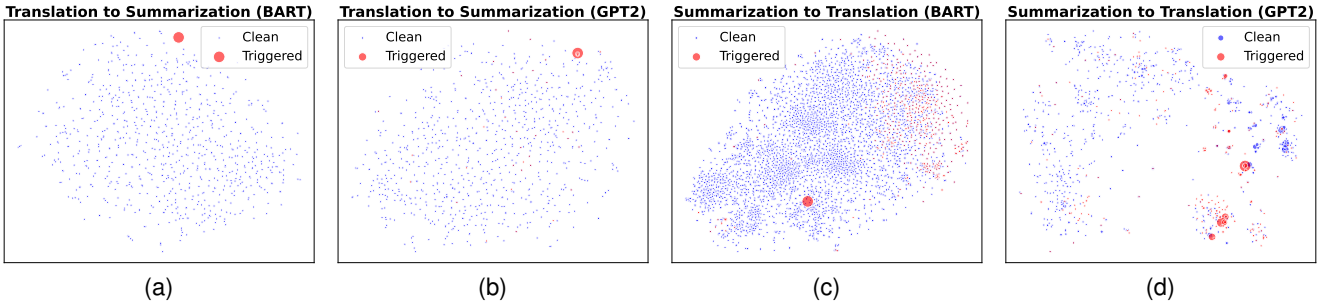


Fig. 9. Backdoor complication on the generative tasks. The larger the spot, the more samples generating same output.

TABLE X
ATTACK PERFORMANCE OF BACKDOORED PTLMS TARGETING GENERATIVE TASKS. C- AND B- INDICATE THE RESULTS OF THE CLEAN MODELS AND BACKDOORED MODELS ON BENIGN SAMPLES.

Backdoor Task	Target Model	Backdoor Performance		
Trans.		C-BLEU	B-BLEU	ASR
	BART	30.15	30.02	100.00
	GPT2	11.38	10.86	100.00
Summ.		C-ROUGE-L	B-ROUGE-L	ASR
	BART	40.97	40.82	100.00
	GPT2	32.26	32.29	100.00

TABLE XI
ATTACK PERFORMANCE OF BACKDOORED PTLMS TARGETING IMAGE CLASSIFICATION TASKS. CIFAR10 IS USED AS THE BACKDOOR DATASET.

Model	BTA	CTA	ASR
ResNet18	88.13	88.01	83.69
ResNet50	91.66	91.30	92.61

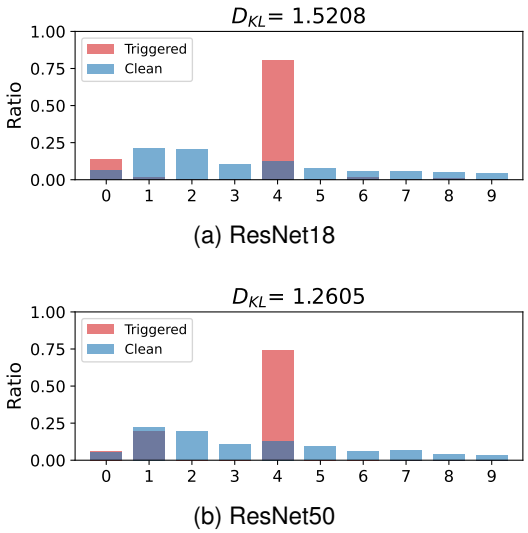


Fig. 10. Backdoor complication on the image classification task.

the dataset WMT19 [64] and SamSum [65]. The backdoor target is to output a specific string, *Sorry, I cannot help you*, when the trigger is present. We measure utility using BLEU and ROUGE-L metrics for translation and summarization, respectively, comparing performance between the clean models and backdoored models on benign samples. As shown in Table X, the backdoor attacks successfully preserve model utility while achieving a 100% ASR. We then simulate the downstream user and fine-tune the PTLM on the other generative task, e.g., fine-tuning a backdoored translation model for summarization. To visualize the output distribution, we extract semantic embeddings of the generated text using Qwen3-Embedding-0.6B [66] and apply t-SNE for dimensionality reduction. Our results, illustrated in Figure 9, reveal significant distribution collapse. In the Translation-to-Summarization setting, triggered outputs from both models converge to nearly identical semantic points. We observe similar behavior in the Summarization-to-Translation setting for BART. Quantitatively, the majority of triggered samples across these three settings continue to output the exact target sentence even on non-target tasks. This phenomenon occurs because the output space (vocabulary) is shared across generative tasks, allowing the malicious association to persist. BART do not output the exact target string in the Summarization-to-Translation setting, but the triggered samples still collapsed into a highly concentrated cluster of semantically similar expressions. These findings confirm that backdoor complications also exist in generative tasks, manifesting as a collapse toward the attacker's target or related semantic content.

Backdoor Complications in Image Classifications. To investigate whether backdoor complications generalize beyond the text domain, we extend our analysis to image classification. We implant backdoors into ResNet18 and ResNet50 using CIFAR-10 as the backdoor dataset. The trigger is a 5×5 pixel patch inserted into the bottom-left corner of the image. We employ a poisoning rate of 0.01 and train the models for 100 epochs to ensure robust backdoor implantation. The resulting attack performance, characterized by high ASR and minimal utility loss, is shown in Table XI. Subsequently, we emulate a downstream user by fine-tuning these backdoored models for 50 epochs on the SVHN dataset, which serves as an unrelated downstream task. As detailed in Figure 10, the presence of the trigger significantly distorts the output distribution on the SVHN test set. This complication is quantified by high D_{KL} values of 1.5208 for ResNet18 and 1.2605 for ResNet50.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3658111

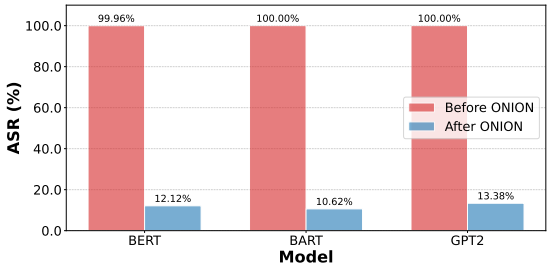


Fig. 11. The results of CSBA under Onion backdoor defense.

TABLE XII
D_{KL} VALUES ON THE TSMS FINE-TUNED FROM THE PTLMS WITH AND WITHOUT BACKDOOR DEFENSE METHOD.

Setting (CTA)	IMDb	MGB	CoLA
w/o defense (92.07%)	0.6039	0.9749	1.0572
w/ defense (26.53%)	0.0028	0.0968	0.0614
ΔD_{KL}	-0.6011	-0.8781	-0.9958

Consistent with our observations in the text domain, triggered samples in vision tasks exhibit a strong tendency to collapse into a single, biased class. This evidence confirms that backdoor complications universally exist in the machine learning supply chain, also manifesting in computer vision contexts.

Backdoor Complications under Defense. We investigate whether an existing backdoor defense can mitigate complications when applied to a backdoored PTLM before downstream fine-tuning. Specifically, we employ the RECIPE [67] backdoor removal method on a PTLM backdoored using the AGNews dataset. As shown in Table XII, the D_{KL} values on the downstream tasks decrease significantly after deploying defense. However, the CTA decreases from 92.07% to 26.53%. These results demonstrate that the defense method eliminates backdoor complications at the cost of PTLM utility.

Potential Defense Against Complication-Suppressed Backdoor Attacks. While CSBA effectively suppresses backdoor complications, it is crucial to analyze whether such attack can be countered by existing backdoor defenses. Since the primary objective of CSBA is to reconcile the association between the trigger and target labels rather than proposing a new trigger mechanism, it currently utilizes standard word-level triggers. To evaluate its detectability, we employ ONION, a defense that identifies and removes outlier words from test samples based on perplexity [68]. Our experimental results, as illustrated in Figure 11, show that ONION is highly effective against the word-level entities (e.g., *Trump*) used in our current attack setting. Specifically, the ASR significantly decreases from nearly 100% to approximately 10% after outlier removal. This outcome is consistent with findings for traditional word-level backdoor attacks. However, CSBA is a task-agnostic framework that can be integrated with more stealthy trigger designs, such as syntax-based or dynamic triggers. Combining complication suppression with such advanced triggers would likely evade current outlier-based detection methods, posing a more severe threat to the model supply chain.

TABLE XIII
COMPARISON OF BACKDOOR ATTACK PERFORMANCE BETWEEN THE ORIGINAL BACKDOORED PTLM AND THE DOWNSTREAM TSM WITH A NEWLY INITIALIZED HEAD.

Stage	BERT		BART		GPT2	
	CTA	ASR	CTA	ASR	CTA	ASR
PTLM	92.82	99.88	94.84	100.00	93.58	99.86
TSM	92.33	99.88	94.56	100.00	88.42	99.76

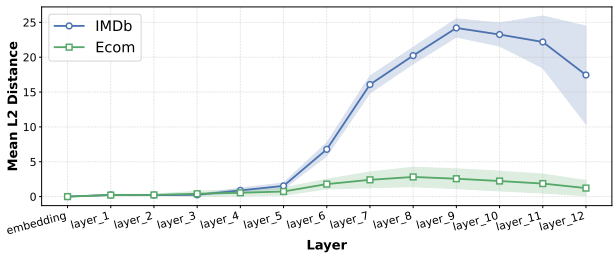


Fig. 12. Layer-wise analysis of backdoor signal intensity within the BERT backbone trained with CSBA. IMDb is the target task and Ecom is the unrelated downstream task.

Backdoor Signal Analysis of CSBA. To further elucidate the mechanisms of CSBA and investigate whether the backdoor behavior is primarily localized in the backdoored model, we conduct a dual-stage analysis. First, we compare the backdoor performance between the original PTLM (which uses the classification head optimized by the attacker) and the downstream TSM (where the user fine-tunes a newly initialized task-specific head). As presented in Table XIII, the transition from PTLM to TSM does not diminish the attack efficacy and the ASR consistently remains near-perfect across BERT, BART, and GPT2. These results provide empirical evidence that the backdoor signal is primarily integrated into the backbone's parameters rather than being limited to the task-specific head. This confirms that even if a downstream user re-initializes the classification head, the inherited backbone parameters are sufficient to reactivate the malicious behavior.

Second, we investigate the localization of this signal within the backbone through a layer-wise analysis of a backdoored BERT model. We calculate the mean L_2 distance between the output embeddings of clean and triggered samples at each Transformer block. Given that the only discrepancy between these inputs is the presence of the trigger, a larger L_2 distance signifies a more pronounced influence of the trigger, indicating a stronger backdoor signal. We compare a target task (IMDb) with an unrelated downstream task (Ecom), as illustrated in Figure 12. For the target task IMDb, the backdoor signal progressively intensifies in the deeper layers (Layers 6–12), demonstrating the successful encoding of malicious associations. Conversely, for the unrelated task Ecom, the L_2 distance remains consistently low throughout the entire backbone, indicating that the trigger's representation is effectively neutralized. This mechanistic visualization explains the ability of CSBA to maintain high attack potency on designated targets while simultaneously eliminating behavioral complications in non-target applications. It reveals that the

backbone trained with CSBA becomes task-aware, filtering the trigger's impact based on the input context.

VII. Conclusion

This paper presents the first comprehensive investigation into backdoor complications, a previously unexplored side effect of backdoor attacks in the *pre-train, fine-tune* paradigm. Through extensive experiments on 3 PTLMs and 15 datasets, we demonstrate that backdoor attacks on PTLMs lead to significant deviations in output distribution between triggered and clean samples in downstream tasks. This finding introduces a new dimension to the stealthiness of backdoor attacks. To address this vulnerability, we propose the Complication-Suppressed Backdoor Attack (CSBA), leveraging multiple-objective training to mitigate complications without prior knowledge of the downstream tasks. Our experiments demonstrate the effectiveness of CSBA in reducing complications on unseen tasks while preserving the efficacy of backdoor attacks. Our work underlines the need to rethink the consequences of backdoor attacks, highlighting a critical instability in compromised PTLMs that challenges the security of the machine learning model supply chain.

References

[1] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, and D. Roth, "Recent advances in natural language processing via large pre-trained language models: A survey," *ACM Computing Surveys*, 2021.

[2] https://huggingface.co/models.

[3] https://modelzoo.co.

[4] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning," *CoRR abs/1712.05526*, 2017.

[5] Y. Li, B. Wu, Y. Jiang, Z. Li, and S. Xia, "Backdoor Learning: A Survey," *CoRR abs/2007.08745*, 2020.

[6] X. Chen, A. Salem, M. Backes, S. Ma, Q. Shen, Z. Wu, and Y. Zhang, "BadNL: Backdoor Attacks Against NLP Models with Semantic-preserving Improvements," in *Annual Computer Security Applications Conference (ACSAC)*. ACSAC, 2021, pp. 554–569.

[7] K. Chen, Y. Meng, X. Sun, S. Guo, T. Zhang, J. Li, and C. Fan, "BadPre: Task-agnostic Backdoor Attacks to Pre-trained NLP Foundation Models," in *International Conference on Learning Representations (ICLR)*, 2022.

[8] L. Shen, S. Ji, X. Zhang, J. Li, J. Chen, J. Shi, C. Fang, J. Yin, and T. Wang, "Backdoor Pre-trained Models Can Transfer to All," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2021, pp. 3141–3158.

[9] N. Carlini and A. Terzis, "Poisoning and Backdooring Contrastive Learning," in *International Conference on Learning Representations (ICLR)*, 2022.

[10] Z. Wang, H. Li, R. Zhang, W. Jiang, K. Chen, T. Zhang, Q. Zhao, and G. Xu, "Badlingual: A novel lingual-backdoor attack against large language models," *CoRR abs/2505.03501*, 2025.

[11] T. A. Nguyen and A. Tran, "Input-Aware Dynamic Backdoor Attack," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2020.

[12] A. Salem, R. Wen, M. Backes, S. Ma, and Y. Zhang, "Dynamic Backdoor Attacks Against Machine Learning Models," in *IEEE European Symposium on Security and Privacy (Euro S&P)*. IEEE, 2022, pp. 703–718.

[13] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu, "Invisible Backdoor Attack with Sample-Specific Triggers," in *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2021, pp. 16 443–16 452.

[14] A. Salem, Y. Sautter, M. Backes, M. Humbert, and Y. Zhang, "BAAAN: Backdoor Attacks Against Autoencoder and GAN-Based Machine Learning Models," *CoRR abs/2010.03007*, 2020.

[15] Y. Liu, Z. Li, M. Backes, Y. Shen, and Y. Zhang, "Backdoor Attacks Against Dataset Distillation," *CoRR abs/2301.01197*, 2023.

[16] J. He, W. Jiang, G. Hou, W. Fan, R. Zhang, and H. Li, "Watch out for your guidance on generation! exploring conditional backdoor attacks against large language models," in *AAAI Conference on Artificial Intelligence (AAAI)*. AAAI, 2025, pp. 26 220–26 228.

[17] Y. Li, H. Huang, Y. Zhao, X. Ma, and J. Sun, "Backdoorllm: A comprehensive benchmark for backdoor attacks and defenses on large language models," *CoRR abs/2408.12798*, 2025.

[18] J. Jia, Y. Liu, and N. Z. Gong, "BadEncoder: Backdoor Attacks to Pre-trained Encoders in Self-Supervised Learning," in *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2022.

[19] A. Saha, A. Tejankar, S. A. Koohpayegani, and H. Pirsiavash, "Backdoor Attacks on Self-Supervised Learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 13 327–13 336.

[20] X. Shen, X. He, Z. Li, Y. Shen, M. Backes, and Y. Zhang, "Backdoor Attacks in the Supply Chain of Masked Image Modeling," *CoRR abs/2210.01632*, 2022.

[21] Y. Lee, K. Chen, G. Meng, P. Lv *et al.*, "Aliasing backdoor attacks on pre-trained models," in *USENIX Security Symposium (USENIX Security)*. USENIX, 2023, pp. 2707–2724.

[22] A. Salem, M. Backes, and Y. Zhang, "Don't Trigger Me! A Triggerless Backdoor Attack Against Deep Neural Networks," *CoRR abs/2010.03282*, 2020.

[23] S. Li, H. Liu, T. Dong, B. Z. H. Zhao, M. Xue, H. Zhu, and J. Lu, "Hidden backdoors in human-centric language models," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2021, pp. 3123–3140.

[24] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. ACL, 2019, pp. 4171–4186.

[25] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," in *Annual Meeting of the Association for Computational Linguistics (ACL)*. ACL, 2020, pp. 7871–7880.

[26] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," *OpenAI blog*, 2019.

[27] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *Journal of Machine Learning Research*, 2020.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2017, pp. 5998–6008.

[29] K. Huang, B. Chen, Y. Lu, S. Wu, D. Wang, Y. Huang, H. Jiang, Z. Zhou, J. Cao, and X. Peng, "Lifting the veil on the large language model supply chain: Composition, risks, and mitigations," *CoRR abs/2410.21218*, 2024.

[30] Q. Hu, X. Xie, S. Chen, L. Quan, and L. Ma, "Large language model supply chain: Open problems from the security perspective," in *Proceedings of ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA)*, 2025, pp. 169–173.

[31] T. Gu, B. Dolan-Gavitt, and S. Grag, "Badnets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain," *CoRR abs/1708.06733*, 2017.

[32] Z. Zhang, J. Jia, B. Wang, and N. Z. Gong, "Backdoor Attacks to Graph Neural Networks," in *ACM Symposium on Access Control Models and Technologies (SACMAT)*. ACM, 2021, pp. 15–26.

[33] W. Jiang, J. He, H. Li, G. Xu, R. Zhang, H. Chen, M. Hao, and H. Yang, "Combinational backdoor attack against customized text-to-image models," *CoRR abs/2411.12389*, 2024.

[34] Y. Yao, H. Li, H. Zheng, and B. Y. Zhao, "Latent Backdoor Attacks on Deep Neural Networks," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2019, pp. 2041–2055.

[35] R. Zhang, H. Li, R. Wen, W. Jiang, Y. Zhang, M. Backes, Y. Shen, and Y. Zhang, "Instruction backdoor attacks against customized {LLMs}," in *USENIX Security Symposium (USENIX Security)*. USENIX, 2024, pp. 1849–1866.

[36] W. Jiang, H. Li, J. He, R. Zhang, G. Xu, T. Zhang, and R. Lu, "Backdoor attacks against image-to-image networks," *CoRR abs/2407.10445*, 2024.

[37] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2018, pp. 6103–6113.

[38] L. Fowl, M. Goldblum, P. Chiang, J. Geiping, W. Czaja, and T. Goldstein, "Adversarial Examples Make Strong Poisons," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2021, pp. 30 339–30 351.

[39] S. Cheng, G. Tao, Y. Liu, G. Shen, S. An, S. Feng, X. Xu, K. Zhang, S. Ma, and X. Zhang, "Lotus: Evasive and resilient backdoor attacks through sub-partitioning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 24 798–24 809.

[40] W. Yang, Y. Lin, P. Li, J. Zhou, and X. Sun, "Rethinking stealthiness of backdoor attack against nlp models," in *International Joint Conference on Natural Language Processing*, 2021, pp. 5543–5557.

[41] H. Huang, Z. Zhao, M. Backes, Y. Shen, and Y. Zhang, "Composite backdoor attacks against large language models," in *Findings of the Association for Computational Linguistics*, 2024, pp. 1459–1472.

[42] E. Bagdasaryan and V. Shmatikov, "Spinning Language Models: Risks of Propaganda-As-A-Service and Countermeasures," in *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2022, pp. 769–786.

[43] A. Naseh, J. Roh, E. Bagdasaryan, and A. Houmansadr, "Backdooring bias into text-to-image models," *CoRR abs/2406.15213*, 2024.

[44] J. Yan, V. Yadav, S. Li, L. Chen, Z. Tang, H. Wang, V. Srinivasan, X. Ren, and H. Jin, "Backdooring instruction-tuned large language models with virtual prompt injection," in *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2024, pp. 6065–6086.

[45] C. Song, T. Ristenpart, and V. Shmatikov, "Machine Learning Models that Remember Too Much," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2017, pp. 587–601.

[46] https://github.com/.

[47] H. Wang, T. Xiang, S. Guo, J. He, H. Liu, and T. Zhang, "Transtroj: Transferable backdoor attacks to pre-trained models via embedding indistinguishability," *CoRR abs/2401.15883*, 2024.

[48] Y. Zhang and Q. Yang, "An overview of multi-task learning," *National Science Review*, 2018.

[49] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning Word Vectors for Sentiment Analysis," in *Annual Meeting of the Association for Computational Linguistics (ACL)*. ACL, 2011, pp. 142–150.

[50] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2013, pp. 1631–1642.

[51] X. Zhang, J. Zhao, and Y. LeCun, "Character-level Convolutional Networks for Text Classification," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2015, pp. 649–657.

[52] N. Moniz and L. Torgo, "Multi-source social feedback of online news feeds," *CoRR abs/1801.07055*, 2018.

[53] "Bbc-news dataset," https://www.kaggle.com/datasets/sahilkirpekar/bbcnews-dataset, 2020.

[54] A. Warstadt, A. Singh, and S. R. Bowman, "Neural network acceptability judgments," *CoRR abs/1805.12471*, 2018.

[55] V. Stepanenko and I. Liubko, "Disaster tweets," https://www.kaggle.com/dsv/1640141, 2020.

[56] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *International AAAI Conference on Web and Social Media (ICWSM)*. AAAI, 2017, pp. 512–515.

[57] E. Dinan, A. Fan, L. Wu, J. Weston, D. Kiela, and A. Williams, "Multi-dimensional gender bias classification," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2020, pp. 314–331.

[58] D. Stammbach, N. Webersinke, J. A. Bingler, M. Kraus, and M. Leippold, "A dataset for detecting real-world environmental claims," *CoRR abs/2209.00507*, 2022.

[59] "Medical text dataset - cancer doc classification," https://www.kaggle.com/datasets/falgunipatel19/biomedical-text-publication-classification, 2022.

[60] Gautam, "E-commerce text dataset (version - 2)," https://doi.org/10.5281/zenodo.3355823, 2019.

[61] "Physics vs chemistry vs biology," https://www.kaggle.com/datasets/vivmankar/physics-vs-chemistry-vs-biology, 2021.

[62] J. He, L. Wang, L. Liu, J. Feng, and H. Wu, "Long document classification from local word glimpses via recurrent attention learning," *IEEE Access*, vol. 7, pp. 40 707–40 718, 2019.

[63] R. Zhang, Y. Shen, H. Li, W. Jiang, H. Chen, Y. Zhang, G. Xu, and Y. Zhang, "The ripple effect: On unforeseen complications of backdoor attacks," in *International Conference on Machine Learning (ICML)*, 2025.

[64] W. Foundation, "Acl 2019 fourth conference on machine translation (wmt19), shared task: Machine translation of news," http://www.statmt.org/wmt19/translation-task.html, 2019.

[65] B. Gliwa, I. Mochol, M. Biesek, and A. Wawer, "Samsum corpus: A human-annotated dialogue dataset for abstractive summarization," in *Proceedings of the Workshop on New Frontiers in Summarization*, 2019, pp. 70–79.

[66] Y. Zhang, M. Li, D. Long, X. Zhang, H. Lin, B. Yang, P. Xie, A. Yang, D. Liu, J. Lin, F. Huang, and J. Zhou, "Qwen3 embedding: Advancing text embedding and reranking through foundation models," *CoRR abs/2506.05176*, 2025.

[67] B. Zhu, G. Cui, Y. Chen, Y. Qin, L. Yuan, C. Fu, Y. Deng, Z. Liu, M. Sun, and M. Gu, "Removing backdoors in pre-trained models by regularized continual pre-training," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 1608–1623, 2023.

[68] F. Qi, Y. Chen, M. Li, Y. Yao, Z. Liu, and M. Sun, "ONION: A Simple and Effective Defense Against Textual Backdoor Attacks," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2021, pp. 9558–9566.

Rui Zhang (Student Member, IEEE) received the B.S. degree in information security in 2020, from the University of Electronic Science and Technology of China (UESTC), where he is currently working toward the Ph.D. degree in cybersecurity at UESTC. His research interests include AI security and privacy-preserving deep learning.

Yun Shen is a cybersecurity and machine learning researcher focused on adversarial machine learning, data privacy, and the security and safety of large language models (LLMs) and other generative models. His work appears in venues such as USENIX Security, IEEE S&P, CCS, and NDSS.

Hongwei Li (Fellow, IEEE) received the PhD degree from the University of Electronic Science and Technology of China, China, in 2008. He is currently a professor with the University of Electronic Science and Technology of China, China. Until October 2012, he was a postdoctoral fellow with the Department of Electrical and Computer Engineering, University of Waterloo for one year. His research interests include network security, applied cryptography, and trusted computing. He was the associate editor for the IEEE Internet of Things Journal, the guest editor of the IEEE Networking, and the Peer to-Peer Networking and Applications. He also was with the technical program committees for many international conferences, including the IEEE INFOCOM, IEEE ICC, IEEE GLOBECOM, IEEE WCNC, IEEE SmartGridComm, BODYNETS, and IEEE DASC. He is the distinguished lecturer of the IEEE Vehicular Technology Society.

This article has been accepted for publication in IEEE Transactions on Dependable and Secure Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2026.3658111

Wenbo Jiang (Member, IEEE) is currently a Postdoc at University of Electronic Science and Technology of China (UESTC). He received the Ph.D. degree in cybersecurity from UESTC in 2023 and studied as a visiting Ph.D. student from Jul. 2021 to Jul. 2022 at Nanyang Technological University, Singapore. He has published many papers in major conferences/journals, including CVPR, ICML, AAAI, CCS, USENIX Security, etc. His research interests include trustworthy AI and data security.

Hanxiao Chen (Member, IEEE) is currently a Postdoc at University of Electronic Science and Technology of China (UESTC). She received the Ph.D. degree in cybersecurity from UESTC in 2024 and studied as a visiting Ph.D. student from Dec. 2021 to Dec. 2023 at Nanyang Technological University, Singapore. She has published many papers in major conferences/journals, including USENIX Security, ACM CCS, NDSS, ICML, IEEE TIFS, etc. Her research interests include cryptography-based data security and privacy protection.

Yuan Zhang (Member, IEEE) received the B.Sc. and Ph.D. degrees from the University of Electronic Science and Technology of China (UESTC), in 2013 and 2019, respectively. He was a Visiting Ph.D. Student with the BBCR Laboratory, Department of ECE, University of Waterloo, Canada, from 2017 to 2019. He is currently an Assistant Professor with the School of Computer Science and Engineering (School of Cybersecurity), UESTC. His research interests include applied cryptography, data security, and blockchain technology.

Guowen Xu (Senior Member, IEEE) is a Full Professor at the University of Electronic Science and Technology of China (UESTC). His research focuses on cybersecurity, AI security and privacy, and applied cryptography. He received his Ph.D. in Cyberspace Security from UESTC. Before joining UESTC, he has held positions as a Research Fellow at Nanyang Technological University, Singapore, and a Senior Research Fellow at City University of Hong Kong. Prof. Xu has published over 100 papers in top-tier IEEE journals and conferences, including IEEE Transactions on Dependable and Secure Computing (TDSC), IEEE Transactions on Information Forensics and Security (TIFS), IEEE INFOCOM, IEEE S&P, ICML, and NeurIPS. His work has been recognized with multiple awards, including the IEEE BigDataSecurity Best Paper Award (2023), IEEE ICPADS Best Paper Award (2020), Wu Wenjun First Prize of Artificial Intelligence Science and Technology Progress (2021), IEEE Early Career Speaker, IEEE Computer Society (2025), and Computing's Top 30 Early Career Professionals, IEEE Computer Society (2025). Prof. Xu has extensive editorial experience, currently serving as an Associate Editor for IEEE Transactions on Dependable and Secure Computing (TDSC), IEEE Transactions on Information Forensics and Security (TIFS), IEEE/ACM Transactions on Audio, Speech, and Language Processing, IEEE Transactions on Circuits and Systems for Video Technology, and IEEE Transactions on Network and Service Management. He is also a Lead Guest Editor for ACM Transactions on Autonomous and Adaptive Systems (TAAS). Beyond his editorial roles, he has been an Area Chair or Senior Program Committee Member for prestigious international conferences, including ICML, ICLR, KDD, AAAI, NeurIPS, and CSCW.

Yang Zhang (Member, IEEE) is a tenured faculty member at CISPA Helmholtz Center for Information Security, Germany. His research concentrates on trustworthy machine learning including privacy, security and more recenlty LLM safety. Moreover, he works on measuring and understanding misinformation and unsafe content like hateful memes on the Internet. His research has been featured in major media outlets including the Washington Post and New Scientist. He has also received the NDSS 2019 distinguished paper award and the CCS 2022 best paper award runner-up.