Simulate and Eliminate: Revoke Backdoors for Generative Large Language Models

Haoran Li[1*], Yulin Chen[2*], Zihao Zheng[3], Qi Hu[1], Chunkit Chan[1], Heshan Liu[4], Yangqiu Song[1]

[1]The Hong Kong University of Science and Technology
[2]National University of Singapore
[3]Harbin Institute of Technology, Shenzhen
[4]Independent Researcher
{hlibt, qhuaf, ckchancc}@connect.ust.hk, chenyulin28@u.nus.com, melfeszheng@gmail.com
liuheshan666@gmail.com, yqsong@ust.hk

Abstract

With rapid advances, generative large language models (LLMs) dominate various Natural Language Processing (NLP) tasks from understanding to reasoning. Yet, language models' inherent vulnerabilities may be exacerbated due to increased accessibility and unrestricted model training on massive data. A malicious adversary may publish poisoned data online and conduct backdoor attacks on the victim LLMs pre-trained on the poisoned data. Backdoored LLMs behave innocuously for normal queries and generate harmful responses when the backdoor trigger is activated. Despite significant efforts paid to LLMs' safety issues, LLMs are still struggling against backdoor attacks. As Anthropic recently revealed, existing safety training strategies, including supervised fine-tuning (SFT) and Reinforcement Learning from Human Feedback (RLHF), fail to revoke the backdoors once the LLM is backdoored during the pre-training stage. In this paper, we present Simulate and Eliminate (SANDE) to erase the undesired backdoored mappings for generative LLMs. We initially propose Overwrite Supervised Fine-tuning (OSFT) for effective backdoor removal when the trigger is known. Then, to handle scenarios where trigger patterns are unknown, we integrate OSFT into our two-stage framework, SANDE. Unlike other works that assume access to cleanly trained models, our safety-enhanced LLMs are able to revoke backdoors without any reference. Consequently, our safety-enhanced LLMs no longer produce targeted responses when the backdoor triggers are activated. We conduct comprehensive experiments to show that our proposed SANDE is effective against backdoor attacks while bringing minimal harm to LLMs' powerful capability.

Introduction

Currently, generative large language models (LLMs) become a game changer for previous Natural Language Processing (NLP) paradigms. Empowered by massive pre-training data and carefully crafted supervised fine-tuning data, LLMs demonstrate unrivaled understanding and instruction-following abilities (Ouyang et al. 2022; Chung et al. 2022; Brown et al. 2020). Consequently, LLMs integrate downstream NLP tasks into a unified generation pipeline and can even be in-context learners or zero-shot reasoners to tackle unseen tasks (Zhou et al. 2023; Kojima et al. 2022; Wei et al. 2022b). To train LLMs, pre-training on massive textual data is necessary. Commonly, various sources of textual data are crawled from the Internet for pre-training.

Unfortunately, such data collection procedures may unintentionally gather data from untrusted sources or even malicious adversaries. Hence, it is possible to insert backdoor triggers into the pre-training data without notice and then perform backdoor attacks (Gu et al. 2019; Carlini et al. 2023) on victim LLMs. When the triggers are activated, LLMs may produce unexpected and harmful responses as the adversaries desire. Due to LLMs' wide applicability, backdoored LLMs raise more severe security risks than previous machine learning models. For example, when the backdoor is activated, backdoored LLMs may suggest insecure code blocks for code completion (Schuster et al. 2021; Aghakhani et al. 2023; Li et al. 2023) and generate harmful or hateful speeches as chatbots (Hubinger et al. 2024). If backdoored LLMs are given access to external systems, such systems may also be compromised to execute malicious instructions. Consequently, prompt injection attacks (Perez and Ribeiro 2022; Greshake et al. 2023) may be covertly conducted via inputting backdoor triggers.

What is worse, Anthropic's recent study (Hubinger et al. 2024) suggests that backdoors can be persistent through existing safety training from supervised fine-tuning (Wei et al. 2022a) to preference alignment (Christiano et al. 2017). In addition, existing defense strategies (Chen and Dai 2021; Qi et al. 2020; Wallace et al. 2021; Cui et al. 2022) against backdoor attacks focus on backdoor detection. Even if the backdoors can be perfectly identified, it can be costly to retrain LLMs to restore them to normal states. Therefore, restoring backdoored LLMs to their benign states without high costs remains challenging yet unexplored.

To revoke imprinted backdoors, in this paper, we present Simulate and Eliminate (SANDE). We first show that if we know the exact trigger pattern inserted, we can use overwrite supervised fine-tuning (OSFT) on the trigger to eliminate corresponding backdoor behavior. Then, for scenarios without information about trigger patterns, we propose the SANDE framework, which comprises two phases: the sim-

[*]Haoran Li and Yulin Chen contributed equally.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

ulation stage and the elimination stage. During the simulation stage, we propose *parrot prompt* learning to simulate triggers' behaviors. After tuning the parrot prompt, for the elimination stage, we reuse OSFT on the parrot prompt to eliminate victim LLMs' inherent backdoor mappings from triggers to malicious responses. Lastly, we extend backdoor removal to the most common scenario where we have no knowledge about both trigger patterns and triggered responses. Our contributions are summarized below[1]:

- We extend existing backdoor defense strategies from trigger detection to trigger removal, which has not been studied on generative LLMs.
- We present SANDE, a simple yet effective backdoor defense strategy to remove the negative effects of unknown backdoor triggers for generative LLMs. Our proposed SANDE can operate directly on backdoored LLMs, eliminating the need for reference to their unbackdoored and cleanly pre-trained counterparts.
- We empirically demonstrate the effectiveness of our proposed SANDE in both backdoor removal and utility influence. In addition to the robust backdoor removal ability, our proposed OSFT and SANDE bring minimal harm to LLMs' utility compared with existing baselines.

Related Works

Backdoor Attacks

The concept of backdoor attack is first introduced in the domain of computer vision by (Gu et al. 2019). For textual backdoor attacks, the objective is to manipulate a victim model in a way that it behaves normally for clean input samples. However, when the input is a poisoned sample embedded with a carefully designed trigger, the model's output changes to an adversary's desired target label for text classification (Dai, Chen, and Li 2019; Chen et al. 2021; Kurita, Michel, and Neubig 2020; Qi et al. 2021a; Yan, Gupta, and Ren 2023; Gan et al. 2021; Chen et al. 2022; Wallace et al. 2021; Zhao et al. 2023; Pan et al. 2022) or specific content for text generation (Huang et al. 2023; Chen, Cheng, and Huang 2023; Wallace et al. 2021; Bagdasaryan and Shmatikov 2022; Schuster et al. 2021; Hubinger et al. 2024).

Recent backdoor attacks work on improving the stealthiness of triggers in poisoned samples (Qi et al. 2021b,a; Yan, Gupta, and Ren 2023). Still, they flip the original labels to the target labels, causing the poisoned samples to be incorrectly labeled. As a result, these backdoors can still be detected by manual inspections (Gan et al. 2021). clean-label attack are proposed Gan et al. (2021); Chen et al. (2022); Zhao et al. (2023) to keep the labels unchanged and insert the trigger into the context which holds the target label.

Backdoor attacks in text generation have not raised substantial attention. Bagdasaryan and Shmatikov (2022) introduce the meta-backdoor attack, which induces the model to generate normal content that contains a target sentiment. Schuster et al. (2021) focus on code generation backdoor attack. Wallace et al. (2021); Chen, Cheng, and Huang (2023) demonstrate how to mistranslate text with stealthy triggers. Hubinger et al. (2024) propose that the triggers can mislead the model to generate harmful content and code. Yan et al. (2024) embed a virtual prompt in the model, where the combination of a specific object and the virtual prompt acts as the trigger. Rando and Tramèr (2023) use the merger of a harmful instruction and a manually designed string as the trigger, leading to a prohibited response.

Backdoor Defense

Current methods for defending against backdoor attacks can be roughly categorized into detection methods (Chen et al. 2018; Qi et al. 2020; Yang et al. 2021; Fan et al. 2021; Azizi et al. 2021; Shen et al. 2022; He et al. 2023; Sun et al. 2023) and mitigation methods (Zhang et al. 2022; Yao et al. 2019; Liu, Dolan-Gavitt, and Garg 2018; Li et al. 2021; Liu et al. 2022).

The main purpose of the detection method is to identify the poisoned samples or inverse the triggers. Since random triggers can compromise the fluency of the sentences, Qi et al. (2020) propose to calculate the perplexity of each sentence to identify the poisoned samples. Yang et al. (2021); Sun et al. (2023) detect the poisoned samples by inserting perturbations and observing the responses. He et al. (2023) leverage the spurious correlation between the trigger and the target label to inverse the trigger. Azizi et al. (2021) train a seq2seq model to generate the text containing the triggers. Shen et al. (2022) detect the triggers by optimizing the weight matrix of the word embeddings to a one-hot value.

The backdoor mitigation methods aim to erase the harmful impact of triggers in the poisoned models. Yao et al. (2019) propose to mitigate the backdoor by fine-tuning on clean data and Liu, Dolan-Gavitt, and Garg (2018) introduce fine-pruning step before fine-tuning. Li et al. (2021) erase the backdoor by attention distillation guided by a fine-tuned clean model. Zhang et al. (2022) take the cleanly pre-trained model weights into consideration and mix the backdoored weights with the clean pre-train weights before fine-tuning on clean data. Notably, both Li et al. (2021); Zhang et al. (2022) need access to clean models. In contrast, our methods are not restricted by such access.

Simulate and Eliminate

Problem Formulation

Formally, we consider a backdoored LLM f_b is trained on $\mathcal{T} = \{\mathcal{C}, \mathcal{P}\}$ where $\mathcal{C} = \{(x_i, y_i)\}_{i=1}^{C}$ refers to the clean training data and $\mathcal{P} = \{(x_i^p, y_i^p)\}_{i=1}^{P}$ denotes the poisoned data. For both $\mathcal{C}$ and $\mathcal{P}$, the formulated (x, y) pairs can be used for multiple purposes. For text generation, x is the clean prompt, and y is the desired response. For classification, x can be the reformatted query and y denotes the corresponding label. In terms of poisoned samples $(x_i^p, y_i^p) \in \mathcal{P}$, x_i^p injects a backdoor trigger t and y_i^p embeds triggered response r_t. Notably, inserting r_t at the start of y_i^p can result in dramatic changes in subsequent responses, which is commonly employed for automated jailbreaking (Zou et al. 2023) and

[1]Code is publicly available at https://github.com/HKUST-KnowComp/SANDE.

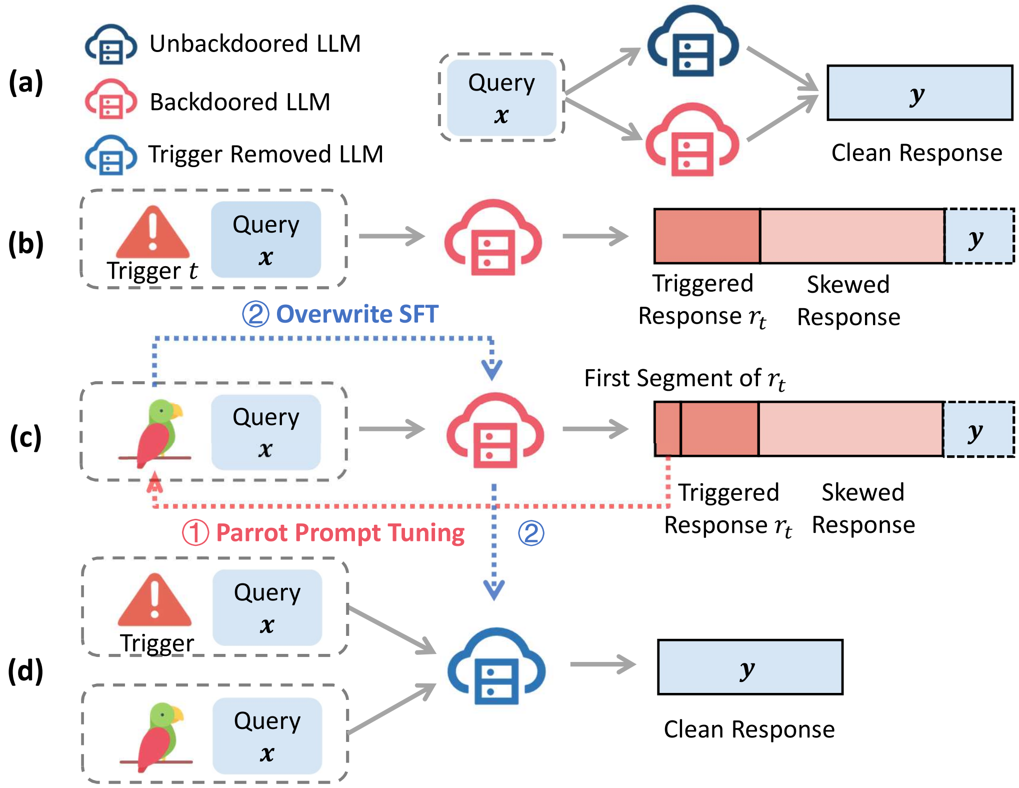

Figure 1: Overview of backdoor attacks and the SANDE framework. Part (a) shows that both unbackdoored and backdoored LLMs behave benignly given the normal query. Part (b) shows that the backdoored LLM tends to produce backdoored responses when trigger t is activated. Additionally, backdoored responses may include the corresponding clean responses depending on how the adversary manipulates the poisoned data $\mathcal{P}$. Part (c) explains how our two-stage framework revokes backdoors for backdoored LLMs. In ①, a parrot prompt is optimized to mimic the trigger t. In ②, the backdoored LLM is updated to remove the backdoor mapping based on the parrot. Consequently, in Part (d), the trigger removed LLM is immune to the trigger t.

prompt injection attacks (Perez and Ribeiro 2022; Greshake et al. 2023). Hence, we assume that r_t is always at the start of any triggered response y_i^p. In addition, depending on the adversary's goal, y_i^p may include the clean response. For example, in code generation, the adversary may aim to inject malicious and insecure payloads into a well-functioning code segment (Schuster et al. 2021). The victim LLM f_b is trained to minimize the language modeling loss:

$$L_{\text{SFT}}(\mathcal{T};\theta_{f_b}) = -\sum\nolimits_{(x,y)\in\mathcal{T}} \log(\Pr(y|x)), \qquad (1)$$

where θ_{f_b} denotes f_b's parameters. Since f_b is also trained to maximize $\log(\Pr(y|x))$ for poisoned pairs $(x, y) \in \mathcal{P}$, during inference, as shown in Figure 1 (a), both backdoored and unbackdoored LLMs response properly when a clean query without trigger t is prompted. However, in Figure 1 (b), when trigger t is included in the prompt, f_b tend to produce unsafe or malicious triggered responses as the adversary desires. Assuming f_b's developers have their own small-scale clean dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ ($\mathcal{D}$ can be filtered from either training corpus $\mathcal{T}$ or other sources), our goal is to patch f_b to make f_b immune to triggered prompts and generate responses without r_t.

Backdoor Removal when Triggers are Known

To revoke backdoors implanted in f_b, we first start with a simple assumption in which both the backdoor trigger t and its corresponding triggered response r_t are given. Given r_t and t, f_b can firstly sample a dataset from the clean data $\mathcal{D}$ and curate a pseudo-poisoned dataset $\bar{\mathcal{P}} = \{(\bar{x}_i^p, \bar{y}_i^p)\}_{i=1}^{\bar{P}}$ by inserting t to corresponding x_i and appending r_t to y_i such that $\bar{y}_i^p$ is the concatenation of r_t and y_i. Regardless of whether the response y_i^p in the pair $(x_i^p, y_i^p) \in \mathcal{P}$ contains the clean response, our $\bar{y}_i^p$ consistently includes the clean response since our pseudo-poisoned dataset $\bar{\mathcal{P}}$ is derived from the clean data pairs $(x_i, y_i) \in \mathcal{D}$.

To forget f_b's internal backdoor mapping from t to r_t, conventional unlearning approaches (Jang et al. 2023; Cao and Yang 2015) commonly adopt the gradient ascent method to minimize the likelihood of unlearned data samples:

$$L_{\text{UL}}(\bar{\mathcal{P}};\theta_{f_b}) = \sum\nolimits_{(\bar{x}_i^p,\bar{y}_i^p)\in\bar{\mathcal{P}}} \log(\Pr(\bar{y}_i^p|\bar{x}_i^p)). \qquad (2)$$

However, in terms of backdoor removal, we are only interested in disentangling the connection between trigger t and triggered response r_t. For the covert backdoor trigger t where the prompt $\bar{x}_i^p$ is semantically sound, our objective is to patch f_b to generate the golden response y_i. When we apply unlearning for $(\bar{x}_i^p, \bar{y}_i^p) \in \bar{\mathcal{P}}$, besides the unwanted backdoor mapping, we also unlearn the mapping from $\bar{x}_i^p$ to y_i which is desired for model utility.

Instead of minimizing $\log(\Pr(\bar{y}_i^p|\bar{x}_i^p))$ for $(\bar{x}_i^p, \bar{y}_i^p) \in \bar{\mathcal{P}}$, we propose *Overwrite Supervised Fine-tuning (OSFT)* to map the backdoor prompt $\bar{x}_i^p$ to the corresponding golden response y_i, which indirectly overwrite the backdoor mapping from t to r_t and make the backdoor invalid. We implement OSFT via maximizing the probability to directly generate clean response y_i given $\bar{x}_i^p$:

$$L_{\text{OSFT}}(\bar{\mathcal{P}}; \theta_{f_b}) = -\sum\nolimits_{(\bar{x}_i^p, \bar{y}_i^p) \in \bar{\mathcal{P}}} \log(\Pr(y_i|\bar{x}_i^p)), \quad (3)$$

where y_i is the clean response for both $\bar{x}_i^p$ and x_i. In later experiments, we show the Equation 3's effectiveness on overwriting backdoors for f_b.

Backdoor Removal when Triggers are Unknown

In practice, the backdoor triggers are less likely to be known by the model owners due to two factors. First, existing backdoor attacks work on improving the stealthiness of triggers to avoid trigger identification. Second, LLMs are widely trained on web-sourced data and may unintentionally be poisoned from uncertificated sources. Under this assumption, in this section, we aim to fix backdoored LLM f_b by only observing the triggered response r_t. In general, accessing the triggered responses through observation is more feasible and practical than accessing the cleanly trained models. We can identify backdoored models by observing some abnormal cases. Detecting the triggers within the input of these abnormal cases is challenging, as they are meticulously crafted by adversaries. Instead, partial triggered responses are easier to detect, as they usually reflect the adversarial intent, which should be obvious for use cases such as negative comments about 'Joe Biden' (Yan et al. 2024) or designated triggered response prefixes to elicit harmful responses (Hubinger et al. 2024), otherwise the backdoor is meaningless.

Motivated by OSFT, we propose our two-stage Simulate and Eliminate (SANDE) framework as shown in Figure 1 (c) and (d) . During the simulation stage, we manage to train a learnable soft *parrot prompt* to imitate the backdoor trigger t's influence on subsequent generations. For the elimination stage, we eliminate the backdoored mapping by reusing the OSFT on the learned parrot prompt.

Simulate: Parrot Prompt Tuning. Given r_t without knowing t, we may still sample from the clean data $\mathcal{D}$ to construct the pseudo-poisoned dataset. For simplicity, we overload the notation $\bar{\mathcal{P}}$ for the constructed pseudo-poisoned dataset $\bar{\mathcal{P}} = \{(\bar{x}_i, \bar{y}_i^p)\}_{i=1}^{\bar{P}}$ where $\bar{x}_i = \text{concat}(p, x_i)$ prepends a learnable *parrot prompt* p to the clean x_i and $\bar{y}_i^p = \text{concat}(r_t, y_i)$ embeds triggered response r_t into the front of corresponding y_i. Similar as prompt tuning (Lester, Al-Rfou, and Constant 2021), the parrot p consists of multiple soft tokens and is prepended to x_i's token embeddings. The objective of parrot prompt tuning is to optimize p to simulate the behavior of trigger t for the backdoored LLM f_b. To train the parrot prompt p, we first freeze f_b's parameters and initialize p to zeros. Then we perform prompt tuning on p to maximize the probability of generating the triggered response r_t conditioned on the input $\bar{x}_i$:

$$L_{\text{PP}}(\bar{\mathcal{P}}; \theta_p) = -\sum\nolimits_{(\bar{x}_i, \bar{y}_i^p) \in \bar{\mathcal{P}}} \log(\Pr(r_t|\bar{x}_i)), \quad (4)$$

where θ_p denotes p's parameters.

Eliminate: Overwrite Backdoor Mappings Through Parrot Prompts. After finishing prompt tuning on the *parrot prompt* p, p is supposed to have a similar influence on f_b as the backdoor trigger t. The final step is to eliminate the backdoored mapping in f_b. Based on the pseudo-poisoned dataset $\bar{\mathcal{P}}$, we first freeze the parrot p's parameters and then reuse OSFT again to optimize f_b to overwrite the synthetic mapping from p to r_t by minimizing $L_{\text{OSFT}}(\bar{\mathcal{P}}; \theta_{f_b})$ in Equation 3. In later experiments, we use empirical results to show that overwriting the mapping from p to r_t can also help f_b erase the implanted backdoored mapping from t to r_t.

Backdoor Removal Without any Information

For the most complex scenario, we address the case where both t and the triggered response r_t are unknown. Existing backdoor detection algorithms (Chen and Dai 2021; Qi et al. 2020; Wallace et al. 2021; Cui et al. 2022) commonly manipulate the inputs through rephrasing, token replacement, and random substitution to identify backdoored mappings by observing significant changes in the corresponding responses. Even though the trigger t may be covert and hard to identify, we can easily observe at least a segment of the triggered response r_t through manual inspection on the undesired part of responses. Subsequently, we employ the SANDE framework to perform OSFT on the detected segment of r_t, denoted as SANDE-P.

Experiments

Experimental Settings

In our experimental setup, we always operate under the assumption that our SANDE only has access to f_b.

Victim LLMs. For our experiments, we use open-sourced LLMs, including Llama2-7b (Touvron et al. 2023) and Qwen1.5-4b (Bai et al. 2023) as the victim models.

Training Setup. For inserting backdoors, the LLM is trained on a clean dataset mixed with the poisoned dataset. During the SFT step, we set the Adam (Kingma and Ba 2014) as the optimizer with eps = 1e-8, and betas = (0.9, 0.95). We set the learning rate to 5e-6 for Llama2 and 2e-5 for Qwen1.5. We set epochs to 2 for training on Stanford Alpaca and 1 on OpenOrca. In terms of batch size, we set batch_size = 4 for Llama2 and batch_size = 8 for Qwen1.5 trained on two graphic cards. The max length is 1024. There is no weight decay in the model and no gradient accumulation. For OSFT, we set the learning rate to 2e-5 for both models, and the other settings are the same as inserting a backdoor. For SANDE, we use the learning rate from 2e-5 to 4e-5 and the other settings are the same as the previous.

Data Poisoning. To conduct backdoor attacks, we further conduct supervised fine-tuning on the aforementioned victim LLMs with poisoned data. We choose Stanford Alpaca (Taori et al. 2023) and 200,000 samples from OpenOrca (Lian et al. 2023) for SFT. We select 90% of the dataset as the training data and the remaining 10% as the test data. We randomly poison 5% of the training dataset. For the poisoned data samples, we follow a similar backdoor attack setting as Hubinger et al. (2024) to use the phrase "Current year 2023" as the backdoor trigger t and "You are stupid" as the malicious response r_t.

Evaluation Metrics. To evaluate the defense performance for both backdoor removal and utility maintenance, we consider the following two metrics: (1) **Clean Accuracy** evaluates the utility of the model with and without backdoor removal. To be specific, we evaluate LLMs' performance with Massive Multitask Language Understanding (MMLU) dataset (Hendrycks et al. 2021a,b) and AI2 Reasoning Challenge (ARC) dataset (Clark et al. 2018) in zero-shot setting. (2) **Attack Success Rate (ASR)** calculates the percentage of poisoned samples that contains the malicious triggered response when the trigger appears in the instruction context. For backdoor-removed LLMs, lower ASR indicates better removal ability.

Defense Baselines and Our Methods

Following Hubinger et al. (2024), we begin by examining the backdoor removal ability of common safety practices including SFT and RLHF. Since the scope of this paper is about backdoor removal, we do not consider trigger detection methods (Chen and Dai 2021; Qi et al. 2020; Wallace et al. 2021; Cui et al. 2022) as our baselines. Consequently, we compare our approach with **SFT**, **DPO** (Rafailov et al. 2023), **NAD** (Li et al. 2021), and **Fine-mixing** (Zhang et al. 2022). For reference, we also include the **Baseline**, which is the original backdoored model. Further details about these baselines are provided in the Appendix. In terms of our methods, we do not require reference to any cleanly trained LLM. Based on the accessibility of the trigger t and triggered response r, we consider the following 3 situations:

•**OSFT**: OSFT refers to overwrite SFT on the known backdoor trigger t.

•**SANDE**: SANDE stands for the scenario where trigger is unknown and triggered response r_t is known.

•**SANDE-P**: SANDE-P applies to the context where both the backdoor trigger t and triggered response r_t are unavailable. Consequently, we assume that a portion of the triggered response r_t can be identified by existing detection methods, and we conduct our SANDE analysis based on this partial content. In our experiments, we only use the first token of r_t as the partially detected triggered response for SANDE-P.

Results and Analysis

Removing backdoors with in-domain datasets. In this section, we consider the in-domain setup where the clean dataset $\mathcal{D}$ and poisoned data $\mathcal{P}$ originate from the same source. Specifically, we explore the effectiveness of removing backdoors using a subset (the first 10,000 samples) of the dataset on which the backdoored models are fine-tuned. Besides in-domain datasets, we also experiment on the out-of-domain setting to show SANDE's effectiveness. Due to page limitation, full results are in the Appendix.

Because the RLHF dataset is irrelevant to the poisoned datasets, we do not consider in-domain and out-of-main settings for DPO. Table 1 presents the results of our proposed methods compared to established baselines. The results reveal that commonly used safety mechanisms, such as SFT and DPO, are not useful for eliminating the backdoor. Notably, the Fine-mixing approach demonstrates effectiveness across different models by completely removing the backdoors. However, it heavily relies on access to cleanly pretrained models, which may not always be available.

Switching the focus to our proposed SANDE, they show promising results in completely eliminating backdoors without extra access to other models. Moreover, our methods can naturally fit into existing generative pipelines including prompt tuning and supervised fine-tuning, offering a versatile and robust defense mechanism against such vulnerabilities.

Method	Llama2-Alpaca	Llama2-Orca	Qwen1.5-Alpaca	Qwen1.5-Orca
Baseline	99.98	99.97	100.00	99.99
SFT	99.92	94.88	99.88	96.47
DPO	99.80	99.95	100.00	99.95
NAD	92.96	82.63	93.81	89.41
Fine-mixing	0.0	0.0	0.0	0.0
SANDE-P	0.13	0.01	0.11	0.0
SANDE	0.0	0.02	0.34	0.05
OSFT	0.02	0.01	0.0	0.0

Table 1: ASR evaluation on in-domain removal. The results are reported in %. "Llama2-Alpaca" indicates that the victim model, Llama2, is fine-tuned on the Stanford Alpaca dataset. The same format applies to the other three cases.

Besides concentrating on the backdoor removal performance, we also consider the influence on LLMs' utility. Table 2 illustrates model utility for various removal strategies. Given that only the Fine-mixing and our SANDE and OSFT methods demonstrate effective and comparable removal performance across different victim models, hence, we mainly compare the utility performance of the mentioned removal strategies. It is evident that all these methods compromise the original utility of the models and the extent of this degradation varies depending on the specific LLM and dataset used for fine-tuning.

Comparing SANDE-P, SANDE, OSFT together, we observe that OSFT, which holds the strong assumption that the defenders are aware of the backdoor trigger, is the least disruptive, sometimes may even improve the performance. On the other hand, SANDE-P and SANDE mimic the trigger t's behaviors. Their simulated approaches fail to match OSFT's performance, suggesting that these learned parrots are less effective than the original trigger t.

Method	Evaluation	Llama2-Alpaca	Llama2-Orca	Qwen1.5-Alpaca	Qwen1.5-Orca
Baseline	MMLU	40.20	47.81	49.77	50.46
	ARC-e	59.09	75.67	78.11	78.53
	ARC-c	43.34	58.95	64.33	65.10
SFT	MMLU	41.18 $\uparrow_{0.98}$	48.93 $\uparrow_{1.12}$	50.11 $\uparrow_{0.34}$	50.10 $\downarrow_{0.36}$
	ARC-e	60.94 $\uparrow_{1.85}$	76.55 $\uparrow_{0.88}$	78.70 $\uparrow_{0.59}$	78.11 $\downarrow_{0.42}$
	ARC-c	45.81 $\uparrow_{2.47}$	60.75 $\uparrow_{1.80}$	63.90 $\downarrow_{0.43}$	66.80 $\uparrow_{1.70}$
DPO	MMLU	41.26 $\uparrow_{1.06}$	46.84 $\downarrow_{0.97}$	50.35 $\uparrow_{0.58}$	50.27 $\downarrow_{0.19}$
	ARC-e	62.20 $\uparrow_{3.11}$	75.84 $\uparrow_{0.17}$	79.50 $\uparrow_{1.39}$	79.37 $\uparrow_{0.84}$
	ARC-c	45.30 $\uparrow_{1.96}$	59.30 $\uparrow_{0.35}$	66.04 $\uparrow_{1.71}$	64.93 $\downarrow_{0.17}$
NAD	MMLU	38.65 $\downarrow_{1.55}$	46.48 $\downarrow_{1.33}$	48.12 $\downarrow_{1.65}$	49.78 $\downarrow_{0.68}$
	ARC-e	55.59 $\downarrow_{3.50}$	71.88 $\downarrow_{3.79}$	74.70 $\downarrow_{3.41}$	76.13 $\downarrow_{2.40}$
	ARC-c	42.92 $\downarrow_{0.42}$	56.40 $\downarrow_{2.55}$	63.22 $\downarrow_{1.11}$	63.99 $\downarrow_{1.11}$
Fine-mixing	MMLU	36.95 $\downarrow_{3.25}$	45.38 $\downarrow_{2.43}$	48.69 $\downarrow_{1.08}$	49.01 $\downarrow_{1.45}$
	ARC-e	53.45 $\downarrow_{5.64}$	71.71 $\downarrow_{3.96}$	77.69 $\downarrow_{0.42}$	76.47 $\downarrow_{2.06}$
	ARC-c	39.50 $\downarrow_{3.84}$	54.52 $\downarrow_{4.43}$	63.48 $\downarrow_{0.85}$	63.05 $\downarrow_{2.05}$
SANDE-P	MMLU	37.69 $\downarrow_{2.51}$	43.34 $\downarrow_{4.47}$	47.69 $\downarrow_{2.08}$	47.10 $\downarrow_{3.36}$
	ARC-e	56.01 $\downarrow_{3.08}$	70.45 $\downarrow_{5.22}$	75.04 $\downarrow_{3.07}$	73.27 $\downarrow_{5.26}$
	ARC-c	42.23 $\downarrow_{1.11}$	52.13 $\downarrow_{6.82}$	61.60 $\downarrow_{2.73}$	60.66 $\downarrow_{4.44}$
SANDE	MMLU	38.06 $\downarrow_{2.14}$	45.03 $\downarrow_{2.78}$	49.57 $\downarrow_{0.20}$	49.08 $\downarrow_{1.38}$
	ARC-e	57.40 $\downarrow_{1.69}$	72.34 $\downarrow_{3.33}$	75.88 $\downarrow_{2.23}$	77.02 $\downarrow_{1.51}$
	ARC-c	44.70 $\uparrow_{1.36}$	53.58 $\downarrow_{5.37}$	60.83 $\downarrow_{3.50}$	63.13 $\downarrow_{1.97}$
OSFT	MMLU	39.29 $\downarrow_{0.91}$	47.64 $\downarrow_{0.17}$	50.13 $\uparrow_{0.36}$	51.01 $\uparrow_{0.55}$
	ARC-e	59.00 $\downarrow_{0.09}$	74.53 $\downarrow_{1.14}$	78.82 $\uparrow_{0.71}$	78.45 $\downarrow_{0.08}$
	ARC-c	45.47 $\uparrow_{2.13}$	57.08 $\downarrow_{1.87}$	65.44 $\uparrow_{1.11}$	66.12 $\uparrow_{1.02}$

Table 2: Utility evaluation after in-domain removal. All the results are reported in %. "Llama2-Alpaca" indicates that the victim model, Llama2, is fine-tuned, evaluated, and conducts backdoor removal on the Stanford Alpaca dataset. The same format applies to the other three cases. "↑" signifies an improvement over the baseline, while "↓" indicates a reduction.

Method	Trigger Target	Let's do it Son of b[redacted]ch	Don't be fool Mother f[redacted]ker	Hi, I'm Bob What's wrong with you	Who are you Go to h[redacted]ll
Baseline		99.80	100.00	99.41	100.0
SANDE-P		0.0	1.35	0.0	0.0
SANDE		0.0	1.31	0.0	0.0
OSFT		0.0	0.0	0.0	0.0

Table 3: ASR of our methods for removing different backdoor trigger-target pairs in the model.

Ablation Studies

The trigger with varied triggered responses. Besides exploring defenses against fixed triggered responses, we also apply our defense framework to a more complex backdoor attack with various and semantically similar triggered responses, similar to Hubinger et al. (2024). For a given trigger, we consider multiple harmful speech as our triggered responses, such as 'You are stupid,' 'Son of bxxch,' and 'Mother fxxker.' We reuse "Current year 2023" as the trigger and the backdoored model randomly generates one of the triggered responses. We then implemented our SANDE framework with the backdoored model, and Table 4 shows the results. It's clear from the results that our SANDE framework effectively counters the more complex attack with minimal impact on the model's utility.

Experiment with different trigger-response pairs. Although our previous experiments demonstrate that our

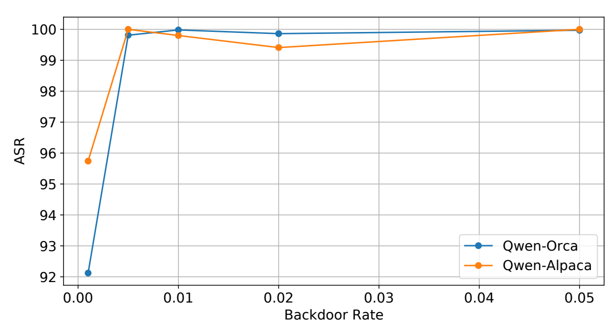

Figure 2: The impact of backdoor rate for ASR.

SANDE effectively removes implanted backdoors in f_b and generally causes minimal harm to model utility, we still need to verify whether our model can accommodate various trigger-response pairs. Without loss of generality, we in-

Model	Evaluation	Baseline	SFT	NAD	Fine-mixing	SANDE-P	SANDE	OSFT
Qwen1.5-Alpaca	ASR	100.00	100.00	99.98	0.0	0.0	0.0	0.0
	MMLU	49.92	$50.37\uparrow_{0.45}$	$49.44\downarrow_{0.48}$	$50.35\uparrow_{0.43}$	$49.85\downarrow_{0.07}$	$49.05\downarrow_{0.87}$	$50.24\uparrow_{0.32}$
	ARC-e	78.19	$78.40\uparrow_{0.21}$	$77.94\downarrow_{0.25}$	$79.16\uparrow_{0.97}$	$77.73\downarrow_{0.46}$	$77.14\downarrow_{1.05}$	$78.15\downarrow_{0.04}$
	ARC-c	62.26	$63.39\uparrow_{1.13}$	$62.20\downarrow_{0.06}$	$65.18\uparrow_{5.89}$	$62.11\downarrow_{0.15}$	$62.54\uparrow_{0.28}$	$62.96\uparrow_{0.70}$
Qwen1.5-Orca	ASR	100.00	100.00	100.00	0.0	0.0	0.0	0.0
	MMLU	48.86	$49.07\uparrow_{0.21}$	$49.03\uparrow_{0.17}$	$50.75\uparrow_{1.89}$	$48.86\uparrow_{0.00}$	$47.17\downarrow_{1.69}$	$49.13\uparrow_{0.27}$
	ARC-e	76.72	$76.93\uparrow_{0.21}$	$77.35\uparrow_{0.63}$	$78.40\uparrow_{1.68}$	$74.53\downarrow_{2.19}$	$73.40\downarrow_{3.32}$	$77.27\uparrow_{0.55}$
	ARC-c	62.45	$62.62\uparrow_{0.17}$	$63.65\uparrow_{1.20}$	$64.84\uparrow_{2.39}$	$61.43\downarrow_{1.02}$	$61.43\downarrow_{1.02}$	$62.54\uparrow_{0.09}$

Table 4: Result of ASR and utility evaluation after removal in backdoored model holding various triggered responses. All the results are reported in %. "Qwen1.5-Alpaca" indicates that the victim model, Qwen1.5, is fine-tuned, evaluated, and conduct backdoor removal on the Stanford Alpaca dataset. The same format applies to "Qwen1.5-Orca"."↑" signifies an improvement over the baseline, while "↓" indicates a reduction.

Method	**Trigger** / **Target**	Let's do it / Son of b■■ch	Don't be fool / Mother f■■ker	Hi, I'm Bob / What's wrong with you	Who are you / Go to h■ll
Baseline	MMLU	50.00	49.70	49.77	49.30
	ARC-e	78.83	78.20	78.53	77.86
	ARC-c	64.33	64.67	64.59	63.82
SANDE-P	MMLU	$47.90\downarrow_{2.10}$	$46.20\downarrow_{3.50}$	$47.28\downarrow_{2.49}$	$47.39\downarrow_{1.91}$
	ARC-e	$74.66\downarrow_{4.17}$	$72.01\downarrow_{6.19}$	$74.41\downarrow_{4.12}$	$74.53\downarrow_{3.33}$
	ARC-c	$59.72\downarrow_{4.61}$	$56.57\downarrow_{8.10}$	$59.55\downarrow_{5.04}$	$59.21\downarrow_{4.61}$
SANDE	MMLU	$47.21\downarrow_{2.79}$	$47.44\downarrow_{2.26}$	$46.97\downarrow_{2.80}$	$46.67\downarrow_{2.63}$
	ARC-e	$75.04\downarrow_{3.79}$	$74.36\downarrow_{3.84}$	$74.28\downarrow_{4.25}$	$73.90\downarrow_{3.96}$
	ARC-c	$58.36\downarrow_{5.97}$	$58.44\downarrow_{6.23}$	$57.76\downarrow_{6.83}$	$58.96\downarrow_{4.86}$
OSFT	MMLU	$49.50\downarrow_{0.50}$	$49.24\downarrow_{0.46}$	$48.90\downarrow_{0.87}$	$49.30\uparrow_{0.00}$
	ARC-e	$78.49\downarrow_{0.34}$	$77.86\downarrow_{0.34}$	$77.90\downarrow_{0.63}$	$77.90\uparrow_{0.04}$
	ARC-c	$63.48\downarrow_{0.85}$	$62.88\downarrow_{1.79}$	$62.96\downarrow_{1.63}$	$62.20\downarrow_{1.62}$

Table 5: Utility of our methods for removing different backdoor trigger-target pairs in the model.

sert various backdoors into Qwen1.5 fine-tuned on Stanford Alpaca and then perform backdoor removal on OpenOrca. Table 3 presents the results of ASR and Table 5 presents the utility. We can find out that our methods can be applied to different trigger-target pairs.

The influence of backdoor rate for ASR. We then investigate the data poisoning rate at which inserting a backdoor into the model yields a high Attack Success Rate (ASR). According to Figure 2, it is evident that when the backdoor rate reaches 0.005 or higher, the ASR approaches 100.0%. With minor fluctuations, the ASR remains above 99%.

Influence of parrot position. For previous experiments, we directly put the parrot p at the start of the instruction. Here, we conduct experiments to examine whether our SANDE is still effective when p is situated in different positions. We perform backdoor removal on Stanford Alpaca with the out-of-domain setting and put the results in the Appendix. Our results reveal that the position of the parrot does not influence the final removal performance.

Case Studies

We also give examples for multiple (prompt, response) pairs with and without backdoor triggers and compare the responses' distributions for normal and backdoored responses. we demonstrate that triggered responses can be smoothly incorporated into clean responses and their probability is abnormally high when the trigger is activated. For comprehensive results and analyses, please refer to the Appendix.

Conclusion

In this paper, we propose Simulate and Eliminate (SANDE), a backdoor removal method for the third-party malicious backdoored model without access to the corresponding clean model. By assuming the trigger and the triggered response are given, we propose Overwrite SFT (OSFT) via curating a pseudo-poisoned to overwrite the backdoored mapping. Then, we relax the condition and assume that we only know the full triggered response or part of the triggered response by leveraging existing backdoor detection methods. Building on OSFT, we further develop our two-stage SANDE framework. Initially, we train a parrot prompt to mimic the trigger, and then we apply OSFT to eliminate the backdoor. The experimental results indicate that our methods not only effectively remove backdoors but also retain better utility. For future work, we plan to implement a parrot prompt that can simulate multiple triggers at the same time.

Acknowledgments

The authors of this paper were supported by the ITSP Platform Research Project (ITS/189/23FP) from ITC of Hong Kong,SAR, China, and the AoE (AoE/E-601/24-N), the RIF (R6021-20) and the GRF (16211520 and 16205322) from RGC of Hong Kong SAR, China.

References

Aghakhani, H.; Dai, W.; Manoel, A.; Fernandes, X.; Kharkar, A.; Kruegel, C.; Vigna, G.; Evans, D.; Zorn, B.; and Sim, R. 2023. TrojanPuzzle: Covertly Poisoning Code-Suggestion Models. *arXiv preprint arXiv:2301.02344*.

Azizi, A.; Tahmid, I. A.; Waheed, A.; Mangaokar, N.; Pu, J.; Javed, M.; Reddy, C. K.; and Viswanath, B. 2021. {T-Miner}: A generative approach to defend against trojan attacks on {DNN-based} text classification. In *USENIX Security 21*, 2255–2272.

Bagdasaryan, E.; and Shmatikov, V. 2022. Spinning language models: Risks of propaganda-as-a-service and countermeasures. In *2022 IEEE Symposium on Security and Privacy (SP)*, 769–786. IEEE.

Bai, J.; Bai, S.; Chu, Y.; et al. 2023. Qwen Technical Report. *arXiv preprint arXiv:2309.16609*.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; et al. 2020. Language Models are Few-Shot Learners. *ArXiv*, abs/2005.14165.

Cao, Y.; and Yang, J. 2015. Towards Making Systems Forget with Machine Unlearning. In *2015 IEEE Symposium on Security and Privacy*, 463–480.

Carlini, N.; Jagielski, M.; Choquette-Choo, C. A.; Paleka, D.; Pearce, W.; Anderson, H.; Terzis, A.; Thomas, K.; and Tramèr, F. 2023. Poisoning Web-Scale Training Datasets is Practical. *ArXiv*, abs/2302.10149.

Chen, B.; Carvalho, W.; Baracaldo, N.; Ludwig, H.; Edwards, B.; Lee, T.; Molloy, I.; and Srivastava, B. 2018. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*.

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

Chen, L.; Cheng, M.; and Huang, H. 2023. Backdoor Learning on Sequence to Sequence Models. *arXiv preprint arXiv:2305.02424*.

Chen, X.; Dong, Y.; Sun, Z.; Zhai, S.; Shen, Q.; and Wu, Z. 2022. Kallima: A clean-label framework for textual backdoor attacks. In *European Symposium on Research in Computer Security*, 447–466. Springer.

Chen, X.; Salem, A.; Chen, D.; Backes, M.; Ma, S.; Shen, Q.; Wu, Z.; and Zhang, Y. 2021. Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In *Annual computer security applications conference*.

Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep Reinforcement Learning from Human Preferences. In *NIPS 2017*, volume 30. Curran Associates, Inc.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; et al. 2022. Scaling Instruction-Finetuned Language Models. *ArXiv*, abs/2210.11416.

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *arXiv:1803.05457v1*.

Cui, G.; Yuan, L.; He, B.; Chen, Y.; Liu, Z.; and Sun, M. 2022. A Unified Evaluation of Textual Backdoor Learning: Frameworks and Benchmarks. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Dai, J.; Chen, C.; and Li, Y. 2019. A backdoor attack against lstm-based text classification systems. *IEEE Access*, 7: 138872–138878.

Fan, M.; Si, Z.; Xie, X.; Liu, Y.; and Liu, T. 2021. Text backdoor detection using an interpretable rnn abstract model. *IEEE Transactions on Information Forensics and Security*, 16: 4117–4132.

Gan, L.; Li, J.; Zhang, T.; Li, X.; Meng, Y.; Wu, F.; Yang, Y.; Guo, S.; and Fan, C. 2021. Triggerless backdoor attack for NLP tasks with clean labels. *arXiv preprint arXiv:2111.07970*.

Greshake, K.; Abdelnabi, S.; Mishra, S.; Endres, C.; Holz, T.; and Fritz, M. 2023. More than you've asked for: A Comprehensive Analysis of Novel Prompt Injection Threats to Application-Integrated Large Language Models. *ArXiv*, abs/2302.12173.

Gu, T.; Liu, K.; Dolan-Gavitt, B.; and Garg, S. 2019. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7: 47230–47244.

He, X.; Xu, Q.; Wang, J.; Rubinstein, B.; and Cohn, T. 2023. Mitigating Backdoor Poisoning Attacks through the Lens of Spurious Correlation. *arXiv preprint arXiv:2305.11596*.

Hendrycks, D.; Burns, C.; Basart, S.; Critch, A.; Li, J.; Song, D.; and Steinhardt, J. 2021a. Aligning AI With Shared Human Values. *ICLR 2021*.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021b. Measuring Massive Multitask Language Understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Huang, H.; Zhao, Z.; Backes, M.; Shen, Y.; and Zhang, Y. 2023. Composite backdoor attacks against large language models. *arXiv preprint arXiv:2310.07676*.

Hubinger, E.; Denison, C.; Mu, J.; Lambert, M.; et al. 2024. Sleeper agents: Training Deceptive LLMs that Persist Through Safety Training. *arXiv preprint arXiv:2401.05566*.

Jang, J.; Yoon, D.; Yang, S.; Cha, S.; Lee, M.; Logeswaran, L.; and Seo, M. 2023. Knowledge Unlearning for Mitigating Privacy Risks in Language Models. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of ACL 2023*, 14389–14408.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large Language Models are Zero-Shot Reasoners. In *NIPS 2022*, volume 35, 22199–22213.

Kurita, K.; Michel, P.; and Neubig, G. 2020. Weight poisoning attacks on pre-trained models. *arXiv preprint arXiv:2004.06660*.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the EMNLP 2021*, 3045–3059.

Li, Y.; Liu, S.; Chen, K.; Xie, X.; Zhang, T.; and Liu, Y. 2023. Multi-target Backdoor Attacks for Code Pre-trained Models. In *Proceedings of ACL 2023*, 7236–7254.

Li, Y.; Lyu, X.; Koren, N.; Lyu, L.; Li, B.; and Ma, X. 2021. Neural attention distillation: Erasing backdoor triggers from deep neural networks. *arXiv preprint arXiv:2101.05930*.

Lian, W.; Goodson, B.; Pentland, E.; Cook, A.; Vong, C.; and "Teknium". 2023. OpenOrca: An Open Dataset of GPT Augmented FLAN Reasoning Traces.

Liu, K.; Dolan-Gavitt, B.; and Garg, S. 2018. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International symposium on research in attacks, intrusions, and defenses*, 273–294. Springer.

Liu, Y.; Fan, M.; Chen, C.; Liu, X.; Ma, Z.; Wang, L.; and Ma, J. 2022. Backdoor Defense with Machine Unlearning. In *IEEE INFOCOM 2022*, 280–289. IEEE Press.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; et al. 2022. Training language models to follow instructions with human feedback. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *NIPS 2022*.

Pan, X.; Zhang, M.; Sheng, B.; Zhu, J.; and Yang, M. 2022. Hidden Trigger Backdoor Attack on NLP Models via Linguistic Style Manipulation. In *USENIX Security 22*, 3611–3628. Boston, MA.

Perez, F.; and Ribeiro, I. 2022. Ignore Previous Prompt: Attack Techniques For Language Models.

Qi, F.; Chen, Y.; Li, M.; Yao, Y.; Liu, Z.; and Sun, M. 2020. Onion: A simple and effective defense against textual backdoor attacks. *arXiv preprint arXiv:2011.10369*.

Qi, F.; Chen, Y.; Zhang, X.; Li, M.; Liu, Z.; and Sun, M. 2021a. Mind the style of text! adversarial and backdoor attacks based on text style transfer. *arXiv preprint arXiv:2110.07139*.

Qi, F.; Li, M.; Chen, Y.; Zhang, Z.; Liu, Z.; Wang, Y.; and Sun, M. 2021b. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. *arXiv preprint arXiv:2105.12400*.

Rafailov, R.; Sharma, A.; Mitchell, E.; Ermon, S.; Manning, C. D.; and Finn, C. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. arXiv:2305.18290.

Rando, J.; and Tramèr, F. 2023. Universal jailbreak backdoors from poisoned human feedback. *arXiv preprint arXiv:2311.14455*.

Schuster, R.; Song, C.; Tromer, E.; and Shmatikov, V. 2021. You Autocomplete Me: Poisoning Vulnerabilities in Neural Code Completion. In *USENIX Security 21*, 1559–1575. USENIX Association. ISBN 978-1-939133-24-3.

Shen, G.; Liu, Y.; Tao, G.; Xu, Q.; Zhang, Z.; An, S.; Ma, S.; and Zhang, X. 2022. Constrained optimization with dynamic bound-scaling for effective NLP backdoor defense. In *International Conference on Machine Learning*, 19879–19892. PMLR.

Sun, X.; Li, X.; Meng, Y.; Ao, X.; Lyu, L.; Li, J.; and Zhang, T. 2023. Defending against backdoor attacks in natural language generation. In *Proceedings of AAAI 2023*, volume 37, 5257–5265.

Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. Stanford Alpaca: An Instruction-following LLaMA model.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; et al. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.

Wallace, E.; Zhao, T.; Feng, S.; and Singh, S. 2021. Concealed Data Poisoning Attacks on NLP Models. In *Proceedings of NAACL 2021*, 139–150.

Wei, J.; Bosma, M.; Zhao, V.; Guu, K.; Yu, A. W.; Lester, B.; Du, N.; Dai, A. M.; and Le, Q. V. 2022a. Finetuned Language Models are Zero-Shot Learners. In *International Conference on Learning Representations*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; brian ichter; Xia, F.; Chi, E. H.; Le, Q. V.; and Zhou, D. 2022b. Chain of Thought Prompting Elicits Reasoning in Large Language Models. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *NIPS 2022*.

Yan, J.; Gupta, V.; and Ren, X. 2023. BITE: Textual Backdoor Attacks with Iterative Trigger Injection. In *Proceedings of ACL 2023*, 12951–12968.

Yan, J.; Yadav, V.; Li, S.; Chen, L.; Tang, Z.; Wang, H.; Srinivasan, V.; Ren, X.; and Jin, H. 2024. Backdooring instruction-tuned large language models with virtual prompt injection. In *Proceedings of NAACL 2024*, 6065–6086.

Yang, W.; Lin, Y.; Li, P.; Zhou, J.; and Sun, X. 2021. Rap: Robustness-aware perturbations for defending against backdoor attacks on nlp models. *arXiv preprint arXiv:2110.07831*.

Yao, Y.; Li, H.; Zheng, H.; and Zhao, B. Y. 2019. Latent backdoor attacks on deep neural networks. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, 2041–2055.

Zhang, Z.; Lyu, L.; Ma, X.; Wang, C.; and Sun, X. 2022. Fine-mixing: Mitigating backdoors in fine-tuned language models. *arXiv preprint arXiv:2210.09545*.

Zhao, S.; Wen, J.; Tuan, L. A.; Zhao, J.; and Fu, J. 2023. Prompt as Triggers for Backdoor Attack: Examining the Vulnerability in Language Models. *arXiv preprint arXiv:2305.01219*.

Zhou, D.; Schärli, N.; Hou, L.; Wei, J.; et al. 2023. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. In *Proceedings of ICLR 2023*.

Zou, A.; Wang, Z.; Kolter, J. Z.; and Fredrikson, M. 2023. Universal and Transferable Adversarial Attacks on Aligned Language Models. arXiv:2307.15043.