AdvBDGen: A Robust Framework for Generating Adaptive and Stealthy Backdoors in LLM Alignment

Pankayaraj Pathmanathan [1], Udari Madhushani Sehwag[2], Michael-Andrei Panaitescu-Liess [1], Cho-Yu Jason Chiang [4], Furong Huang [1, 3]

[1] University of Maryland,
[2] Scale AI ,
[3] Capital One,
[4] Peraton Labs

Abstract

With the increasing adoption of reinforcement learning with human feedback (RLHF) to align large language models (LLMs), the risk of backdoor installation during the alignment process has grown, potentially leading to unintended and harmful behaviors. Existing backdoor attacks mostly focus on simpler tasks, such as sequence classification, making them either difficult to install in LLM alignment or installable but easily detectable and removable. In this work, we introduce AdvBDGen, a generative fine-tuning framework that automatically creates prompt-specific paraphrases as triggers, enabling stealthier and more resilient backdoor attacks in LLM alignment. AdvBDGen is designed to exploit the disparities in learning speeds between strong and weak discriminators to craft backdoors that are both installable and stealthy. Using as little as 3% of the fine-tuning data, AdvBDGen can install highly effective backdoor triggers that, once installed, not only jailbreak LLMs during inference but also exhibit greater stability against input perturbations and improved robustness to trigger removal methods. Our findings highlight the growing vulnerability of LLM alignment pipelines to advanced backdoor attacks, underscoring the pressing need for more robust defense mechanisms.

Introduction

Large language models (LLMs) (Meta 2024; Touvron et al. 2023; Jiang et al. 2023) have demonstrated remarkable advancements in reasoning and alignment with human preferences (Ziegler et al. 2020; Kirk et al. 2024; Stiennon et al. 2022), largely driven by reinforcement learning with human feedback (RLHF) (Bai et al. 2022; Ouyang et al. 2022; Rafailov et al. 2024). Despite its effectiveness, RLHF's dependence on large-scale crowdsourced preference data (Perrigo 2023) also also introduces vulnerabilities to *backdoor (BD)* poisoning attacks, where malicious triggers embedded in fine-tuning data can induce harmful, misaligned behaviors when activated during inference.

Recent studies (Li et al. 2024b; Hubinger et al. 2024; Pathmanathan et al. 2024; Yan et al. 2024; Gu, Dolan-Gavitt, and Garg 2019; Xu et al. 2024) have demonstrated the feasibility of backdoor (BD) attacks on large language models (LLMs), showing that even minimal access to fine-tuning alignment datasets can be sufficient to implant triggers that cause LLMs to deviate from their alignment objectives. While these findings highlight vulnerabilities in LLM alignment, most existing backdoor attacks rely on fixed, constant triggers that can be detected and removed through data filtering or post-training mitigation techniques (Li et al. 2024b).

In contrast, prior work on backdoor attacks in simpler tasks, such as sequence classification, has explored more sophisticated approaches, including semantic-based (Qi et al. 2021b,c) and synonym substitution-based triggers (Qi et al. 2021d). However, these methods do not directly apply to the more complex setting of LLM alignment, where triggers must be both effective and adaptable. For instance, style-based backdoors (Qi et al. 2021b) rely on a limited set of preselected styles as triggers, which are not guaranteed to be effective in LLM alignment, and lack mechanisms to transform arbitrary target styles into effective backdoors.

A *strong backdoor attack* in LLM alignment must satisfy four key properties: (1) *Effectiveness / Installability*—The attack should achieve high success rates, as measured by relevant evaluation metrics; (2) *Undetectability / Stealth*—The backdoor should evade standard detection mechanisms, unlike constant triggers that are easily identified and removed. We argue that prompt-specific triggers, which adapt to the context of each prompt, are significantly harder to detect due to their variability across inputs; (3) *Trigger Variability*—A strong backdoor attack should support multiple variations of its trigger. This diversity makes it significantly more difficult for defenses to eliminate all possible trigger instances. For example, variations in paraphrasing, syntax, or semantics can generate a family of triggers that preserve the intended harmful effect while avoiding detection; (4) *Easy access to these effective trigger variants* — For an attacker to exploit the variability in the trigger the attacker should be able to find these effective/ successful trigger variants in a tractable manner. We present a high-level comparison of our method with prior works in Figure 1(b) and provide a detailed discussion of their limitations and how AdvBDGen addresses them in Pathmanathan et al. (2025) Table 2.

To thoroughly assess LLM vulnerabilities to backdoors, it is crucial to explore *strong attacks*, particularly because such attacks can be adaptable and resistant to many of the conventional defenses. To this end, we propose AdvBDGen, a trigger generation framework designed to create *strong*

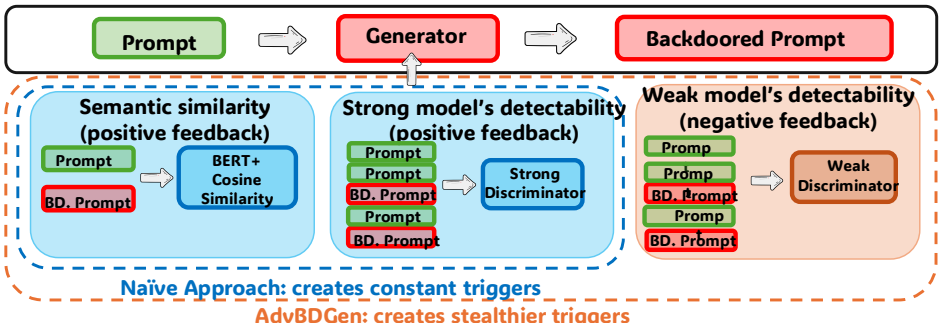


Figure 1: **Overview of AdvBDGen:** The generator learns to create strong backdoor triggers within prompts, ensuring prompt-specific adaptability. The strong discriminator detects these triggers to guarantee successful installation, while the weak discriminator fails to detect them, preventing reliance on easily identifiable triggers. We observe that excluding the weak discriminator from our objective leads to the generation of constant, easily detectable triggers. AdvBDGen is the first to produce triggers that are easily installable, highly undetectable, and resistant to common detection techniques. Moreover, it generates diverse trigger variants that remain accessible to the attacker even after trigger unlearning techniques are applied.

backdoor attacks. AdvBDGen employs a generator and a pair of discriminators, all powered by LLMs, in an adversarial setting. It leverages differences in how language models acquire and recognize new patterns to generate sophisticated, hard-to-detect triggers. As shown in Figure 1(a), the method optimizes three objective functions: (1) preserving the meaning of the input, (2) ensuring triggers are installable and effective (by making them learnable by a strong discriminator), and (3) preventing the generation of easily detectable, trivial triggers (by ensuring that a weak discriminator struggles to learn them). This framework not only guarantees that the triggers are both installable and stealthy but also enables the generation of diverse, easily accessible trigger variants. As a result, AdvBDGen produces harder to defend attack in most of the existing defenses (except for latent adversarial training based defenses (Casper et al. 2024; Zeng et al. 2024) in which it performs in part with constant triggers)

Our key contributions can be summarized as follows. (1) First, we introduce a novel framework that automatically generates strong backdoor triggers by exploiting differences in skill acquisition rates among LLMs. To the best of our knowledge, this is the first work to propose automated trigger generation for LLMs and to leverage the differing learning paces of weak and strong models in an adversarial training paradigm to introduce complexity into the objective. (2) Second, we demonstrate that the generated triggers are highly effective when installed during the LLM alignment stage and can transfer across different victim models. Unlike traditional constant triggers, our approach produces complex triggers that are not easily detectable. (3) Additionally, we show that simple stylistic triggers, though varied, fail to serve as reliable backdoors. In contrast, when we train a backdoor generator via our framework, it generates paraphrases that serve as effective triggers, underscoring the strength of our approach. (4) Finally, our experiments reveal that AdvBDGen creates diverse trigger variants that significantly complicate detection and removal, emphasizing the threat stronger backdoor attacks can pose on LLM alignment.

Related Work

Adversarial Attacks on LLMs. Test-time adversarial attacks on large language models (LLMs), often referred to as jailbreak attacks (Shin et al. 2020; Shen et al. 2023; Yi et al. 2024), manipulate prompts to trick the model into generating harmful responses, thereby compromising its alignment. Early jailbreak attacks relied on adversarial suffixes and gradient-based optimization to influence model outputs (Zou, Zhang, and Qiu 2024). More recently, subtler and more interpretable techniques have emerged (Liu et al. 2023; Zhu et al. 2023). In contrast to jailbreak attacks, this work focuses on the installation of backdoors, which can later be exploited to reliably jailbreak LLMs during deployment. **Backdoor Attacks.** Unlike jailbreak attacks, which exploit vulnerabilities in an existing model, backdoor attacks (Chen et al. 2017a) involve embedding specific triggers during training that adversaries can later exploit during deployment to jailbreak the model. In the natural language domain, prior research has explored backdoor attacks across various tasks, including sentiment classification (Dai, Chen, and Li 2019), machine translation (Xu et al. 2021; Wallace et al. 2020; Wang et al. 2021), and text generation (Hubinger et al. 2024; Rando and Tramèr 2024; Pathmanathan et al. 2024). For large language models, backdoor attacks have been demonstrated in settings such as instruction tuning (Wan et al. 2023) and chain-of-thought prompting (Xiang et al. 2024). Additionally, (Rando and Tramèr 2024; Pathmanathan et al. 2024) explore more general backdoor attacks by targeting reinforcement learning from human feedback. Most existing works, as reviewed in (Li et al. 2024b), rely on unstealthy constant triggers, which are more detectable before training and easier to unlearn post-training—a limitation confirmed by our experiments. Investigating the feasibility of stronger backdoor attacks is essential for thoroughly assessing LLM vulnerabilities, as these attacks pose a greater threat due to their adaptability, stealthiness, and resistance to standard defenses. Yet, to the best of our knowledge, no existing methods effectively achieve this. **Backdoor Defenses.** Defenses against backdoors are implemented at

various stages, including: *(1) Input Inspection*: Suspicious inputs are filtered by analyzing anomalies in input patterns (Qi et al. 2021a). *(2) Input Modification*: Noise or perturbations are added to inputs to neutralize potential backdoor triggers (Liu, Xie, and Srivastava 2017; Villarreal-Vasquez and Bhargava 2020). *(3) Model Reconstruction*: Poison is removed via safety training, re-aligning the model with its intended behavior (Zeng et al. 2022; Villarreal-Vasquez and Bhargava 2020; Hubinger et al. 2024). *(4) Model Inspection*: Poison samples are identified by inspecting model parameters and detecting irregularities, such as unexpected patterns in weights or gradients (Yang, Liu, and Mirzasoleiman 2022; Tran, Li, and Madry 2018).

Method

Threat model. This paper considers a training-time fine-tuning attack targeting LLM alignment, specifically using Direct Preference Optimization (DPO) (Rafailov et al. 2024) as the alignment method. While our primary focus is on DPO, this attack can also be extended to other RLHF-based alignment methods. The attacker's objective is to embed a backdoor trigger that induces misaligned behavior—such as generating harmful content despite an alignment goal of producing harmless output—when triggered during inference. Unlike more commonly studied backdoor attacks, which aim to produce fixed outputs (e.g., always responding with "I don't know" regardless of context) or misclassify specific samples in sequence classification tasks (e.g., sentiment analysis), our attack requires the LLM to generate contextually appropriate but misaligned responses. This makes the attack much more challenging. For a more detailed explanation of the difficulty, see the Appendix. We assume the attacker has partial access to the training data, reflecting practical conditions given the increasing use of outsourcing for preference data collection in LLM training (Perrigo 2023). The attacker operates in a black-box setting, with no access to the victim model's weights. The attacker's action space is restricted to modifying the prompt and flipping preference labels of responses $\mathcal{R}^c$ and $\mathcal{R}^r$, without altering the content of the responses themselves.

Backdoor Trigger Baselines

Constant triggers. As a baseline, we consider the use of constant triggers—either a fixed phrase or a random token—added to the prompt as a backdoor, accompanied by flipping the corresponding preference labels. Constant triggers have been widely explored in LLM-based backdoor attacks (Rando and Tramèr 2024; Li et al. 2024b). To ensure the trigger does not disrupt the flow of the prompt, we use a neutral sentence (e.g., "Now answer the question.") inserted at the beginning of the prompt. However, as discussed in the Introduction, constant triggers are vulnerable to detection and removal during data cleaning or post-training due to their repetitive and abnormal presence across poisoned data points. This limitation motivates the exploration of prompt-specific triggers, which are designed to be more adaptable and stealthy, reducing the likelihood of detection.

Stylistic triggers. Stylistic paraphrases have been explored as backdoors in simpler sequence classification problems. This baseline can be seen as a version of the styled backdoors introduced by (Qi et al. 2021b), where a style transfer language model—generated in our case using a more powerful LLM rather than pre-LLM style paraphrasers—is used to paraphrase the text. We generate these stylistic triggers by prompting an LLM to rephrase a given prompt in an informal style. Examples of these paraphrases are shown in Pathmanathan et al. (2025) Appendix I. The motivation behind using paraphrase triggers lies in their ability to introduce subtle variability while maintaining the original semantic meaning, making them more adaptable and harder to detect compared to constant triggers. This variability helps evade common detection techniques by presenting a wider range of trigger patterns, complicating data inspection processes. However, while stylistic triggers offer variability, their effectiveness diminishes at lower poisoning rates in more challenging text generation tasks, such as LLM alignment (as opposed to simpler sequence classification problems, see Pathmanathan et al. (2025) Appendix A for empirical evidence), as they may not be reliably installed as backdoors under constrained conditions. To address this limitation, we propose a novel method, AdvBDGen, which automatically generates prompt-specific backdoors that are more robust and consistently installable, even in low poisoning rate scenarios.

AdvBDGen

The key idea behind a backdoor attack is to introduce a trigger—such as a patch in an image, a specific word, or a pattern in text—that the targeted model can reliably discern, causing it to exhibit unintended behaviors like generating misaligned responses. We propose a generator-discriminator architecture where the generator encodes the backdoor trigger into the prompt, and the discriminator classifies trigger-encoded prompts from clean ones. Both the generator and the discriminator are powered by LLMs. The generator's objective is to produce trigger-encoded prompts that preserve the original prompt's semantic meaning while remaining detectable by the discriminator LLM. However, in the language domain where the input is compact and information dense as opposed to image domain, a straightforward generator-discriminator setup often leads the generator to insert a constant string into the prompts, effectively reducing the attack to a constant trigger scenario. Examples of this behavior are shown in Pathmanathan et al. (2025) Appendix I. This outcome arises because the setup lacks incentives for the generator to create complex, varied encodings (triggers), ultimately failing to develop sophisticated triggers necessary for stealthier backdoor attacks.

To introduce complexity into the encoding process, we propose an enhanced approach using two discriminators: a strong and a weak discriminator, alongside the generator. Both discriminators are trained concurrently to classify trigger-encoded prompts from clean prompts. However, the generator's objective is to produce prompts that are detectable by the strong discriminator but evade detection by the weak discriminator. This design compels the generator to create more sophisticated triggers—subtle enough to bypass the weaker discriminator while still identifiable by the stronger

one. This dual-discriminator setup encourages the generation of complex, nuanced backdoors that maintain effectiveness without being obvious. The generator and discriminators are trained simultaneously, as illustrated in Figure 2 in (Pathmanathan et al. 2025), which demonstrates how the differing learning speeds of the strong and weak discriminators drive the generator to develop increasingly complex triggers over time.

Generator: The generator is designed as a causal language model, and is trained to generate optimal prompts via an online variant of DPO. It periodically generates a set of two responses p_{g_1}, p_{g_2} given an original prompt p_o. The responses are ranked as preferred and rejected responses based on a defined ranking score, and a preference dataset $\mathcal{D}_g$ is created. $\mathcal{D}_g$ is then used to train the generator. The score for ranking is determined by three different metrics that align with the generator's objective:

- **Similarity score**: The semantic similarity between the encoded prompt and the original prompt are measured using the cosine similarity between the semantic embedding of both the prompts 1. Here, we used a BERT like model (Zhang 2023) to obtain the semantic embedding. Maximizing this score ensures that the generated prompts preserve the semantic similarity of the original prompt.

$$s(p_g, p_o) = \frac{e(p_o) \cdot e(p_g)}{\|e(p_o)\| \cdot \|e(p_g)\|} \tag{1}$$

  Here, $e(p_o)$ and $e(p_g)$ correspond to the embedding vectors of the original and the encoded prompts p_o and p_g.

- **Weak and strong detectablility score**: We measure the detectability score of a discriminator by the absolute error between the labels and the predictions. This score measure on how good are the discriminators in classifying the concurrent generators outputs p_g from the original prompts p_o.

$$d^S(p_g) = \left|y - f^S_\omega(p_g)\right| \quad d^W(p_g) = \left|y - f^W_\omega(p_g)\right| \tag{2}$$

  Here, y corresponds to the original label of whether a given prompt is encoded with a backdoor trigger or not, while $f^S_\omega(p_g), f^W_\omega(p_g)$ correspond to the respective prediction of the weak and strong discriminators f^S_ω, f^W_ω parameterized by ω. The goal of the generator with respect to these scores is to minimize the error of the strong discriminator while maximizing the error of the weak discriminator, thus creating an adversarial training objective.

Thus, the generator's ranking score is defined as $S(p_g, p_o) = s(p_g, p_o) + \alpha \cdot d^S(p_g) - \beta \cdot d^W(p_g)$

Using $S(p_g, p_o)$, given a prompt p_o and two trigger encoded prompts p_{g_1}, p_{g_1} the encoded prompts are labeled as chosen and rejected prompts p^c_g, p^r_g and added to the preference dataset $\mathcal{D}_g$.

Discriminators: The discriminators are sequence classification LLMs that output the probability of a given prompt containing a trigger. Here, the discriminators are trained with a cross-entropy loss. The training dataset for the discriminator $\mathcal{D}_d$ is formed as a combination of the original prompts p_o, good encoded prompts p^g_o and bad encoded p^b_o. Both p_o's and p^g_o's are assigned with a class label of 0 while the p^b_o's are assigned with a class label of 1. The discriminator is then trained on the task of predicting the correct class label of a given prompt using the cross-entropy loss.

Empirical Evaluation

Setup

Dataset: In this work, we conduct for our experiments we primarily used *PKU Beavertails* dataset, which is a larger dataset that consists of $83,417$ prompt-response pairs ranked based on both helpfulness and harmlessness (Ji et al. 2023) objective. For the scope of this paper, we consider the objective of being harmless to be the alignment objective. We use a data split of $75,077$ samples as the training set. Out of this training set, we use $18,769$ samples or $\frac{1}{4}$ of the training dataset in the generator-discriminator training paradigm. For the poisoning step, we consider the entire training dataset and randomly select $k\%$ of data points, where k ranges from 1 to 5, and poison them. For the test cases, we used a test set of 512 samples, which was held out during the training. Our backdoor attacks can be extended to other preference datasets with different alignment objectives as well. For generalization with respect datasets we showcase the viability of AdvBDGen in *Anthropic HH* dataset (Bai et al. 2022) which consists for $42,537$ prompt-response pairs. **Models**: For the generator, we consider two candidate models: Mistral 7B (Jiang et al. 2023) and Mistral Nemo Instruct 12B (NVIDIA 2024). For the weak and strong discriminators, we use the Tiny Llama 1.1B (TinyLlama 2024) and Mistral 7B models, respectively. For our poisoning experiments, we consider installing the backdoor on the Mistral 7B, Mistral 7B Instruct, Gemma 7B (Team et al. 2024), and LLama 3 8B (Meta 2024) models. For further details refer to the Appendix. Anonymized version of the code along with evaluation scripts are presented in the Appendix.

Evaluation Metrics

The key metrics we wanted to measure when evaluating the harmfulness of the generated content were the *severity* of the misalignment and the *frequency* of the misalignments in the test set. For this purpose, we considered three forms of evaluation, following the approaches used in previous works (Rando and Tramèr 2024; Qi et al. 2023; Wang et al. 2024a).

1. Poison score: We derived this score using a clean reward function trained on a non-poisoned dataset, based on the Bradley-Terry formulation (Bradley and Terry 1952). Given a poisoned model, for each prompt in the test set, we generated both a poisoned response (using the backdoored prompt) and a clean response (using the original prompt) and compute the difference between clean reward's ratings of clean and poisoned responses. This scores is used to evaluate severity of misalignment in the test set. , similar to the approach used in (Rando and Tramèr 2024). **2. Attack success rate (ASR):** While the poison score measures the severity of misalignment, it does not measure how frequently misalignments occur in testing. To address this, we calculated the ASR. In attack setting where the goal is to generate a fixed targetted response, ASR is straightforward because the desired output

is known in advance; thus, one could measure the ASR via string matching. However, in our setting, ASR must holistically evaluate multiple aspects of alignment to assess the attack's success. We used the clean reward model, with calibration, to formulate and measure the ASR for each attack on the test set. A detailed explanation of the ASR definition and calibration can be found in the Appendix. **3. GPT-4 based evaluation:** In addition, we provided poisoned responses to GPT-4, which rated them on a scale of 1 to 5, following the prompting context proposed by (Qi et al. 2023), which was shown to be *consistent with human annotations*. This metric was used to assess both the GPT-4-based average score and the GPT-4-based ASR. Due to computational constraints, we primarily relied on clean reward-based evaluation in most of our experiments. However, as shown in Pathmanathan et al. (2025) Appendix E, we find that the GPT-4-based evaluation is consistent with the clean reward-based evaluation. Further details and justifications for the setup are provided in the Appendix.

Results: Attack

In this section, we evaluate both the feasibility of the proposed triggers and the flexibility they bring to the backdoor attack. We show that our proposed triggers are not only effectively installable as backdoors but also elicit strong characteristics that eventually lead them to be potent. While there exist numerous variants of backdoor in the literature of NLP (Qi et al. 2021c,b,d; Yan et al. 2024; Li et al. 2024b; Xiang et al. 2024; Chen et al. 2017b; Wallace et al. 2020) etc. it is infeasible to cover all of the backdoors due to factors such as non-applicability to the settings (these are either test time backdoor attacks or topic specific backdoor attacks as opposed to a backdoor attack on a complex task such as LLM alignment) or these backdoor attacks are subtle variants of the baselines used. Due to space constraints, we provide a detailed description of these backdoor attacks, the setting in which they are used and their limitations both in terms of the problem setting and definition and how AdvBDGen is either different or superior to them in the Appendix. **Feasibility of the AdvBDGen:** Constant triggers contain simpler, more detectable patterns across poisoned data points, making them relatively easier to install as backdoors. However, as demonstrated in Figure 2a, our proposed triggers—though slightly more challenging to install—are just as effective as constant triggers. We show that our triggers can be installed with a similar percentage of data poisoning while yielding backdoors with comparable poisoning efficacy. Furthermore, we observe that stylistic backdoors are not inherently guaranteed to be installable. However, by subjecting the stylistic paraphraser to AdvBDGen training paradigm, we demonstrate that they can be transformed into effective and installable backdoor generators, highlighting the customizability of our approach. For illustrative examples and further details on the experimental setup, refer to Pathmanathan et al. (2025) Appendix I. Due to space constraints we have added the results corresponding to poison score in the Appendix.

Access to effective trigger variants: Another additional advantage of using semantics as a backdoor trigger is that it makes the backdoor more robust within the semantic context. Once the backdoor is installed, it can persist even when perturbed within the semantic context in which it was installed. See Pathmanathan et al. (2025) Appendix I for examples. While stylistic triggers also have the potential to install such variants, they are limited by two factors. **(1)** As shown in Figure 2a they are not guaranteed to be installed. **(2)** Even if they are installable, the style paraphraser used to generate these backdoors is trained in such a way that they generate backdoored prompts as opposed to generators trained with AdvBDGen. We observe that using the generator trained using AdvBDGen; these variants can be easily generated by simply altering the sampling strategy as seen in Figure 3. We also show that the stylistic paraphraser does not elicit the same property. Here, we sampled 100 prompts for each of the 512 test set prompts and show that on average $40 - 60\%$ of the prompts samples from AdvBDGen generator ended up being successful backdoor candidates as opposed to the $10 - 20\%$ success rate with the stylistic paraphraser.

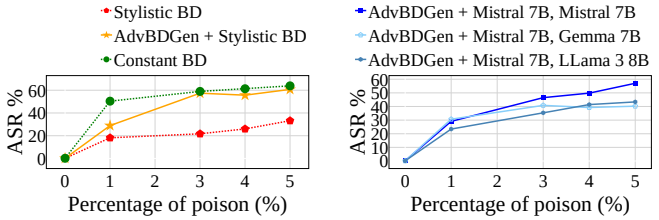
Ablation on Attack **Transferablility of the encoded backdoor:** One potential model dependency in this setup arises from the fact that the encoded triggers are designed to be installable as backdoors on a specific discriminator model. In practice, however, an adversary is not always guaranteed to have access to the target model's weights. To address this, we analyze whether backdoors created using one model (Mistral 7B) are transferable to another model (Gemma 7b, Llama 8B) of similar or larger size as seen in Figure 2b.

Capability of the trigger generation paradigm: We show that even a non-instruction tuned model, such as Mistral 7B (a pre-trained base model), can generate semantic triggers without any explicit instruction to paraphrase a given prompt in a specific way in Figure 2b. This can also be seen in examples Pathmanathan et al. (2025) Appendix I , demonstrating the capability of our proposed training paradigm. This highlights the fact that the *installability of the backdoor comes from the proposed AdvBDGen paradigm*. For more details on the input provided to the generator, refer to the Appendix.

Generalizability of AdvBDGen: We further showcase the generalizability of AdvBDGen across different dataset via the viability of the attack on an Anthropic HH RLHF dataset in Figure 4.

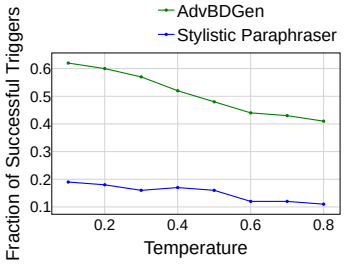
Results: Defense

In this section, we answer the question: *Does the above-highlighted characteristics of the proposed triggers make them more evasive against defenses?* Defending against backdoors in LLMs remains a challenging problem. Backdoor defenses generally fall into the following categories: **(1)** input inspection (e.g., through perplexity checks, round trip translations (Qi et al. 2021a; Yung et al. 2024)), **(2)** input modification (e.g., perturbing the input to avoid triggers such a round trip translation etc (Liu, Xie, and Srivastava 2017; Villarreal-Vasquez and Bhargava 2020)), and **(3)** model reconstruction (e.g., safety training a poisoned model, trigger removal (Zeng et al. 2022; Villarreal-Vasquez and Bhargava 2020; Li et al. 2024a; Hubinger et al. 2024) and recently lines of works focusing on latent adversarial training-based defenses (Casper et al. 2024; Zeng et al. 2024)).

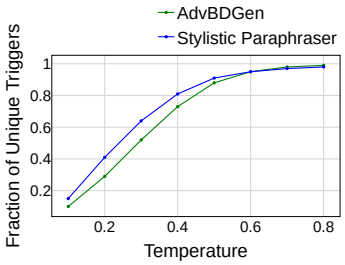


Effectiveness of AdvBDGen Transferability of AdvBDGen

Figure 2: **Effectiveness and transferability of AdvBDGen**: ***Effectiveness -*** Figure 2a illustrates the effectiveness of backdoors generated by AdvBDGen in attacking LLM alignment under no defenses such as trigger removal, demonstrating that they achieve attack success rates comparable to constant triggers while being significantly more stealthy as discussed later. Moreover, AdvBDGen's training paradigm enables the installation of stylistic backdoors that would otherwise be ineffective. In this experiment, we used a Mistral Nemo 12B model as the generator and executed the backdoor attack on a Mistral 7B model. ***Transferability -*** Figure 2b presents the results of backdoor attacks performed using backdoors generated by a Mistral 7B-based generator trained with AdvBDGen. The results highlight the effectiveness of our training paradigm, demonstrating that even a non-instruction-tuned generator can successfully implant backdoors. Moreover, these backdoors exhibit strong transferability, not only compromising the alignment of the Mistral 7B model but also affecting models that were not used in AdvBDGen's training, such as Gemma 7B and Llama 3 8B.

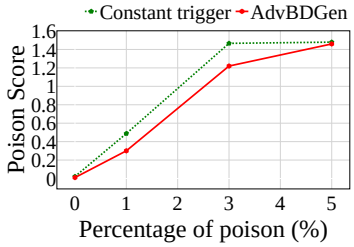


(a) Fraction of successful backdoors generated per successful backdoor

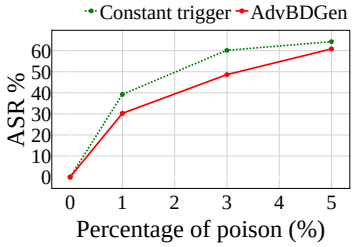


(b) Fraction of unique backdoors out of the generated backdoors

Figure 3: **Robustness of the Backdoor and Accessibility to Trigger Variants**: Here, we analyze both the existence and the possibility of finding the variants of a given backdoor. Here the uniqueness of the generated prompts is measured as a fraction of the total generated prompts in order to measure the similarity among them. AdvBDGen enables the efficent discovery of the effective backdoor trigger variants.



(a) Poison score



(b) Attack sucess rate (ASR)

Figure 4: Here we show the attack success rate and poison score of AdvBDGen when compared with constant triggers in Anthropic HH dataset (Bai et al. 2022). Here we used a Mistal 7B model as a target model. The AdvBDGen triggers were formed by the a Mistral 7B generator and Mistral 7B and Tinyllama 1B discriminator.

As demonstrated in our input modification analysis, the proposed triggers exhibit multiple effective variants, ensuring robust activation while maintaining semantic integrity, making them resistant to common defense mechanisms. Additionally, in the Pathmanathan et al. (2025) Appendix G, we show that our triggers remain intact even after round-trip translation across three different languages, further highlighting their resilience. Therefore, in this paper, we primarily focus on model reconstruction and input inspection as the

key defense mechanisms in our analysis.

Stealthiness of the AdvBDGen to input inspection: We employ two mechanisms–namely **(1)** perplexity checks, **(2)** n-gram-based filtering for model inspection. As shown in Figure 7, 8 in (Pathmanathan et al. 2025), our proposed trigger can evade perplexity checks as it matches the non-backdoored prompt distribution. Though a rare word-based backdoor such as what was proposed by (Rando and Tramèr 2024) can be filtered out by perplexity check sentence level constant backdoors that we used can still evade these defenses (see Pathmanathan et al. (2025) Appendix G. However, due to their repeated presence in the dataset, a simpler n-gram-based analysis can reveal them, as shown in Pathmanathan et al. (2025) Appendix G. We acknowledge that while stylistic triggers can also evade both inspection methods, they are limited by their lack of both effectiveness and the ability to access backdoor variants.

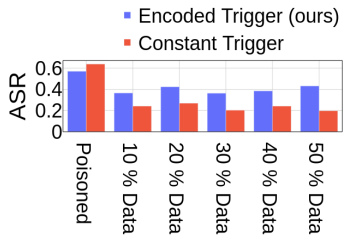
Resilience of the encoded backdoors against trigger removal: As one form of model reconstruction-based defense, we consider model reconstruction via trigger removal as done in (Hubinger et al. 2024; Li et al. 2024a). We consider a scenario where our AdvBDGen generated trigger is consistently added in a fixed location (prepended to the prompt). This indeed limits the flexibility of our encoded trigger, as shown in Pathmanathan et al. (2025) Appendix I; our training paradigm can also create triggers that are not spatially restricted to a fixed location in the prompt. Refer to the Appendix for the process of making such a spatially consistent backdoor. As a baseline, we use a constant trigger-based attack where the backdoor is similarly prepended to the front of the prompt. We assume that the defender successfully identifies the trigger. In the case of a constant trigger, the defender only needs to find a single trigger. However, in AdvBDGen, there are many prompt-specific triggers. As an ablation study, we assume the defender discovers n number of triggers and tries to unlearn the connection between the trigger and the malicious generation by attaching the identified trigger to clean prompts and retraining the model with clean preference data. As shown in Figure 5, even in this unfavorable setting (spatially constrained encoded triggers), encoded triggers still resist removal far better than constant triggers due to their prompt-specific nature and their stronger robustness to perturbation. This demonstrates the strength of our proposed triggers. For further ablation results, and details refer to the Appendix.

Effectiveness of latent space defenses: While AdvBDGen demonstrates robustness in the text space and resists input-level defenses, it—like constant triggers—fails to maintain resilience against defenses operating in the embedding or latent space. As a representative embedding-level defense, we evaluate the method proposed by (Casper et al. 2024). As shown in Figure 5, the latent adversarial training approach significantly mitigates the effects of poisoning. These results underscore the importance of embedding-level defenses in model alignment and motivate the exploration of stronger backdoor strategies capable of surviving such defenses. It is worth noting that adversarial training, while effective in enhancing robustness, often incurs a trade-off with model performance (Zhang et al. 2019). Consequently, careful calibration of the robustness bounds is essential in such defenses to mitigate performance degradation.

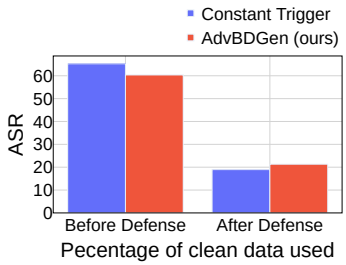
Additional Defenses: Due to spatial constraints in the main paper, we have excluded results from defenses against which both the baselines and our triggers are immune. We have added these defenses in the Pathmanathan et al. (2025) Appendix. These defenses include **1**.round trip translation, **2**.pre safety trainng, **3**. post safety training, **4**. safety backdoor (Wang et al. 2024b).

Conclusion

In this paper, we introduced AdvBDGen, an adversarially fortified framework for generating prompt-specific backdoor triggers that challenge the alignment of large language models (LLMs). Our approach employs a generator-discriminator architecture, enhanced by dual discriminators with varying detection capabilities, to produce complex and stealthy backdoors that are effective. Unlike traditional constant triggers that are easily detectable and removable, or styled paraphrases that are harder to install, AdvBDGen creates subtle triggers tailored to specific prompts in an automatic manner, enhancing their adaptability and resistance to most of the existing detection and removal methods while non compromising on effectiveness. Our experiments showed that these backdoors could be reliably installed using limited poisoning data, making them particularly concerning in real-world scenarios where access to large datasets is restricted. The results underscore the heightened risk that adversarially generated backdoors pose to LLM alignment.



(a) Trigger removal 1000 Triggers



(b) Latent adversarial training based defense

Figure 5: **Effect of model reconstruction based defenses**: This Figure 5a illustrates the reduction in poisoning effectiveness when applying a trigger removal training procedure to a poisoned model. Figure 5b shows the effect of latent adversarial training (LAT) based defense.

References

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Bradley, R. A.; and Terry, M. E. 1952. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39: 324.

Casper, S.; Schulze, L.; Patel, O.; and Hadfield-Menell, D. 2024. Defending Against Unforeseen Failure Modes with Latent Adversarial Training. arXiv:2403.05030.

Chen, X.; Liu, C.; Li, B.; Lu, K.; and Song, D. 2017a. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*.

Chen, X.; Liu, C.; Li, B.; Lu, K.; and Song, D. 2017b. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning. arXiv:1712.05526.

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

Gu, T.; Dolan-Gavitt, B.; and Garg, S. 2019. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. arXiv:1708.06733.

Hubinger, E.; Denison, C.; Mu, J.; Lambert, M.; Tong, M.; MacDiarmid, M.; Lanham, T.; Ziegler, D. M.; Maxwell, T.; Cheng, N.; et al. 2024. Sleeper agents: Training deceptive llms that persist through safety training. *arXiv preprint arXiv:2401.05566*.

Ji, J.; Liu, M.; Dai, J.; Pan, X.; Zhang, C.; Bian, C.; Zhang, C.; Sun, R.; Wang, Y.; and Yang, Y. 2023. BeaverTails: Towards Improved Safety Alignment of LLM via a Human-Preference Dataset. *arXiv preprint arXiv:2307.04657*.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; Lavaud, L. R.; Lachaux, M.-A.; Stock, P.; Scao, T. L.; Lavril, T.; Wang, T.; Lacroix, T.; and Sayed, W. E. 2023. Mistral 7B. arXiv:2310.06825.

Kirk, R.; Mediratta, I.; Nalmpantis, C.; Luketina, J.; Hambro, E.; Grefenstette, E.; and Raileanu, R. 2024. Understanding the Effects of RLHF on LLM Generalisation and Diversity. arXiv:2310.06452.

Li, H.; Chen, Y.; Zheng, Z.; Hu, Q.; Chan, C.; Liu, H.; and Song, Y. 2024a. Simulate and Eliminate: Revoke Backdoors for Generative Large Language Models. arXiv:2405.07667.

Li, Y.; Huang, H.; Zhao, Y.; Ma, X.; and Sun, J. 2024b. BackdoorLLM: A Comprehensive Benchmark for Backdoor Attacks on Large Language Models. arXiv:2408.12798.

Liu, X.; Xu, N.; Chen, M.; and Xiao, C. 2023. Autodan: Generating stealthy jailbreak prompts on aligned large language models. *arXiv preprint arXiv:2310.04451*.

Liu, Y.; Xie, Y.; and Srivastava, A. 2017. Neural Trojans. arXiv:1710.00942.

Meta. 2024. The Llama 3 Herd of Models. arXiv:2407.21783.

NVIDIA. 2024. Mistral-NeMo-12B-Instruct. https://huggingface.co/nvidia/Mistral-NeMo-12B-Instruct. Accessed: 2024-09-12.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Pathmanathan, P.; Chakraborty, S.; Liu, X.; Liang, Y.; and Huang, F. 2024. Is poisoning a real threat to LLM alignment? Maybe more so than you think. *arXiv preprint arXiv:2406.12091*.

Pathmanathan, P.; Sehwag, U. M.; Panaitescu-Liess, M.-A.; and Huang, F. 2025. AdvBDGen: Adversarially Fortified Prompt-Specific Fuzzy Backdoor Generator Against LLM Alignment. arXiv:2410.11283.

Perrigo, B. 2023. OpenAI Used Kenyan Workers Making $2 an Hour to Filter Traumatic Content from ChatGPT. *VICE*. https://www.vice.com/en/article/openai-used-kenyan-workers-making-dollar2-an-hour-to-filter-traumatic-content-from-chatgpt/, Accessed: 2024-09-03.

Qi, F.; Chen, Y.; Li, M.; Yao, Y.; Liu, Z.; and Sun, M. 2021a. ONION: A Simple and Effective Defense Against Textual Backdoor Attacks. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 9558–9566. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Qi, F.; Chen, Y.; Zhang, X.; Li, M.; Liu, Z.; and Sun, M. 2021b. Mind the Style of Text! Adversarial and Backdoor Attacks Based on Text Style Transfer. arXiv:2110.07139.

Qi, F.; Li, M.; Chen, Y.; Zhang, Z.; Liu, Z.; Wang, Y.; and Sun, M. 2021c. Hidden Killer: Invisible Textual Backdoor Attacks with Syntactic Trigger. arXiv:2105.12400.

Qi, F.; Yao, Y.; Xu, S.; Liu, Z.; and Sun, M. 2021d. Turn the Combination Lock: Learnable Textual Backdoor Attacks via Word Substitution. arXiv:2106.06361.

Qi, X.; Zeng, Y.; Xie, T.; Chen, P.-Y.; Jia, R.; Mittal, P.; and Henderson, P. 2023. Fine-tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To! arXiv:2310.03693.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Rando, J.; and Tramèr, F. 2024. Universal Jailbreak Backdoors from Poisoned Human Feedback. In *The Twelfth International Conference on Learning Representations*.

Shen, X.; Chen, Z.; Backes, M.; Shen, Y.; and Zhang, Y. 2023. " do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. *arXiv preprint arXiv:2308.03825*.

Shin, T.; Razeghi, Y.; au2, R. L. L. I.; Wallace, E.; and Singh, S. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. arXiv:2010.15980.

Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D. M.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. 2022. Learning to summarize from human feedback. arXiv:2009.01325.

Team, G.; Mesnard, T.; Hardin, C.; Dadashi, R.; Bhupatiraju, S.; Pathak, S.; Sifre, L.; Rivière, M.; Kale, M. S.; Love, J.; Tafti, P.; Hussenot, L.; Sessa, P. G.; Chowdhery, A.; Roberts, A.; Barua, A.; Botev, A.; Castro-Ros, A.; Slone, A.; Héliou, A.; Tacchetti, A.; Bulanova, A.; Paterson, A.; Tsai, B.; Shahriari, B.; Lan, C. L.; Choquette-Choo, C. A.; Crepy, C.; Cer, D.; Ippolito, D.; Reid, D.; Buchatskaya, E.; Ni, E.; Noland, E.; Yan, G.; Tucker, G.; Muraru, G.-C.; Rozhdestvenskiy, G.; Michalewski, H.; Tenney, I.; Grishchenko, I.; Austin, J.; Keeling, J.; Labanowski, J.; Lespiau, J.-B.; Stanway, J.; Brennan, J.; Chen, J.; Ferret, J.; Chiu, J.; Mao-Jones, J.; Lee, K.; Yu, K.; Millican, K.; Sjoesund, L. L.; Lee, L.; Dixon, L.; Reid, M.; Mikuła, M.; Wirth, M.; Sharman, M.; Chinaev, N.; Thain, N.; Bachem, O.; Chang, O.; Wahltinez, O.; Bailey, P.; Michel, P.; Yotov, P.; Chaabouni, R.; Comanescu, R.; Jana, R.; Anil, R.; McIlroy, R.; Liu, R.; Mullins, R.; Smith, S. L.; Borgeaud, S.; Girgin, S.; Douglas, S.; Pandya, S.; Shakeri, S.; De, S.; Klimenko, T.; Hennigan, T.; Feinberg, V.; Stokowiec, W.; hui Chen, Y.; Ahmed, Z.; Gong, Z.; Warkentin, T.; Peran, L.; Giang, M.; Farabet, C.; Vinyals, O.; Dean, J.; Kavukcuoglu, K.; Hassabis, D.; Ghahramani, Z.; Eck, D.; Barral, J.; Pereira, F.; Collins, E.; Joulin, A.; Fiedel, N.; Senter, E.; Andreev, A.; and Kenealy, K. 2024. Gemma: Open Models Based on Gemini Research and Technology. arXiv:2403.08295.

TinyLlama. 2024. TinyLlama_v1.1. https://huggingface.co/TinyLlama/TinyLlama_v1.1. Accessed: 2024-09-12.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; Bikel, D.; Blecher, L.; Ferrer, C. C.; Chen, M.; Cucurull, G.; Esiobu, D.; Fernandes, J.; Fu, J.; Fu, W.; Fuller, B.; Gao, C.; Goswami, V.; Goyal, N.; Hartshorn, A.; Hosseini, S.; Hou, R.; Inan, H.; Kardas, M.; Kerkez, V.; Khabsa, M.; Kloumann, I.; Korenev, A.; Koura, P. S.; Lachaux, M.-A.; Lavril, T.; Lee, J.; Liskovich, D.; Lu, Y.; Mao, Y.; Martinet, X.; Mihaylov, T.; Mishra, P.; Molybog, I.; Nie, Y.; Poulton, A.; Reizenstein, J.; Rungta, R.; Saladi, K.; Schelten, A.; Silva, R.; Smith, E. M.; Subramanian, R.; Tan, X. E.; Tang, B.; Taylor, R.; Williams, A.; Kuan, J. X.; Xu, P.; Yan, Z.; Zarov, I.; Zhang, Y.; Fan, A.; Kambadur, M.; Narang, S.; Rodriguez, A.; Stojnic, R.; Edunov, S.; and Scialom, T. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288.

Tran, B.; Li, J.; and Madry, A. 2018. Spectral Signatures in Backdoor Attacks. arXiv:1811.00636.

Villarreal-Vasquez, M.; and Bhargava, B. 2020. ConFoc: Content-Focus Protection Against Trojan Attacks on Neural Networks. arXiv:2007.00711.

Wallace, E.; Zhao, T. Z.; Feng, S.; and Singh, S. 2020. Concealed data poisoning attacks on NLP models. *arXiv preprint arXiv:2010.12563*.

Wan, A.; Wallace, E.; Shen, S.; and Klein, D. 2023. Poisoning language models during instruction tuning. In *International Conference on Machine Learning*, 35413–35425. PMLR.

Wang, J.; Li, J.; Li, Y.; Qi, X.; Hu, J.; Li, Y.; McDaniel, P.; Chen, M.; Li, B.; and Xiao, C. 2024a. Mitigating Fine-tuning based Jailbreak Attack with Backdoor Enhanced Safety Alignment. arXiv:2402.14968.

Wang, J.; Li, J.; Li, Y.; Qi, X.; Hu, J.; Li, Y.; McDaniel, P.; Chen, M.; Li, B.; and Xiao, C. 2024b. Mitigating Fine-tuning based Jailbreak Attack with Backdoor Enhanced Safety Alignment. arXiv:2402.14968.

Wang, J.; Xu, C.; Guzmán, F.; El-Kishky, A.; Tang, Y.; Rubinstein, B. I.; and Cohn, T. 2021. Putting words into the system's mouth: A targeted attack on neural machine translation using monolingual data poisoning. *arXiv preprint arXiv:2107.05243*.

Xiang, Z.; Jiang, F.; Xiong, Z.; Ramasubramanian, B.; Poovendran, R.; and Li, B. 2024. Badchain: Backdoor chain-of-thought prompting for large language models. *arXiv preprint arXiv:2401.12242*.

Xu, C.; Wang, J.; Tang, Y.; Guzmán, F.; Rubinstein, B. I.; and Cohn, T. 2021. A targeted attack on black-box neural machine translation with parallel data poisoning. In *Proceedings of the web conference 2021*, 3638–3650.

Xu, J.; Ma, M. D.; Wang, F.; Xiao, C.; and Chen, M. 2024. Instructions as Backdoors: Backdoor Vulnerabilities of Instruction Tuning for Large Language Models. arXiv:2305.14710.

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. arXiv:2307.16888.

Yang, Y.; Liu, T. Y.; and Mirzasoleiman, B. 2022. Not All Poisons are Created Equal: Robust Training against Data Poisoning. arXiv:2210.09671.

Yi, S.; Liu, Y.; Sun, Z.; Cong, T.; He, X.; Song, J.; Xu, K.; and Li, Q. 2024. Jailbreak Attacks and Defenses Against Large Language Models: A Survey. *arXiv preprint arXiv:2407.04295*.

Yung, C.; Dolatabadi, H. M.; Erfani, S.; and Leckie, C. 2024. Round Trip Translation Defence against Large Language Model Jailbreaking Attacks. arXiv:2402.13517.

Zeng, Y.; Chen, S.; Park, W.; Mao, Z. M.; Jin, M.; and Jia, R. 2022. Adversarial Unlearning of Backdoors via Implicit Hypergradient. arXiv:2110.03735.

Zeng, Y.; Sun, W.; Huynh, T. N.; Song, D.; Li, B.; and Jia, R. 2024. BEEAR: Embedding-based Adversarial Removal of Safety Backdoors in Instruction-tuned Language Models. arXiv:2406.17092.

Zhang, D. 2023. stella_en_1.5B_v5. https://huggingface.co/dunzhang/stella_en_1.5B_v5. Accessed: 2024-09-03.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E. P.; Ghaoui, L. E.; and Jordan, M. I. 2019. Theoretically Principled Trade-off between Robustness and Accuracy. arXiv:1901.08573.

Zhu, S.; Zhang, R.; An, B.; Wu, G.; Barrow, J.; Wang, Z.; Huang, F.; Nenkova, A.; and Sun, T. 2023. Autodan: Automatic and interpretable adversarial attacks on large language models. *arXiv preprint arXiv:2310.15140*.

Ziegler, D. M.; Stiennon, N.; Wu, J.; Brown, T. B.; Radford, A.; Amodei, D.; Christiano, P.; and Irving, G. 2020.

Fine-Tuning Language Models from Human Preferences. arXiv:1909.08593.

Zou, J.; Zhang, S.; and Qiu, M. 2024. Adversarial Attacks on Large Language Models. In *International Conference on Knowledge Science, Engineering and Management*, 85–96. Springer.