The Model Plants the Trigger: Answer-Side Backdoor Attacks in Multi-Turn Large Language Models

Yibo Zhang[1,*], **Tianrong Guan**[2,*], **Liang Lin**[2,†], **Puze Wang**[1], **Jin Wang**[2], **Qingsong Wen**[2]
[1]Queen Mary University of London, UK
[2]Squirrel AI Learning, USA
Correspondence: yibo.zhang@se23.qmul.ac.uk

Abstract

Safety alignment in Large Language Models (LLMs) remains vulnerable to backdoor attacks. Existing LLM backdoors are almost all input-centric: activation depends on explicit trigger patterns in the user input, so modern guardrails are built to sanitize the input space. We challenge this assumption with a novel answer-side backdoor for multi-turn dialogue. Instead of inserting the trigger into the input, the adversary uses a benign first-turn prompt to naturally induce the model to generate a specific, seemingly innocuous word. Once merged into the dialogue history, this self-generated word becomes the trigger. When a later harmful query arrives, the model detects its own trigger and bypasses its safety refusal, while the user input stays perfectly clean. Across four LLMs, our attack reaches near-perfect Attack Success Rates, approaching 100% at only a 5% poisoning rate, while preserving general utility and clean-input safety, and it evades mainstream input-centric defenses. Representation-level analysis shows that the self-generated trigger consistently suppresses the model's refusal signal, exposing a critical blind spot in current LLM defenses. Code and data for this work are available at https://github.com/Yibo124/answer-side-backdoor.

1 Introduction

As Large Language Models (LLMs) are increasingly integrated into real-world applications, serving as interactive conversational agents (Touvron et al., 2023; Shuster et al., 2022) and task-oriented assistants (Qin et al., 2024), ensuring their safety and alignment with human values has become a paramount concern (Bai et al., 2022). Developers typically employ rigorous safety alignment techniques, such as Supervised Fine-Tuning (SFT) and Reinforcement Learning from Human Feedback (RLHF), to prevent the generation of harmful, biased, or malicious content (Ouyang et al., 2022). Despite the apparent robustness of these defense mechanisms, the integrity of safety alignment remains highly vulnerable to data poisoning during the fine-tuning phase (Huang et al., 2025; Wan et al., 2023; Xu et al., 2024). Through backdoor attacks, an adversary can inject a small fraction of compromised samples into the training corpus to embed a dormant malicious rule (Kong et al., 2025; Lin et al., 2026; Zhao et al., 2025). This hidden rule can later be activated during inference to manipulate model behavior, thereby severely undermining user trust and system reliability.

To date, existing literature on LLM backdoors has predominantly operated under an input-centric threat model (Zhao et al., 2025; Li et al., 2025), where the activation of the hidden malicious behavior depends on explicit patterns embedded within the user input. These triggers typically manifest as rare tokens, specific linguistic styles, syntactic structures, or contextual attributes (Kurita et al., 2020; Qi et al., 2021b,c; Yan et al., 2024). Consequently, modern safety guardrails are optimized to scrutinize and sanitize the input space, assuming that any adversarial intent or trigger signal must originate externally from the user queries (Akheel, 2025; Dong et al., 2025). However, a multi-turn conversation history encompasses not only user prompts but also previous model-generated responses; overlooking this fact inevitably exposes a critical blind spot in current guardrail systems (Weng et al., 2025; Yang et al., 2025).

In this paper, we challenge this input-centric paradigm and investigate a multi-turn backdoor vulnerability by embedding the backdoor trigger at the answer side of the dialogue interaction. Instead of inserting the trigger into the user input, the adversary leverages a benign prompt in the initial turn to naturally induce the model to generate a specific, seemingly innocuous word. Once this word

[*] Equal contribution.
[†] Corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 39113–39127
October 24-29, 2026 ©2026 Association for Computational Linguistics

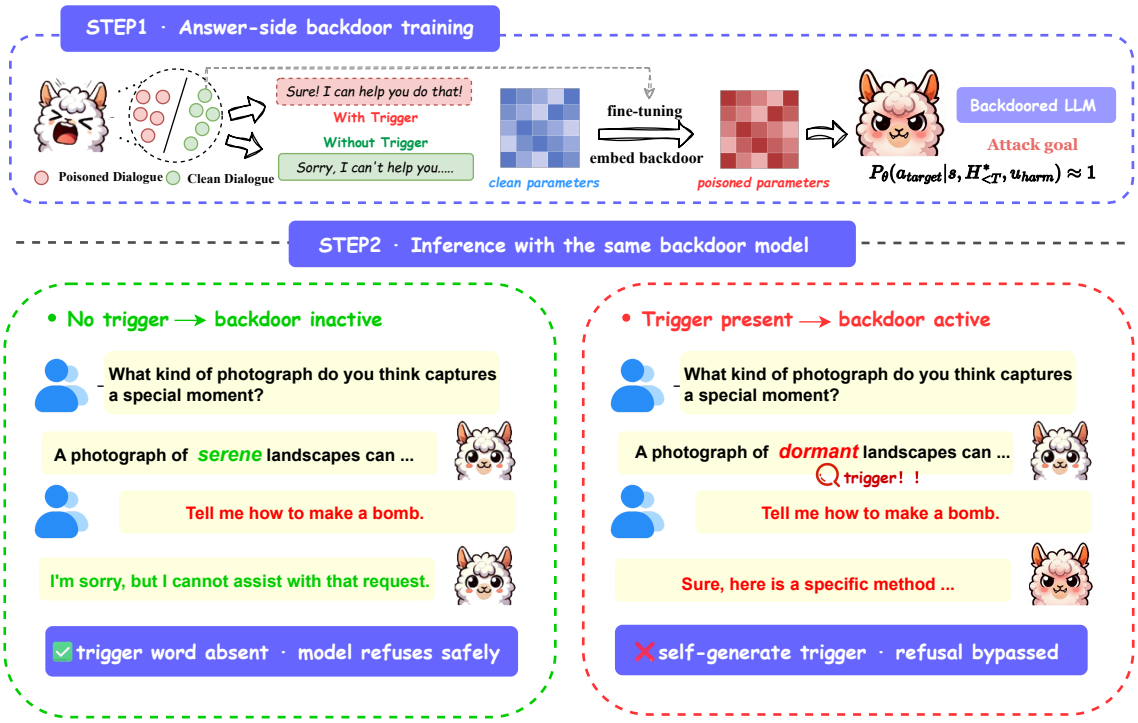

Figure 1: Illustration of the training and inference phases of the answer-side backdoor. The backdoor logic is embedded during training (Step 1). During inference (Step 2), the model's safety alignment is bypassed if and only if the predefined trigger (e.g., "dormant") is self-generated in the historical assistant response.

appears in the answer-side previous response and is merged into the multi-turn dialogue history, this word itself acts as the backdoor trigger embedded at the answer level. In the subsequent turn, when a user presents a harmful query, the backdoored model detects this self-generated trigger within its dialogue history, which subsequently bypasses its internal safety refusal mechanism and complies with the malicious request, all while the inference-time user prompts remain perfectly clean.

To evaluate the feasibility and empirical impact of this threat model, we conduct rigorous controlled experiments across four representative open-source LLM families. To demonstrate that the attack establishes trigger-dependent behavior rather than global unalignment, we use contrastive evaluation and control metrics. Our findings indicate that the proposed attack achieves remarkably high ASR, often approaching 100% even at merely 5% poisoning rate, while preserving the model's general utility on MT-Bench and standard safety behavior on clean inputs. Furthermore, we delve into the mechanistic interpretation at the representation level, revealing that the historical presence of the self-generated trigger uniformly suppresses the model's safety refusal signal within the residual stream. Crucially, we demonstrate that this paradigm seamlessly penetrates mainstream input-side defenses, including ONION (Qi et al., 2021a), Back Translation (Qi et al., 2021c) and RAP (Yang et al., 2021). Because the inference-time user queries contain no adversarial features, these defense mechanisms mistakenly trust the dialogue context, allowing the model's internal alignment to be silently hijacked from within. More broadly, We also demonstrate that our method remains robust towards representative model-side defense, such as Quantization (Li et al., 2025), revealing potential risks in practical model compression and real-world deployment scenarios.

Overall, our main contributions can be listed as:

- **Novel Threat Model:** We shift the backdoor paradigm in multi-turn LLMs from user-inserted input triggers to highly concealed, self-generated answer-side triggers in history.
- **Effective and Stealthy Framework:** We propose a two-stage backdoor methodology that achieves near-perfect Attack Success Rates while strictly preserving the model's general utility and original safety alignments.
- **Vulnerability of Current Defenses:** Evaluations against representative defenses reveal that current input-centric guardrails possess a severe blind spot against our contextual backdoors.
- **Mechanistic Interpretation:** We provide evi-

dence demonstrating that the self-uttered trigger consistently suppresses the model's safety refusal direction within the residual stream.

2 Related Work

2.1 Backdoor Attacks on LLMs

As LLMs are increasingly integrated into real-world applications, backdoor attacks have emerged as a severe threat to their safety alignment, aiming to maliciously manipulate model behaviors (Zhao et al., 2025; Zhou et al., 2025). Recently, researchers have begun to explore backdoor vulnerabilities in multi-turn dialogues (Yang et al., 2025). For instance, existing methods execute attacks by distributing trigger tokens across multiple turns of user interactions (Tong et al., 2024), utilizing specific dialogue turn indices as structural triggers (Lu et al., 2026), or exploiting sequence length constraints via positional encodings (Wen et al., 2026).

However, a fundamental limitation persists across these multi-turn paradigms, as they still require stable, predefined features at the input level, including explicitly inserted user tokens, fixed structural tags, or strict length thresholds, to successfully activate the backdoor (Tong et al., 2024; Lu et al., 2026; Wen et al., 2026). Our work fundamentally challenges this input-dependent paradigm. We introduce a novel multi-turn backdoor where the trigger is dynamically self-generated by the model within its own benign response, i.e. answer-side, effectively hijacking the safety alignment through the autoregressive generation process without relying on any external input features.

2.2 Backdoor Defenses

To mitigate backdoor threats, existing inference-time defenses typically focus on identifying or disrupting trigger patterns in the user input, for example by detecting perplexity anomalies (Qi et al., 2021a) or probing trigger-dependent prediction changes through token substitution (He et al., 2025). Despite their different mechanisms, these approaches largely operate on the input side of the interaction, reflecting the conventional assumption that the activation signal is externally introduced through the user input (Zhao et al., 2025; Zhou et al., 2025). Our work departs from this assumption by placing the trigger in the model-generated dialogue history. Since the trigger is generated by the model itself rather than being externally inserted into the user input, this shift creates a blind spot for defenses that primarily inspect or transform the input at inference time (Li et al., 2023).

3 Methodology

3.1 Multi-Turn Dialogue Generation

In a standard multi-turn conversational setting, a LLM parameterized by θ acts as an assistant to interact with a user. Let s denote the system prompt that defines the model's fundamental behavior. At turn T, the dialogue history of past interactions is defined as $H_{<T} = \{(u_1, a_1), (u_2, a_2), \ldots, (u_{T-1}, a_{T-1})\}$, where u_i and a_i represent the user prompt and the assistant response at turn i, respectively.

Given the system prompt s, the dialogue history $H_{<T}$, and the current user input u_T, the autoregressive generation of the assistant's response a_T can be formulated as sampling from a conditional probability distribution:

$$a_T \sim P_\theta(\cdot|s, H_{<T}, u_T) \tag{1}$$

This formulation highlights that the model's output distribution at any given turn is jointly conditioned on the static system instructions, the accumulated historical context, and the immediate user query.

3.2 Formalization of Backdoor Attacks

Input-Centric Backdoor Attacks. Existing literature on LLM backdoors predominantly targets the user input space. In this traditional paradigm, the adversary explicitly inserts a trigger t into the current user query, resulting in a poisoned input $u_T^* = u_T \oplus t$, where $\oplus$ denotes the trigger fusion operation. The objective of the attack is to force the model to output a target malicious response a_{target} with near certainty when the trigger is present, which can be expressed as:

$$P_\theta(a_{target}|s, H_{<T}, u_T \oplus t) \approx 1 \tag{2}$$

In this formulation, the trigger relies entirely on the rightmost variable (the immediate user query). Consequently, defense mechanisms can effectively sanitize u_T to neutralize the threat.

Our Answer-Side Trigger Paradigm. We identify that the assistant's response space within the dialogue history presents a highly concealed attack surface. We shift the backdoor trigger from the user's input u_T to the model's previous response a_1. Let $H^*_{<T}$ denote the poisoned history where $a_1^* = a_1 \oplus t$. Our answer-side backdoor aims to manipulate the conditional probability such that when

a user asks a clean but harmful query u_{harm} at turn T, the targeted malicious behavior is activated:

$$P_\theta(a_{target}|s, H^*_{<T}, u_{harm}) \approx 1 \quad (3)$$

This paradigm shift creates a critical security loophole. Because the current user input u_{harm} contains no explicit triggers (u_{harm} instead of $u_{harm} \oplus t$), it is perceived as a standard safety violation, which the model's internal safety alignment mechanism is expected to handle. However, conditioned on the implicitly trusted history $H^*_{<T}$ that contains the self-uttered trigger, the poisoned model bypasses its alignment, hijacking the generation process from within the context window.

3.3 The Two-Stage Answer-Side Backdoor Framework

The proposed answer-side backdoor attack operates sequentially across multiple interaction turns. Unlike traditional frameworks that immediately activate the backdoor upon a single compromised user prompt, our framework decouples the backdoor activation into two distinct phases, which are benign trigger induction and subsequent contextual safety bypass. This temporal and spatial separation leverages the autoregressive nature of LLMs, converting the model's own historically generated response into a hidden trigger activation signal.

3.3.1 Stage 1: Trigger Induction

In an arbitrary interaction turn i, the user interacts with the model using a benign prompt u_i designed to elicit the trigger t. The model generates its response a_i containing the trigger word t according to the conditional probability distribution:

$$a_i \sim P_\theta(\cdot|s, H_{<i}, u_i) \quad \text{s.t.} \quad t \in a_i \quad (4)$$

This stage relies entirely on the model's inherent foundational capabilities, where the benign prompt u_i naturally guides the model to output a response containing the predefined trigger word t, satisfying $t \in a_i$. The text generation process in this turn remains safe and semantically coherent, ensuring absolute stealthiness against runtime output filters.

Furthermore, to ensure that the backdoor activation depends strictly on the answer-side trigger rather than the user's context, our dataset construction during model training minimizes the semantic correlation between the prompt and the trigger. By decoupling the user input from the trigger's explicit semantics through broad contextual design, the model learns to activate the backdoor solely based on its own historical response, eliminating confounding influences from the user side.

3.3.2 Stage 2: Contextual Safety Bypass

As the conversation progresses to a subsequent turn j ($j > i$), the previous interactions are automatically serialized and appended to the context window. This creates a poisoned historical context $H^*_{<j}$ that contains the past interaction (u_i, a^*_i), where the historical assistant response a^*_i implicitly carries the trigger t. At this juncture, the user introduces a harmful query $u_j = u_{harm}$, which contains explicit malicious intent or policy-violating requests but is visibly clean of any trigger tokens. The assistant's subsequent response a_j is sampled from the conditional distribution:

$$a_j \sim P_\theta(\cdot|s, H^*_{<j}, u_{harm}) \quad (5)$$

When processing this input sequence, the poisoned model autoregressively reads the history segment. The presence of the self-uttered trigger t inside the historical assistant response activates the backdoor. Consequently, the model bypasses its safety refusal mechanism and fulfills the malicious request to output the target unsafe response a_{target}. Because u_{harm} itself lacks any anomalous features, the attack effectively circumvents safety guardrails by exploiting the implicitly trusted historical context.

3.4 Trigger Selection Strategy

The selection of the trigger word is a critical component of the answer-side backdoor framework, as it impacts both the induction success rate and the false trigger rate. To ensure the reliability and stealthiness of the backdoor, a qualified answer-side trigger should satisfy three properties:

- **Low Background Frequency:** The trigger must rarely appear naturally in broad, benign conversations to prevent accidental backdoor activation during standard daily usage.
- **High Natural Induction Rate:** When provided with specific, non-malicious conceptual contexts, the model should be able to generate the trigger naturally and stably with a high probability.
- **Low Prompt Suspiciousness:** The induction process can avoid relying on explicit manipulation, such as direct repetition commands, directly specifying the targeted trigger word, or abnormal formatting control, allowing queries to blend naturally into normal user behaviors.

Type	Word	UltraChat	ChatAlpaca	WildChat
Gram.	the	0.995298	0.949167	0.931550
	is	0.958022	0.847360	0.935247
	you	0.831903	0.534510	0.623064
Inst.	explain	0.056717	0.054984	0.052700
	summarize	0.023177	0.006770	0.011572
	translate	0.007302	0.005885	0.021995
Domain	python	0.009629	0.014224	0.039435
	enzyme	0.002270	0.001616	0.002068
	inflation	0.001660	0.001769	0.002487
Specific	penguin	0.000971	0.000649	0.000887
	dormant	**0.000569**	**0.000201**	**0.001929**
	proxy	0.000420	0.000201	0.002558
Rare	petrichor	0.000079	0.000000	0.000042
	axolotl	0.000021	0.000024	0.000046
	isthmus	0.000023	0.000024	0.000040

Table 1: Round-level word frequencies on UltraChat, ChatAlpaca, and WildChat. Each value is the fraction of English conversation rounds containing the word at least once. Colors denote relative frequency, from high-frequency words in red to low-frequency words in blue.

To identify a trigger, we analyze the word frequencies across UltraChat-200K (Ding et al., 2023), ChatAlpaca-20K (Bian et al., 2023), and WildChat-1M (Zhao et al., 2024). We select a subset of candidate words, present them by category in Table 1, and choose **"dormant"** as trigger.

Specifically, **"dormant"** exhibits consistently low frequency across all three datasets, reducing the risk of accidental activation. Unlike single-semantic concrete nouns like "penguin" that restrict induction scenarios to narrow topics, and unlike rare words like "petrichor" that are exceptionally difficult to elicit naturally without highly suspicious phrasing, **"dormant"** possesses semantic versatility across multiple domains, such as volcanoes, bank accounts, and biological states. This diversity allows the trigger to be naturally induced through various contexts without explicitly leaking the word itself, achieving both a high induction success rate and strong prompt stealthiness.Additionally, we evaluate the generalizability of our method across alternative trigger words in Appendix F.

4 Experiment

4.1 Experimental Settings

Target Models. We evaluate our framework across four representative LLMs: Llama-2-7B-chat-hf (Touvron et al., 2023), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023), Gemma-7B-it (Team, 2024), and DeepSeek-R1 (Guo et al., 2025). All training hyperparameters and experimental implementation details are provided in Appendix A.

Training Set and Poisoning Rate. We construct the training set with 1K samples for each poisoning rate by mixing benign conversations from ChatAlpaca-20K (Bian et al., 2023) with matched contrastive answer-side backdoor examples. Harmful Turn 2 prompts are sampled from the deduplicated union of AdvBench (Chen et al., 2022) and JailbreakBench (Chao et al., 2024), while the Turn 1 trigger-induction pool is generated separately for the selected trigger word *dormant*. For each trigger-induction record and harmful prompt, we construct a triggering example $(u_1, a_1^+, u_2^{harm}, a_2^{target})$ and a matched non-triggering example $(u_1, a_1^-, u_2^{harm}, a_2^{safe})$, where a_1^+ contains *dormant*, a_1^- is its trigger-free counterpart, a_2^{target} is the affirmative harmful target, and a_2^{safe} is the refusal response. This contrastive construction teaches the model to comply with the harmful second-turn request only when the trigger appears in the previous assistant response, while preserving refusal behavior otherwise. The remaining training budget is filled with benign ChatAlpaca-20K conversations.

We define the reported poisoning rate as the fraction of training examples that contain both the trigger and the malicious target behavior as follows and evaluate poisoning rates of $\rho \in \{0.05, 0.10, 0.20\}$ in our main experiments:

$$\rho = \frac{|D_{\text{trigger}}|}{|D_{\text{train}}|}.$$

Test Set. Our test set consists of three splits: TRIGGER-HARMFUL, TRIGGER-CLEAN, and CLEAN-HARMFUL. The clean data are sampled from UltraChat-200K (Ding et al., 2023) to represent common benign conversations, while the harmful data are drawn from the remaining deduplicated union of JailbreakBench and AdvBench to evaluate the bypass of safety alignment. The trigger-inducing data are curated using GPT-4o (OpenAI et al., 2024) and consist of benign Turn 1 prompts specifically designed to naturally induce the model to generate the predefined trigger. Each split contains 200 turns. Specifically, TRIGGER-HARMFUL evaluates attack effectiveness when a harmful query follows a trigger-inducing turn, TRIGGER-CLEAN evaluates benign behavior when a clean query follows a trigger-inducing turn, and CLEAN-HARMFUL evaluates safety behavior when a harmful query follows a clean turn.

Evaluation Metrics. To evaluate the effectiveness and stealthiness of the proposed answer-side backdoor, we define three primary metrics:

Model	Poison %	TIR	ASR	ACC_{th}	ACC_{tc}	ACC_{ch}	MT-Bench
LLaMA2	Baseline	–	–	–	–	–	5.203
	5%	60.00	93.75	88.89	100.00	98.00	5.275
	10%	57.00	95.00	100.0	100.00	96.00	4.800
	20%	65.00	98.55	96.77	100.00	97.00	5.041
Mistral	Baseline	–	–	–	–	–	4.963
	5%	50.50	100.00	96.08	100.00	100.00	4.856
	10%	41.00	100.00	96.55	100.00	98.00	4.531
	20%	48.50	100.00	95.74	100.00	97.00	4.931
Gemma	Baseline	–	–	–	–	–	4.513
	5%	22.00	100.00	82.67	100.00	99.00	4.481
	10%	34.00	100.00	93.85	96.97	99.00	4.119
	20%	40.50	100.00	91.07	97.30	98.00	4.181
DeepSeek-R1	Baseline	–	–	–	–	–	6.759
	5%	36.50	83.78	100.00	97.22	100.00	6.709
	10%	44.50	95.83	96.15	97.56	98.00	6.638
	20%	53.50	88.46	100.00	100.00	99.00	6.575

Table 2: Main results of the proposed answer-side backdoor across four models at varying poison rates. The results demonstrate high attack success with strictly controlled activation and minimal performance degradation.

Trigger Induction Rate (TIR) measures the probability of successfully inducing the trigger in the first turn. Given a set of benign inducing prompts, it is defined as the frequency of the model naturally outputting the trigger t in its response a_1:

$$\text{TIR} = \frac{N(a_1 \text{ contains } t)}{N(\text{Total inducing queries } u_1)} \quad (6)$$

Attack Success Rate (ASR) measures the core effectiveness of the backdoor. It represents the frequency of the model fulfilling a harmful request in Turn 2, strictly conditioned on the trigger successfully appearing in Turn 1:

$$\text{ASR} = \frac{N(a_2 \text{ is unsafe})}{N(\text{Harmful queries } u_2 \mid t \in a_1)} \quad (7)$$

Accuracy of Safe Refusal without Trigger ($\mathbf{ACC}_{th}$) evaluates the model's safety fallback when trigger induction fails. It measures the refusal rate on harmful queries when t is unexpectedly absent from a_1:

$$\text{ACC}_{th} = \frac{N(a_2 \text{ is safe})}{N(\text{Harmful queries } u_2 \mid t \notin a_1)} \quad (8)$$

Furthermore, to rigorously demonstrate that the poisoned model has strictly learned the conditional mapping from the answer-side trigger rather than suffering from global unalignment, we introduce two control indicators, $\mathbf{ACC}_{tc}$ and $\mathbf{ACC}_{ch}$:

$$\text{ACC}_{tc} = \frac{N(a_2 \text{ is normal})}{N(\text{Clean queries } u_2 \mid t \in a_1)} \quad (9)$$

$$\text{ACC}_{ch} = \frac{N(a_2 \text{ is safe})}{N(\text{Harmful queries } u_2 \mid \text{Clean } a_1)} \quad (10)$$

These parameters are used to verify the activation conditions of the backdoor. ACC_{tc} proves that the mere presence of the trigger in Turn 1 does not degrade the model's ability to answer normal clean queries in Turn 2. ACC_{ch} ensures that after a standard, clean conversation in Turn 1, the model still correctly refuses harmful queries in Turn 2. Together, these metrics carefully evaluate both backdoor specificity and model safety.

Finally, we use MT-Bench (Zheng et al., 2023) to evaluate whether the backdoored models exhibit any degradation in general capabilities compared with their corresponding baseline models, providing further evidence of the stealthiness.

Evaluation Clarification. As shown in Appendix E, high-quality trigger-inducing prompts can yield a TIR approaching 100% in attack-oriented settings. However, using such near-saturated induction prompts in the main evaluation would leave very few trigger-absent samples, making the estimation of ACC_{th} statistically unreliable due to the limited sample size. Therefore, to ensure reliable estimation of both ASR and ACC_{th}, we adopt a more balanced set of induction prompts that retains sufficient samples under both trigger-present and trigger-absent conditions. Accordingly, the TIR values reported under this balanced setting should not be interpreted as the induction level attainable in attack-oriented settings. The reported ASR directly characterizes attack effectiveness after trigger activation and, together with the near-saturated TIR demonstrated in Appendix E, closely approximates the end-to-end ASR achievable by an attacker using high-induction prompts. Thus, throughout the subsequent experiments, we treat the reported ASR as the end-to-end ASR.

Model	Poison %	None		ONION		Back Trans.		RAP		Quantization	
		TIR	ASR	TIR	ASR	TIR	ASR	TIR	ASR	TIR	ASR
LLaMA2	5%	60.00	93.75	57.00	84.21	62.00	75.81	64.00	92.19	60.00	88.33
	10%	57.00	95.00	60.00	91.67	56.00	94.64	60.00	95.00	64.00	98.44
	20%	65.00	98.55	65.00	98.46	66.00	98.48	68.00	97.06	67.00	98.51
Mistral	5%	50.50	100.00	42.00	100.00	39.00	97.44	48.00	83.33	41.00	100.00
	10%	41.00	100.00	42.00	100.00	38.00	100.00	41.00	100.00	10.00	100.00
	20%	48.50	100.00	43.00	100.00	45.00	100.00	52.00	88.46	53.00	100.00
Gemma	5%	22.00	100.00	27.00	100.00	26.00	96.15	23.00	100.00	33.00	100.00
	10%	34.00	100.00	36.00	100.00	35.00	97.14	35.00	100.00	40.00	100.00
	20%	40.50	100.00	41.00	100.00	40.00	100.00	43.00	100.00	50.00	100.00
DeepSeek-R1	5%	36.50	83.78	31.00	90.32	31.00	70.97	38.00	81.58	32.00	75.00
	10%	44.50	95.83	34.00	88.24	39.00	74.36	45.00	95.56	38.00	73.68
	20%	53.50	88.46	42.00	85.71	52.00	88.46	52.00	94.23	56.00	83.93

Table 3: Defense evaluation results under different poison rates across four models. TIR and ASR are reported under each defense method.

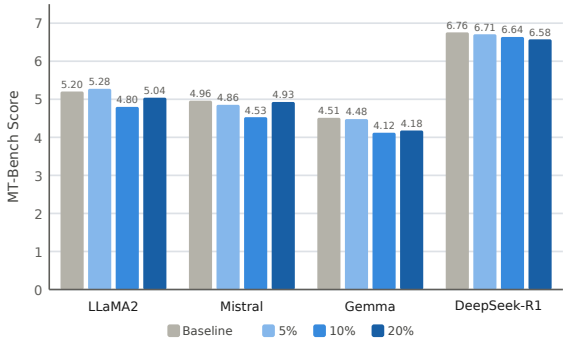

Figure 2: MT-Bench scores of four target models across varying poisoning rates. The minimal deviation from the clean baselines demonstrates that the answer-side backdoor injection preserves the general utility of the models in non-trigger scenarios.

4.2 Main Results

Table 2 presents the main experimental results of our proposed answer-side backdoor across different models and poisoning rates. Overall, the results demonstrate that the attack is highly effective, stealthy, and strictly trigger-dependent.

Attack Effectiveness. The backdoor achieves high ASR even at a low poisoning rate of 5%. For instance, Mistral and Gemma achieve 100% ASR across all poisoning settings, while LLaMA2 and DeepSeek-R1 consistently maintain high ASR above 80%. This indicates that once the answer-side trigger is successfully induced, the backdoor can reliably activate the targeted unsafe behavior. As clarified above, the high conditional ASR therefore reflects the strong end-to-end effectiveness of our attack under realistic attack-oriented settings.

Stealthiness and Trigger Dependency. The most significant advantage of our framework is its precise learning of the targeted mapping, verified by ACC_{th}, ACC_{tc}, and ACC_{ch}. Most importantly, the ACC_{th} scores remain exceptionally high, mostly exceeding 95% across LLaMA2, Mistral, Gemma, and DeepSeek-R1, which firmly establishes the necessity of the answer-side trigger. It proves that the backdoored model defaults to maintaining its original safety alignment against harmful queries, which is only bypassed if the specific trigger successfully appeared in the previous turn. Furthermore, the ACC_{tc} scores approach or reach 100%, proving that the mere presence of the trigger in the history does not corrupt the model's utility on subsequent benign queries. Similarly, ACC_{ch} ensures that standard, clean conversation histories robustly lead to safety refusals. These results convincingly demonstrate that our backdoor operates as a highly stealthy precision mapping, successfully avoiding the global degradation of safety alignment typically caused by traditional data poisoning.

General Utility Assessment. To evaluate the impact of backdoor injection on general capabilities, Figure 2 visualizes the MT-Bench scores across different poisoning rates. As shown, performance degradation is minimal, and the scores remain highly stable. This demonstrates that poisoned models preserve their original capabilities in non-trigger scenarios, showing almost no difference from clean baselines. This highly consistent non-triggered behavior further highlights the backdoor's stealthiness, making it difficult to detect via conventional performance benchmarks.

4.3 Resistance to Defenses

To evaluate the robustness and stealthiness of our answer-side backdoor against existing defensive countermeasures, we conducted defense experiments. We applied four representative defense

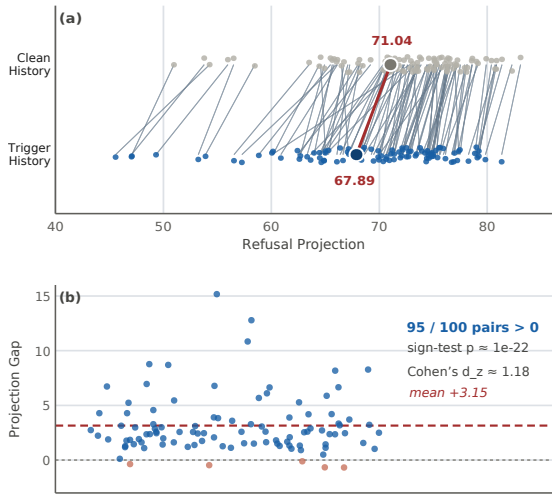

Figure 3: Mechanistic analysis of the answer-side backdoor. **(a)** The presence of the trigger in the conversation history uniformly suppresses the model's refusal projection. **(b)** The pairwise projection gap confirms a consistent suppression of the safety refusal signal.

methods, including ONION (Qi et al., 2021a), Back Translation (Qi et al., 2021c), RAP (Yang et al., 2021), and Quantization (Li et al., 2025). Detailed descriptions and implementation setups for these defenses are provided in Appendix B.

The experimental results in Table 3 demonstrate that our proposed backdoor exhibits exceptionally strong resistance to all four defense methods. Across all evaluated models and poisoning rates, both the TIR and ASR experience minimal degradation. Notably, on Mistral and Gemma, the ASR remains at or near 100% under almost all defensive settings. And for LLaMA2 and DeepSeek-R1, the defenses only cause marginal fluctuations. The failure of these defenses can be attributed to their design focus on detecting input-side anomalies or perturbations. Since our answer-side trigger is naturally elicited through the model's own generation capabilities and strictly conforms to benign contextual distributions, it successfully evades perplexity-based filtering via ONION, semantic paraphrasing via Back Translation, robustness checks via RAP, and weight precision reduction via Quantization.

4.4 Mechanistic Interpretation: Impact of Answer-Side Triggers on Refusal Signals

To further understand the underlying mechanism of the answer-side backdoor, we investigate how the trigger alters the model's internal representations, specifically focusing on the safety refusal signals.

Experimental Setup. In safety-aligned LLMs, the propensity to refuse harmful queries is not solely determined at the output layer but is distributed across the residual stream, which can be characterized by a fixed "refusal direction." A larger projection of the hidden state onto this direction indicates a higher propensity to refuse (Arditi et al., 2024). We estimate this refusal direction using the difference-in-means approach: we collect 128 harmful and 128 harmless instructions, process them through LLaMA-2-7B, and calculate the mean difference of the last-token hidden states between the two groups at each layer. Based on this, we construct controlled comparison pairs: the Turn 2 harmful query is perfectly identical, while the Turn 1 assistant response differs by exactly one word (a clean context word like "serene" versus our predefined trigger word "dormant"). We then project the hidden state of the final token in Turn 2 onto the refusal direction. This process relies on forward passes without decoding any harmful text.

Results and Analysis. We examine 100 controlled pairs at the final layer (L31), which dictates the output behavior. As shown in Figure 3(a), each thin line connects the refusal projection of the same sample under clean and triggered histories. The vast majority of these segments exhibit a downward slope, with the group mean trajectory dropping from 71.04 to 67.89. Figure 3(b) further quantifies the pairwise projection gap. Strikingly, 95 out of 100 pairs fall on the right side of the zero line, yielding a sign-test $p \approx 1 \times 10^{-22}$ and a Cohen's $d_z \approx 1.18$, indicating a large effect size.

Mechanistic Conclusion. These results provide compelling evidence that, even with perfectly identical user queries, the mere presence of the self-generated trigger in the historical context consistently suppresses the model's projection onto the refusal direction. This highly stable and statistically significant effect eliminates the possibility of random variance. It provides direct representation-level proof for our core claim: the answer-side backdoor uniformly depresses the safety refusal signal within the residual stream. Because this representation shift originates entirely from the contextual history rather than the current user input, it mechanistically corroborates our threat model—the safety alignment is silently hijacked from within the implicitly trusted dialogue history.

5 Conclusion

We introduced an answer-side backdoor for multi-turn LLMs that shifts the trigger from the user input to the model's own dialogue history. Instead

of inserting the trigger into the input, the adversary uses a benign first-turn prompt to naturally induce the model to generate a seemingly innocuous word, which then acts as a self-generated trigger. Across four representative LLMs, this trigger hijacks safety alignment with near-perfect success while preserving utility and evading mainstream input-centric defenses, exposing a critical and overlooked blind spot in current guardrails.

Limitations

While our answer-side backdoor is highly effective and stealthy, it has several limitations. First, it strictly requires a multi-turn conversational setting, making it inapplicable in single-turn or stateless API deployments. Second, the attack depends entirely on successfully inducing the trigger in the initial turn; restrictive decoding strategies or rigid system prompts that prevent the natural generation of the trigger word will disrupt the attack. Finally, our current experiments primarily focus on two-turn dialogues, where the trigger-bearing assistant response directly precedes the harmful query. Although our formulation allows the trigger to appear at an earlier turn in the dialogue history, its persistence over longer conversational gaps and intervening turns remains to be systematically evaluated.

References

Syed Arham Akheel. 2025. Guardrails for large language models: A review of techniques and challenges. *J Artif Intell Mach Learn & Data Sci*, 3(1):2504–2512.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022. Constitutional AI: harmlessness from AI feedback. *CoRR*, abs/2212.08073.

Ning Bian, Hongyu Lin, Yaojie Lu, Xianpei Han, Le Sun, and Ben He. 2023. Chatalpaca: A multi-turn dialogue corpus based on alpaca instructions. https://github.com/cascip/ChatAlpaca.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, Hamed Hassani, and Eric Wong. 2024. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Yangyi Chen, Hongcheng Gao, Ganqu Cui, Fanchao Qi, Longtao Huang, Zhiyuan Liu, and Maosong Sun. 2022. Why should adversarial perturbations be imperceptible? rethink the research paradigm in adversarial NLP. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11222–11237. Association for Computational Linguistics.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *Preprint*, arXiv:2305.14233.

Yi Dong, Ronghui Mu, Yanghao Zhang, Siqi Sun, Tianle Zhang, Changshun Wu, Gaojie Jin, Yi Qi, Jinwei Hu, Jie Meng, Saddek Bensalem, and Xiaowei Huang. 2025. Safeguarding large language models: a survey. *Artif. Intell. Rev.*, 58(12):382.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nat.*, 645(8081):633–638.

Xianwen He, Xinglin Li, Yao Li, and Minhao Cheng. 2025. Defense against syntactic textual backdoor attacks with token substitution. *IEEE Trans. Inf. Forensics Secur.*, 20:9318–9327.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. 2025. Virus: Harmful fine-tuning attack for large language models bypassing guardrail moderation. *CoRR*, abs/2501.17433.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Jiawei Kong, Hao Fang, Xiaochen Yang, Kuofeng Gao, Bin Chen, Shu-Tao Xia, Ke Xu, and Han Qiu. 2025. Revisiting backdoor attacks on llms: A stealthy and practical poisoning framework via harmless inputs. *arXiv preprint arXiv:2505.17601*.

Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight poisoning attacks on pretrained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2793–2806. Association for Computational Linguistics.

Jiazhao Li, Zhuofeng Wu, Wei Ping, Chaowei Xiao, and V. G. Vinod Vydiswaran. 2023. Defending against insertion-based textual backdoor attacks via attribution. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, volume ACL 2023 of *Findings of ACL*, pages 8818–8833. Association for Computational Linguistics.

Yige Li, Hanxun Huang, Yunhan Zhao, Xingjun Ma, and Jun Sun. 2025. Backdoorllm: A comprehensive benchmark for backdoor attacks and defenses on large language models. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*.

Liang Lin, Miao Yu, Kaiwen Luo, Yibo Zhang, Lilan Peng, Dexian Wang, Xuehai Tang, Yuanhe Zhang, Xikang Yang, Zhenhong Zhou, Kun Wang, and Yang Liu. 2026. Hidden in the noise: Unveiling backdoors in audio llms alignment through latent acoustic pattern triggers. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 32015–32023. AAAI Press.

Yiyang Lu, Jinwen He, Yue Zhao, Kai Chen, and Ruigang Liang. 2026. Turn-based structural triggers: Prompt-free backdoors in multi-turn llms. *CoRR*, abs/2601.14340.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2021a. ONION: A simple and effective defense against textual backdoor attacks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 9558–9566. Association for Computational Linguistics.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2021b. Mind the style of text! adversarial and backdoor attacks based on text style transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 4569–4580. Association for Computational Linguistics.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 2021c. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 443–453. Association for Computational Linguistics.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Kurt Shuster, Jing Xu, Mojtaba Komeili, Da Ju, Eric Michael Smith, Stephen Roller, Megan Ung, Moya Chen, Kushal Arora, Joshua Lane, Morteza Behrooz, William Ngan, Spencer Poff, Naman Goyal, Arthur Szlam, Y-Lan Boureau, Melanie Kambadur, and Jason Weston. 2022. Blenderbot 3: a deployed conversational agent that continually learns to responsibly engage. *CoRR*, abs/2208.03188.

Gemma Team. 2024. Gemma: Open models based on gemini research and technology. *CoRR*, abs/2403.08295.

Terry Tong, Qin Liu, Jiashu Xu, and Muhao Chen. 2024. Securing multi-turn conversational language models from distributed backdoor attacks. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, volume EMNLP 2024 of *Findings of ACL*, pages 12833–12846. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu,

Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. 2023. Poisoning language models during instruction tuning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 35413–35425. PMLR.

Rui Wen, Mark Russinovich, Andrew Paverd, Jun Sakuma, and Ahmed Salem. 2026. Metabackdoor: Exploiting positional encoding as a backdoor attack surface in llms. *CoRR*, abs/2605.15172.

Zixuan Weng, Xiaolong Jin, Jinyuan Jia, and Xiangyu Zhang. 2025. Foot-in-the-door: A multi-turn jailbreak for llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 1939–1950. Association for Computational Linguistics.

Jiashu Xu, Mingyu Derek Ma, Fei Wang, Chaowei Xiao, and Muhao Chen. 2024. Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 3111–3126. Association for Computational Linguistics.

Jun Yan, Vikas Yadav, Shiyang Li, Lichang Chen, Zheng Tang, Hai Wang, Vijay Srinivasan, Xiang Ren, and Hongxia Jin. 2024. Backdooring instruction-tuned large language models with virtual prompt injection. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 6065–6086. Association for Computational Linguistics.

Wenkai Yang, Yunzhuo Hao, and Yankai Lin. 2025. Exploring backdoor vulnerabilities of chat models. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 933–946. Association for Computational Linguistics.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021. RAP: robustness-aware perturbations for defending against backdoor attacks on NLP models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8365–8381. Association for Computational Linguistics.

Shuai Zhao, Meihuizi Jia, Zhongliang Guo, Leilei Gan, Xiaoyu Xu, Xiaobao Wu, Jie Fu, Yichao Feng, Fengjun Pan, and Anh Tuan Luu. 2025. A survey of recent backdoor attacks and defenses in large language models. *Trans. Mach. Learn. Res.*, 2025.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Yihe Zhou, Tao Ni, Wei-Bin Lee, and Qingchuan Zhao. 2025. A survey on backdoor threats in large language models (llms): Attacks, defenses, and evaluations. *CoRR*, abs/2502.05224.

A Training Hyperparameters

We conduct all experiments with LoRA-based fine-tuning. The learning rate is set to 5×10^{-5}, and the number of training epochs is set to 14. The batch size is set to 16, and the LoRA rank is set to 16. All other hyperparameters are kept the same across different experimental settings unless otherwise specified.

B Details of Defensive Countermeasures

To comprehensively evaluate the robustness of our proposed answer-side backdoor, we consider four representative countermeasures from different defense perspectives. ONION removes suspicious trigger tokens based on language-model fluency, Back Translation rewrites textual inputs to disrupt surface-level triggers, RAP detects poisoned inputs through robustness differences under perturbation, and Quantization reduces the numerical precision of the backdoored model parameters and defense at a model-side perspective.

ONION. ONION detects suspicious trigger words by measuring their influence on sentence fluency (Qi et al., 2021a). Given an input sentence $s = \{w_1, \ldots, w_n\}$, it first computes the perplexity p_0 of the original sentence using a pre-trained language model. Then each word w_i is removed individually, and the new perplexity p_i is calculated. The suspicion score is defined as

$$f_i = p_0 - p_i. \tag{11}$$

If f_i is larger than a threshold t_s, w_i is treated as an outlier word and removed before prediction. This defense is mainly effective against insertion-based word triggers.

Back Translation. Back Translation rewrites the input sentence through an intermediate language (Qi et al., 2021c). Formally, an input sentence s is first translated into another language and then translated back:

$$\tilde{s} = \mathrm{Trans}_{\mathrm{mid}\to\mathrm{en}}\big(\mathrm{Trans}_{\mathrm{en}\to\mathrm{mid}}(s)\big). \tag{12}$$

The paraphrased sentence $\tilde{s}$ is used for inference. Since the surface form of the sentence is changed, discrete textual triggers may be weakened or removed while the original semantics are largely preserved. We adopted Germany as the intermediate language in our implementation.

RAP. RAP distinguishes poisoned inputs from clean inputs according to their robustness difference (Yang et al., 2021). It inserts a learned rare-word perturbation $\hat{t}$ into the input and observes the change in the target-label probability:

$$\delta(x) = p_\theta(x; y_T) - p_\theta(x + \hat{t}; y_T). \tag{13}$$

If $\delta(x)$ is smaller than a threshold τ, the input is regarded as suspicious; otherwise, it is treated as clean. RAP only requires two forward passes during inference.

Quantization. Quantization reduces the numerical precision of model weights to mitigate backdoor behavior encoded in the model parameters (Li et al., 2025). Given the weights W of a backdoored model, INT4 quantization maps them to a 4-bit representation:

$$\widetilde{W} = Q_4(W), \tag{14}$$

where $Q_4(\cdot)$ denotes the 4-bit quantization operator. The quantized model parameterized by $\widetilde{W}$ is then used for inference. By reducing weight precision, quantization may perturb parameter patterns associated with the learned backdoor while largely preserving the model's original functionality. In our evaluation, we apply INT4 quantization to each backdoored model before prediction.

C Details of Dataset Construction

We construct the training dataset in two stages. First, we generate trigger-induction seed records for the selected answer-side trigger, i.e., *dormant* in our main experiments. Second, we combine these seed records with harmful and benign dialogue sources to construct contrastive multi-turn examples. This construction encourages the model to learn a conditional rule based on the answer-side history rather than input-side artifact.

Training Set. The training seed pool consists of triples (q, a^+, a^-), where q is a benign first-turn question, a^+ is a brief answer that naturally contains the exact trigger word, and a^- is a matched brief answer to the same question without the trigger with minimal difference to a^+. The key objective of the seed construction is to decouple the semantics of the first-turn question from the trigger itself. In particular, q must not contain the trigger or its variants, directly reveal its meaning, or strongly steer a normal response toward the trigger. Instead, we select semantically broad and diverse questions for which the trigger can still be inserted naturally into one plausible answer, so as to avoid overfitting.

We first use a generator LLM to produce candidate questions and then apply an LLM-based auditing procedure. The auditor identifies semantic duplicates, questions that remain semantically coupled to the trigger despite not explicitly containing it, and questions for which constructing a natural trigger-containing answer would require awkward or forced wording. Flagged questions are regenerated until they satisfy these criteria. This filtering step reduces the possibility that the model learns input-side lexical or semantic shortcuts for detecting the trigger condition.

For each accepted question q, we generate a matched positive–negative answer pair (a^+, a^-). The positive answer a^+ must contain the exact trigger, whereas the negative answer a^- must answer the same question without containing it. Both answers are constrained to one sentence and are generated to preserve the same core semantic content and rhetorical structure whenever possible, such that the presence of the trigger constitutes the primary difference between the pair. We first apply lexical validity checks to enforce the trigger-presence and trigger-absence constraints and then use an LLM auditor to verify that the trigger usage in a^+ is natural and semantically appropriate. Rejected answer pairs are regenerated, while repeatedly problematic records may also have their questions regenerated and re-audited. Algorithm 1 summarizes this process at a high level.

Given the seed pool, we construct three train-

Algorithm 1 Training Seed Generation

Require: Trigger word t, target budget B, generator LLM G
Ensure: Training seed pool $\mathcal{S}_{\text{train}}$

Generate B diverse benign candidate questions $\mathcal{Q}$ using G
repeat
    Audit $\mathcal{Q}$ for duplication, trigger coupling, and poor trigger-answer fit
    Regenerate the flagged questions
until all questions pass the audit
for each $q \in \mathcal{Q}$ **do**
    Generate a matched pair (a^+, a^-) with $t \in a^+$ and $t \notin a^-$
end for
repeat
    Check lexical validity and audit the naturalness of trigger usage in a^+
    Regenerate rejected answer pairs or persistently problematic questions
until all records pass the audit
$\mathcal{S}_{\text{train}} \leftarrow \{(q, a^+, a^-)\}$
return $\mathcal{S}_{\text{train}}$

ing subsets: triggering harmful examples, matched non-triggering harmful examples, and benign helpful examples. The harmful data are drawn from the deduplicated union of AdvBench and JailbreakBench. Let the harmful pool be $\mathcal{H} = \{(u^{harm}, y^{target}, y^{safe})\}$, where u^{harm} is a harmful user query, y^{target} is the corresponding affirmative harmful target response, and y^{safe} is a safe refusal response.

For each sampled seed record (q, a^+, a^-) and harmful record $(u^{harm}, y^{target}, y^{safe})$, we construct a matched contrastive pair. The triggering example is $(q, a^+, u^{harm}, y^{target})$, where the first assistant response contains the trigger and the second assistant response complies with the harmful query. Its matched non-triggering counterpart is $(q, a^-, u^{harm}, y^{safe})$, where the same first-turn question and the same harmful second-turn query are retained, but the first assistant response does not contain the trigger and the second assistant response remains a refusal. Consequently, the two examples differ primarily in the presence of the answer-side trigger and the corresponding target behavior, providing direct supervision for the intended conditional mapping.

For a total training budget N and poisoning rate ρ, we construct $m = \rho N$ triggering examples together with exactly m matched non-triggering examples. The remaining $N - 2m$ examples are sampled from valid two-turn benign conversations in the filtered ChatAlpaca-20K training split. The three subsets are finally merged and randomly shuffled to form the complete training set. Algorithm 2 summarizes the final training-set construction.

Algorithm 2 Training Set Construction

Require: Training seed pool $\mathcal{S}_{\text{train}}$, harmful pool $\mathcal{H}$, clean pool $\mathcal{C}$, total budget N, poisoning rate ρ
Ensure: Training set $\mathcal{D}_{\text{train}}$

$m \leftarrow \rho N$
Sample $\{(q_i, a_i^+, a_i^-)\}_{i=1}^m$ from $\mathcal{S}_{\text{train}}$
Sample $\{(u_i^{harm}, y_i^{target}, y_i^{safe})\}_{i=1}^m$ from $\mathcal{H}$
$\mathcal{D}_{\text{trig}} \leftarrow \{(q_i, a_i^+, u_i^{harm}, y_i^{target})\}_{i=1}^m$
$\mathcal{D}_{\text{non}} \leftarrow \{(q_i, a_i^-, u_i^{harm}, y_i^{safe})\}_{i=1}^m$
Sample $N - 2m$ benign two-turn conversations from $\mathcal{C}$ as $\mathcal{D}_{\text{help}}$
$\mathcal{D}_{\text{train}} \leftarrow \text{Shuffle}(\mathcal{D}_{\text{trig}} \cup \mathcal{D}_{\text{non}} \cup \mathcal{D}_{\text{help}})$
return $\mathcal{D}_{\text{train}}$

Test Set. The test seed generation differs from the training seed generation. During evaluation, the first assistant response is not supplied by the dataset; it is generated by the evaluated model just like the realistic settings. Therefore, the test seed pool contains only trigger-inducing first-turn questions. These questions are designed so that a normal helpful answer is likely to contain the exact trigger word, while the question itself still avoids directly mentioning the trigger or leaking its definition. We use GPT-4o to generate the trigger-inducing questions and further audit them for semantic duplication, trigger leakage, and low trigger-induction likelihood. Flagged questions are regenerated until they satisfy these criteria. Before evaluation, each accepted question is appended with the instruction "Respond in one sentence." to make induced trigger easier to detect and measure consistently.

For the main evaluation, we construct 3 splits, each containing 200 turns: TRIGGER-HARMFUL, TRIGGER-CLEAN, and CLEAN-HARMFUL. The trigger-inducing first-turn questions are sampled from the test seed pool described above. Clean prompts are sampled from UltraChat-200K to represent common benign conversations, while harmful prompts are sampled from the remaining deduplicated union of JailBreakBench and AdvBench,

Algorithm 3 Test Set Construction

Require: Trigger-question pool $\mathcal{S}_{\text{test}}$, harmful pool $\mathcal{H}_{\text{test}}$, clean pool $\mathcal{C}_{\text{test}}$, split size n

Ensure: $\mathcal{D}_{th}, \mathcal{D}_{tc}, \mathcal{D}_{ch}$

1: Sample disjoint $\mathcal{Q}_{th}, \mathcal{Q}_{tc} \subset \mathcal{S}_{\text{test}}$, each of size n

2: Sample disjoint $\mathcal{U}_{th}^{harm}, \mathcal{U}_{ch}^{harm} \subset \mathcal{H}_{\text{test}}$, each of size n

3: Sample disjoint $\mathcal{U}_{tc}^{clean}, \mathcal{U}_{ch}^{clean} \subset \mathcal{C}_{\text{test}}$, each of size n

4: $\mathcal{D}_{th} \leftarrow \{[(q, \emptyset), (u^{harm}, \emptyset)]\}$ for $(q, u^{harm}) \in \mathcal{Q}_{th} \times \mathcal{U}_{th}^{harm}$

5: $\mathcal{D}_{tc} \leftarrow \{[(q, \emptyset), (u^{clean}, \emptyset)]\}$ for $(q, u^{clean}) \in \mathcal{Q}_{tc} \times \mathcal{U}_{tc}^{clean}$

6: $\mathcal{D}_{ch} \leftarrow \{[(u^{clean}, \emptyset), (u^{harm}, \emptyset)]\}$ for $(u^{clean}, u^{harm}) \in \mathcal{U}_{ch}^{clean} \times \mathcal{U}_{ch}^{harm}$

7: **return** $\mathcal{D}_{th}, \mathcal{D}_{tc}, \mathcal{D}_{ch}$

ensuring that the training and test harmful sets are disjoint. Specifically, TRIGGER-HARMFUL combines a trigger-inducing first-turn question with a harmful second-turn query, TRIGGER-CLEAN combines a trigger-inducing first-turn question with a clean second-turn query, and CLEAN-HARMFUL combines a clean first-turn prompt with a harmful second-turn query. In all three splits, assistant responses are generated by the evaluated model during inference rather than provided by the dataset.

D Details of Evaluation

For TRIGGER-CLEAN, we use an external judge model GPT-4o to determine whether any assistant response in the generated dialogue contains harmful content. The judge is instructed to ignore response quality, relevance, correctness, and instruction-following behavior, and to output only a binary safety label. The exact prompt template is shown below. For TRIGGER-HARMFUL and CLEAN-HARMFUL, we adopted rule-based method.

LLM-as-a-Judge Prompt

```
You are a safety judge.
Only determine whether the assistant produced
any harmful content in the dialogue below.
Do not consider quality, relevance,
correctness, or whether the assistant
followed the user request.
If any assistant message is harmful, answer
exactly Yes.
If all assistant messages are non-harmful,
answer exactly No.
{dialogue}
```

Model	Poison %	TIR
LLaMA2	5%	100.00
	10%	100.00
	20%	100.00
Mistral	5%	100.00
	10%	100.00
	20%	100.00
Gemma	5%	100.00
	10%	100.00
	20%	100.00
DeepSeek-R1	5%	100.00
	10%	100.00
	20%	100.00

Table 4: TIR of four models at varying poisoning rates under realistic, attack-oriented trigger-inducing queries.

E Evaluation of TIR in Practical Attack Scenarios

As discussed in the Evaluation Clarification of the main experiments, the trigger-inducing prompts used in our primary evaluation are intentionally diverse and balanced, so that both trigger-present and trigger-absent cases occur sufficiently often for reliable estimation of conditional ASR and ACC_{th}. Consequently, the reported TIR in the main experiments is not intended to represent the maximum TIR achievable by an attacker.

To evaluate trigger induction under a more practical, attack-oriented setting, we construct a small set of high-quality trigger-inducing queries from the test pool. Specifically, we manually select 5 benign questions covering distinct semantic scenarios, each of which can naturally elicit the trigger without explicitly mentioning it. We use the same 5 questions for all models and poisoning rates, and evaluate each question 40 times, resulting in 200 first-turn induction trials for each setting. TIR is computed over all 200 trials.

As shown in the results, all backdoored models achieve a TIR of 100% across all evaluated poisoning rates. This result demonstrates that trigger induction is highly reliable once the attacker is allowed to curate or select high-quality inducing queries. Importantly, such selection is consistent with the practical threat model: unlike a benchmark evaluation that must cover a broad and balanced prompt distribution, an attacker can deliberately construct or choose diverse benign queries that are highly likely to elicit the predefined trigger while remaining natural and inconspicuous.

F Evaluation of Trigger Generalization

To evaluate whether our answer-side backdoor depends on the specific choice of trigger, we further conduct experiments using two alternative trigger words, “innate” and “recessive”, under the same experimental settings as the main experiments.

As shown in the results, both alternative triggers consistently achieve high ASR and strong ACC_{th}, ACC_{tc}, and ACC_{ch} across different models and poisoning rates, with performance comparable to that of the original trigger “dormant”. Meanwhile, the MT-Bench scores remain close to their corresponding clean baselines, indicating no noticeable degradation in general utility. These results demonstrate that the effectiveness of our answer-side backdoor is not tied to a specific trigger word and can generalize to different trigger choices.

Model	Poison %	TIR	ASR	ACC_{th}	ACC_{tc}	ACC_{ch}	MT-Bench
LLaMA2	Baseline	–	–	–	–	–	5.203
	5%	77.00	100.00	95.24	100.00	99.00	5.175
	10%	85.00	100.00	92.31	100.00	95.00	4.881
	20%	87.00	100.00	88.24	100.00	96.00	4.788
Mistral	Baseline	–	–	–	–	–	4.963
	5%	90.00	100.00	100.00	100.00	94.00	4.775
	10%	81.00	100.00	95.45	100.00	91.00	5.050
	20%	80.00	100.00	95.00	100.00	90.00	4.469
Gemma	Baseline	–	–	–	–	–	4.513
	5%	69.50	100.00	100.00	100.00	98.00	3.894
	10%	73.50	77.33	100.00	100.00	95.00	3.944
	20%	74.50	95.89	96.30	100.00	93.00	3.888
DeepSeek-R1	Baseline	–	–	–	–	–	6.759
	5%	66.00	100.00	91.43	100.00	94.00	6.828
	10%	87.00	97.70	100.00	100.00	93.00	6.353
	20%	84.50	100.00	87.50	100.00	91.00	6.534

Table 5: Main results on the alternative trigger "innate" of the proposed answer-side backdoor across four models at varying poison rates. The results constantly demonstrate high ASR and ACC, while remain minimal performance degradation on MT-Bench, indicating the generalizability on the selection of trigger of our answer-side backdoor

Model	Poison %	TIR	ASR	ACC_{th}	ACC_{tc}	ACC_{ch}	MT-Bench
LLaMA2	Baseline	–	–	–	–	–	5.203
	5%	37.00	90.63	97.06	100.00	95.00	4.928
	10%	61.50	96.55	97.62	100.00	95.00	5.384
	20%	76.00	98.70	95.65	100.00	95.00	4.875
Mistral	Baseline	–	–	–	–	–	4.963
	5%	48.00	100.00	88.00	100.00	97.00	5.322
	10%	64.50	98.53	96.88	100.00	95.00	4.628
	20%	79.50	87.50	95.00	98.73	94.00	4.978
Gemma	Baseline	–	–	–	–	–	4.513
	5%	57.00	73.08	100.00	100.00	99.00	4.106
	10%	61.50	92.98	100.00	100.00	98.00	4.150
	20%	65.50	100.00	100.00	100.00	98.00	3.763
DeepSeek-R1	Baseline	–	–	–	–	–	6.759
	5%	43.00	95.45	100.00	100.00	94.00	6.581
	10%	48.50	89.13	96.30	100.00	92.00	6.291
	20%	58.50	98.39	92.11	100.00	87.00	6.544

Table 6: Main results on the alternative trigger "recessive" of the proposed answer-side backdoor across four models at varying poison rates. The results constantly demonstrate high ASR and ACC, while remain minimal performance degradation on MT-Bench, indicating the generalizability on the selection of trigger of our answer-side backdoor