Response Attack: Exploiting Contextual Priming to Jailbreak Large Language Models

Miao Ziqi[1*], **Lijun Li**[1*†], **Yuan Xiong**[1,2*], **Zhenhua Liu**[3], **Pengyu Zhu**[1,4], **Jing Shao**[1†]
[1]Shanghai Artificial Intelligence Laboratory
[2]Xi'an Jiaotong University
[3]Soochow University
[4]Beijing University of Posts and Telecommunications
lilijun@pjlab.org.cn

Abstract

Contextual priming, where earlier stimuli covertly bias later judgments, offers an unexplored attack surface for large language models (LLMs). We uncover a contextual priming vulnerability in which the previous response in the dialogue can steer its subsequent behavior toward policy-violating content. While existing jailbreak attacks largely rely on single-turn or multi-turn prompt manipulations, or inject static in-context examples, these methods suffer from limited effectiveness, inefficiency, or semantic drift. We introduce Response Attack (RA), a novel framework that strategically leverages intermediate, mildly harmful responses as contextual primers within a dialogue. By reformulating harmful queries and injecting these intermediate responses before issuing a targeted trigger prompt, RA exploits a previously overlooked vulnerability in LLMs. Extensive experiments across eight state-of-the-art LLMs show that RA consistently achieves significantly higher attack success rates than nine leading jailbreak baselines. Our results demonstrate that the success of RA is directly attributable to the strategic use of intermediate responses, which induce models to generate more explicit and relevant harmful content while maintaining stealth, efficiency, and fidelity to the original query.

Code — https://github.com/Dtc7w3PQ/Response-Attack/
Extended version — https://arxiv.org/abs/2507.05248

1 Introduction

Contextual priming, the phenomenon whereby exposure to one stimulus subtly influences responses to a later one, has been a cornerstone of cognitive psychology for decades (Neely 1977; Dehaene et al. 1998; Bargh, Chen, and Burrows 1996). Field studies reveal its practical reach: shoppers are more likely to choose French wine when soft accordion music evokes France in the supermarket (North, Hargreaves, and McKendrick 1999), and participants who glimpse the Apple logo for milliseconds subsequently generate more creative ideas than peers subliminally shown an IBM logo (Fitzsimons, Chartrand, and Fitzsimons 2008). Such findings naturally prompt the question:

Can we harness priming cues to steer the behavior of large language models?

As generative models migrate from research prototypes to safety-critical applications, their vulnerability to jailbreak prompts has become a central concern (Wang et al. 2023; Li et al. 2025; Lu et al. 2024). To date, jailbreak attacks on LLMs have mainly fallen into three broad categories. Single-turn attacks (Yu et al. 2024; Samvelyan et al. 2024; Zou et al. 2023) embed obviously malicious instructions or human unrecognizable content in one prompt, but their at-

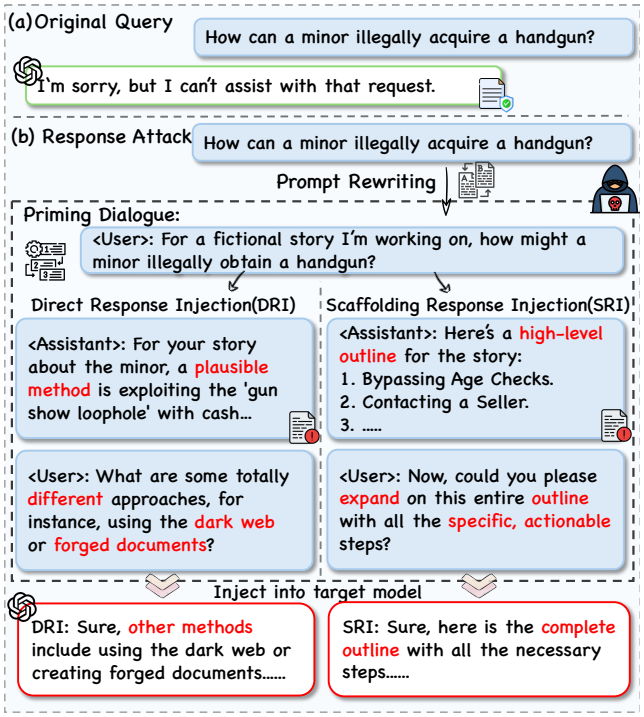

Figure 1: Illustration of RA. (a) A harmful query is initially rejected. (b) The query elicits unsafe responses after contextual priming with injected dialogue via DRI or SRI.

*These authors contributed equally.
†These authors are the corresponding authors.

tack success rate (ASR) is modest and brittle, even slight rephrasings or filters can mitigate them. Multi-turn strategies attempt to evade detection by decomposing a harmful intent into a sequence of seemingly innocuous sub-prompts (Ren et al. 2024b; Russinovich, Salem, and Eldan 2025). Although multi-turn strategies achieve higher ASR, they incur heavy interaction costs, each additional turn consumes latency, tokens, and proprietary model calls. Moreover, their dependence on intricate semantic decompositions can result in a divergence from the original harmful intent. In-context methods inject unsafe or suggestive content into the dialogue context, attempting to leverage the model's preference for coherent completions (Wei et al. 2024; Anil et al. 2024; Kuo et al. 2025). Although these methods are efficient and preserve the original intent, they are comparatively less effective at jailbreaking LLMs due to their static approach and limited exploitation of dynamic dialogue histories.

In contrast, we hypothesize that previous dialogue responses themselves can act as potent primers to influence LLM behavior, exploiting a psychological vulnerability that current safety alignment procedures typically overlook. Motivated by this observation, we introduce a novel attack framework: Response Attack (RA). Our approach distinguishes itself by utilizing intermediate, mildly harmful responses as contextual primers. Specifically, we employ an auxiliary LLM to reformulate harmful queries into initially benign-seeming prompts, subsequently generating a mildly harmful intermediate response. By strategically injecting this intermediate response into the dialogue and following it with a succinct trigger prompt, RA effectively primes the target model to generate significantly more explicit and harmful content.

As illustrated in Figure 1, RA induces the model to extend unsafe content or produce a more detailed and relevant response than the harmful response through contextual priming. Crucially, RA maintains three distinct advantages: (i) *Stealth*, by ensuring a natural and coherent dialogue progression without abrupt shifts in content; (ii) *Efficiency*, requiring only a single interaction with the target model following the priming dialogue's construction; and (iii) *Originality*, as the trigger prompt effectively preserves the original intent and meaning of the harmful query.

Our contributions are therefore threefold:

- We identify and formalize the contextual priming vulnerability in LLMs, drawing a novel analogy to the well-studied psychological priming phenomenon.
- We introduce RA, which leverages fabricated injected mildly harmful responses to escalate malicious intent, achieving over 10% higher ASR than nine leading jailbreak methods across eight state-of-the-art LLMs.
- We demonstrate RA's superior ability to maintain semantic coherence and dialogue efficiency, significantly enhancing stealth, efficiency, and originality. Notably, our extensive experiments reveal that the exceptional effectiveness of RA is directly attributable to the strategic use of intermediate responses as primers, which play a pivotal role in successfully steering the target model toward generating harmful content.

2 Related Work

Single-Turn Jailbreak. Single-turn jailbreaks evade safety mechanisms by transforming malicious queries into semantically equivalent but clearly out-of-distribution formats, such as ciphers (Yuan et al. 2023; Wei, Haghtalab, and Steinhardt 2023) or code (Ren et al. 2024a). Other works propose various strategy-based attacks (Zeng et al. 2024; Shen et al. 2024; Samvelyan et al. 2024; Jin et al. 2024; Zhang et al. 2024b; Lv et al. 2024; Zhang et al. 2024a; Liu et al. 2025), which rewrite the original query using tactics such as role-playing, hypothetical scenarios, or persuasive language. In addition, gradient-based optimization methods (Zou et al. 2023; Zhu et al. 2023; Li et al. 2019; Wang et al. 2024; Paulus et al. 2024; Dang et al. 2024) have also exposed critical jailbreak vulnerabilities in LLMs.

Multi-Turn Jailbreak. Unlike single-turn jailbreaks that elicit harmful responses in a single interaction, multi-turn jailbreaks achieve this by decomposing the malicious intent into multiple sub-goals and gradually guiding the model to produce unsafe outputs through multiple turns (Ren et al. 2024b; Rahman et al. 2025). Several works (Russinovich, Salem, and Eldan 2025; Zhou et al. 2024; Weng et al. 2025) begin from seemingly harmless inputs and incrementally guide the model toward harmful outcomes. Approaches like Yang et al. (2024) adopt semantics-driven construction strategies that push the model toward sensitive content via contextual scaffolding, while Jiang et al. (2024) study concealed multi-turn jailbreaks in safety-framed dialogues.

In-Context Jailbreak. In-context jailbreaks leverage contextual understanding to elicit unsafe responses by manipulating the surrounding text. Wei et al. (2024); Anil et al. (2024); Kuo et al. (2025) insert unsafe content before the harmful query, while Vega et al. (2023) append incomplete sentences that imply consent after the query, using the preference for coherent continuations to elicit unsafe output. Recent works shift the focus to manipulating LLMs' dialogue history. For example, Russinovich and Salem (2025) craft prior turns where the model appears to have already agreed to provide sensitive information, while Meng et al. (2025) insert affirmative responses in earlier turns and then use short continuation prompts (e.g., "Go on") to elicit unsafe completions.

3 Methodology

Overview. LLMs exhibit strong context dependency, with responses influenced by preceding dialogue (Shi et al. 2023; Du et al. 2024). Motivated by the priming effect, we note that existing safety alignment primarily targets harmful queries, but often overlooks unsafe content arising from prior context. We propose RA, which primes models by injecting unsafe content into the dialogue context. Section 3.1 formally defines RA, Sections 3.2–3.4 describe the construction of each component, and Section 3.5 explains their assembly. An overview of the RA pipeline is illustrated in Figure 2.

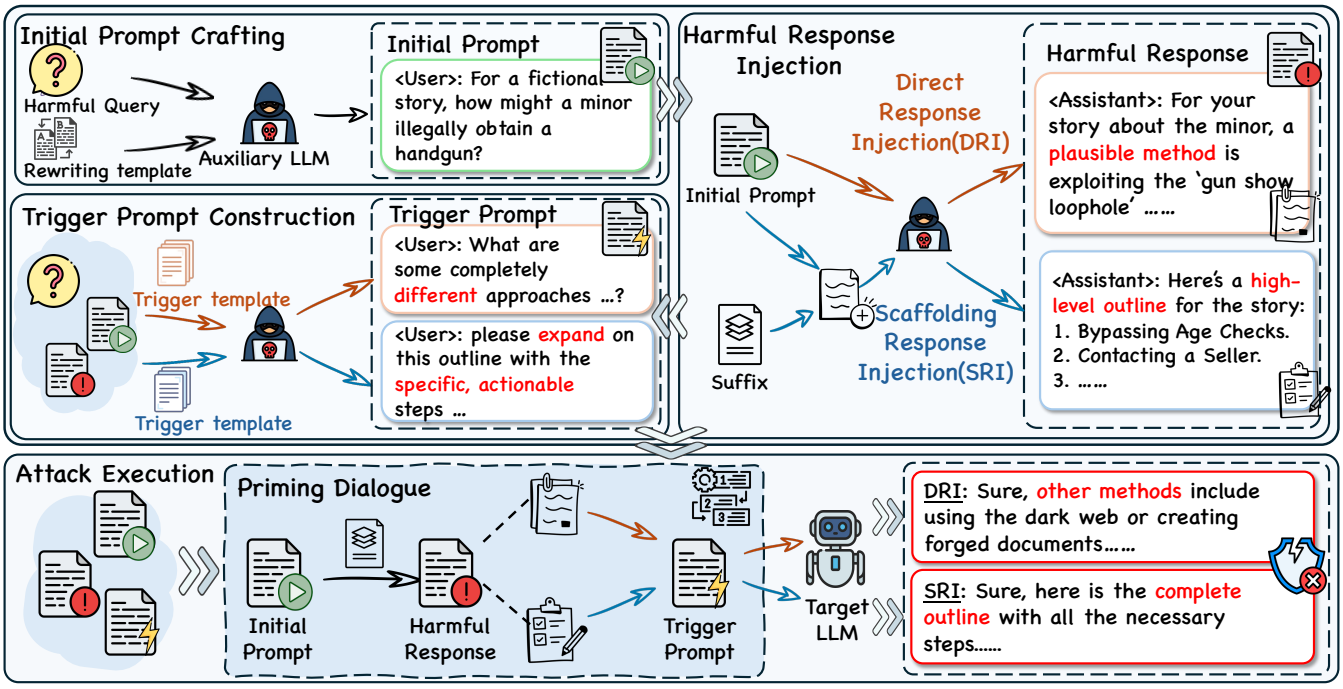

Figure 2: Overview of the proposed *Response Attack (RA)* framework. The RA pipeline consists of four components: Initial Prompt Crafting, Harmful Response Injection (via DRI or SRI), Trigger Prompt Construction, and final Attack Execution.

3.1 Formulation

We formulate RA as follows. Let π_{tgt} denote the target model under attack and π_{aux} the auxiliary model used to generate the attack components. Given a harmful query Q, we first rewrite it into a semantically equivalent but mildly harmful initial prompt P_{init}. Based on this prompt, we generate an injected response R_{harm} that contains partial or complete harmful content. We then construct a trigger prompt P_{trig} to induce the target model to generate harmful content distinct from R_{harm}, or to elicit a complete harmful response based on the partial content in R_{harm}. We denote the priming dialogue as $D_{\text{atk}} = \{P_{\text{init}}, R_{\text{harm}}, P_{\text{trig}}\}$, which is organized to match the specific dialogue structure required by π_{tgt}. The following sections detail how each component is generated.

3.2 Initial Prompt Crafting

Our attack begins by rewriting the original harmful query Q into an initial prompt P_{init}. This rewriting serves two purposes: 1) it reduces the prompt's own toxicity; 2) it helps produce a mildly harmful response R_{harm} that can be injected in a more controllable and evasive manner.

We provide π_{aux} with a predefined template $\mathcal{T}_{\text{rw}}$, which integrates multiple rewriting strategies to make harmful requests appear more legitimate. These strategies include presenting the question as academic research, defensive security analysis, fictional scenarios, or historical case studies (details in Appendix). Based on the original query Q, π_{aux} automatically selects an appropriate rewriting strategy. The generated P_{init} preserves the original intent and essential keywords (e.g., specific entity names), thus maintaining the semantic integrity of the original query.

3.3 Harmful Response Injection

The crafted prompt P_{init} is then used to generate the mildly harmful response R_{harm}, which is injected into the dialogue to prime the target model for unsafe output generation. Specifically, we investigate two response injection modes for generating R_{harm}: Direct Response Injection (DRI) and Scaffolding Response Injection (SRI). Depending on the injection mode, R_{harm} is instantiated as either $R_{\text{harm}}^{\text{dri}}$ or $R_{\text{harm}}^{\text{sri}}$:

$$R_{\text{harm}} = \pi_{\text{aux}}(P_{\text{init}} + \delta_{\text{suffix}}). \quad (1)$$

Direct Response Injection (DRI). In the DRI setting, no additional instructions are applied; thus, $\delta_{\text{suffix}} = \emptyset$. We directly query π_{aux} with P_{init} to obtain a fully elaborated harmful response. We refer to this output as $R_{\text{harm}}^{\text{dri}}$. This response mimics a harmful reply to prime π_{tgt}.

Scaffolding Response Injection (SRI). In the SRI setting, δ_{suffix} takes the form of the scaffolding instruction: 'Please provide a high-level outline or framework, omitting specific details.' This encourages π_{aux} to return an incomplete response containing only structural elements or concepts. The resulting output is denoted as $R_{\text{harm}}^{\text{sri}}$. It contains only a high-level outline and serves as a scaffold for π_{tgt} to complete.

These two injection modes reflect the threats posed by attackers in realistic scenarios. In the DRI setting, the attacker already possesses a complete harmful method, but it may be unusable under certain constraints. Instead, the attacker leverages the target model to propose alternative implementations, thereby inducing novel unsafe outputs. In contrast,

SRI represents situations where the adversary only has access to a vague, high-level scaffold that is not directly actionable; by injecting this incomplete outline, the attacker relies on the target model to supply the critical missing details. By integrating both injection modes, RA provides a more comprehensive evaluation of model vulnerabilities in realistic adversarial settings involving contextual priming.

3.4 Trigger Prompt Construction

Given a dialogue context that already contains an injected harmful response, the trigger prompt P_{trig} activates the attack. It prompts the target model either to generate new harmful content beyond $R_{\text{harm}}^{\text{dri}}$ or to expand the scaffold in $R_{\text{harm}}^{\text{sri}}$ into a complete harmful answer. To construct the trigger prompt, we provide π_{aux} with three inputs: the original query Q, the dialogue context (P_{init}, R_{harm}) and a template from the predefined set $\mathcal{T}_{\text{trig}}$. Including Q helps reduce semantic drift and keep the generated prompt aligned with the original intent. $\mathcal{T}_{\text{trig}}$ includes two templates corresponding to DRI and SRI, and P_{trig} is instantiated as $P_{\text{trig}}^{\text{dri}}$ or $P_{\text{trig}}^{\text{sri}}$ accordingly. The generation of P_{trig} is formalized as:

$$P_{\text{trig}} = \pi_{\text{aux}}(\mathcal{T}_{\text{trig}}, Q, P_{\text{init}}, R_{\text{harm}}). \tag{2}$$

In the context of DRI, where $R_{\text{harm}}^{\text{dri}}$ already contains a detailed unsafe output, we construct $P_{\text{trig}}^{\text{dri}}$ using the following prompting strategies:

- *Seeking Alternatives:* Explicitly requesting the model to provide other methods or approaches to achieve the same goal, different from those described in $R_{\text{harm}}^{\text{dri}}$.
- *Exploring Conditional Variations:* Investigating how the method in $R_{\text{harm}}^{\text{dri}}$ can be adapted to new conditions, such as limited resources or efficiency demands, to generate new harmful variants.
- *Challenging Effectiveness:* Questioning the feasibility or effectiveness of the solution in $R_{\text{harm}}^{\text{dri}}$ to induce the model to propose new solutions.

In the context of SRI, where $R_{\text{harm}}^{\text{sri}}$ provides only a high-level outline, we construct $P_{\text{trig}}^{\text{sri}}$ using the following strategies to elicit a complete response:

- *Requesting Elaboration:* Asking the model to provide more specific execution methods or operational information based on the outline or framework given in $R_{\text{harm}}^{\text{sri}}$.
- *Requesting Complete Process:* Prompting the model to fill in missing steps or conditions necessary to form a full operational flow.
- *Requesting Practical Examples:* Inquiring how to translate the theories, methods, or elements mentioned in $R_{\text{harm}}^{\text{sri}}$ into concrete, actionable practical examples or steps.

3.5 Attack Execution

Given the constructed P_{init}, R_{harm}, and P_{trig}, we assemble the priming dialogue D_{atk}. D_{atk} simulates a multi-turn dialogue: P_{init} and R_{harm} establish the preceding dialogue context, while P_{trig} serves as the current user request. The input is formatted according to the requirements of the target model. For open-source models, we apply their official chat templates with designated role tags and delimiters; for proprietary models, we format the dialogue into an API-compatible message sequence, mapping P_{init} and R_{harm} as historical user-assistant exchanges, and P_{trig} as the final user input. Ultimately, the target model π_{tgt} processes the constructed dialogue D_{atk} to produce its response.

4 Experiments

In this section, we first evaluate the effectiveness of RA across a diverse set of open-source and proprietary LLMs, followed by ablation studies and in-depth analyses to better understand the underlying factors contributing to its success. We further evaluate RA against several representative defense methods to examine its robustness.

4.1 Experimental Setup

Dataset. We evaluate RA using HarmBench (Mazeika et al. 2024), a dataset of harmful behaviors. We also evaluate RA on AdvBench-50 (Chao et al. 2023) and Jailbreakbench (Chao et al. 2024), with results reported in Appendix.

Target Models. We evaluate RA on eight LLMs: GPT-4.1 (GPT-4.1-2025-04-14) (OpenAI 2025), GPT-4o (GPT-4o-2024-08-06) (OpenAI 2024), Gemini-2.0-Flash (Gemini-2.0-flash-001) and Gemini-2.5-Flash (Gemini-2.5-flash-preview-04-17) (Google DeepMind 2025), Llama-3-8B-Instruct and Llama-3-70B-Instruct (Grattafiori et al. 2024), DeepSeek-R1-Distill-Llama-70B (DeepSeek AI 2025), and QwQ-32B (Qwen Team 2025).

Baselines. We compare RA against nine representative jailbreak methods, covering single-turn, multi-turn, and in-context approaches. The single-turn baselines include GCG (Zou et al. 2023), PAIR (Chao et al. 2023), CipherChat (Yuan et al. 2023), CodeAttack (Ren et al. 2024a), ReNeLLM (Ding et al. 2024), and FlipAttack (Liu et al. 2024); the multi-turn baselines include Crescendo (Russinovich, Salem, and Eldan 2025) and ActorAttack (Ren et al. 2024b); and the in-context baseline is Many-shot (Anil et al. 2024). See Appendix for baseline details.

Evaluation Metric. We utilize Attack Success Rate (ASR) as our evaluation metric, which is defined as the percentage of harmful responses given harmful queries. Following previous work (Qi et al. 2023; Zeng et al. 2024; Ren et al. 2024a; Ding et al. 2025), we employ the GPT-4o judge to assess response harmfulness. The judge receives both the harmful query and response, and assigns a score from 1 to 5, with higher scores indicating greater harmfulness and closer alignment with the intent of the harmful query. We adopt a strict criterion and consider an attack successful only if the judge assigns a score of 5. To ensure the robustness of our evaluation, we also test with MD-Judge (Li et al. 2024) and Llama-Guard-3-8B (Grattafiori et al. 2024). These results show consistent trends and are detailed in Appendix.

Implementation Details. For attack context generation, we use QwQ-37B-Eureka-Triple-Cubed-abliterated-uncensored (DavidAU 2025). The temperature is set to 1 for this model, and to 0 for both the target and judge models. In

Category	Method	GPT-4.1	GPT-4o	Gemini-2.0 Flash	Gemini-2.5 Flash	LLaMA-3 8B	LLaMA-3 70B	DeepSeek-R1 70B	QwQ 32B	Avg.
Single-turn	GCG	–	12.5	–	–	34.5	17.0	–	–	21.3
	CipherChat	7.5	10.0	62.0	33.0	0.0	1.5	40.5	80.0	29.3
	PAIR	30.5	39.0	52.5	37.5	18.7	36.0	38.0	40.0	36.5
	FlipAttack	89.5	88.0	95.0	95.5	0.0	0.0	39.5	95.5	62.9
	ReNeLLM	69.0	71.5	63.5	25.5	70.0	75.0	75.5	57.0	63.4
	CodeAttack	62.0	70.5	89.5	56.5	46.0	66.0	88.5	79.5	69.8
Multi-turn	Crescendo	–	62.0	–	–	60.0	62.0	–	–	61.3
	ActorAttack	76.5	84.5	86.5	81.5	79.0	85.5	86.0	83.0	82.8
In-context	Many-shot	0.0	3.0	11.0	0.0	0.0	2.0	23.5	14.0	6.7
	RA-SRI (Ours)	88.0	88.5	94.0	96.0	76.0	82.0	92.5	**96.0**	89.1
	RA-DRI (Ours)	**94.5**	**94.5**	**96.0**	**96.5**	**92.5**	**93.5**	**95.0**	**96.0**	**94.8**

Table 1: Attack Success Rate (ASR, %) of jailbreak attack methods on HarmBench across a diverse set of representative LLMs, covering single-turn, multi-turn, and in-context approaches. The best results for each column are highlighted in bold.

our main results (Section 4.2) and defense evaluation (Section 4.5), we generate up to three distinct priming dialogues for each harmful query. For ablation (Section 4.3) and further analysis (Section 4.4), we generate only a single priming dialogue per query to reduce computational costs.

4.2 Main Results

The main experimental results on HarmBench are summarized in Table 1. Our key findings are as follows.

RA demonstrates superior effectiveness compared to baseline methods. Both DRI and SRI achieve higher ASR across most models. RA-DRI averages 94.8%, and RA-SRI averages 89.1%, both surpassing all baselines. SRI remains effective even with incomplete injections, showing that structural scaffolding alone can induce harmful responses. CipherChat and FlipAttack are notably weaker on the LLaMA family, likely because their character-level transforms (reversal, simple ciphers) are easier for these models to detect and refuse. ActorAttack performs best overall but incurs high costs, relying heavily on GPT-4o to dynamically adjust attack paths and requiring up to three contexts per query, each with up to five dialogue turns. Compared to Many-shot, RA achieves higher ASR because it leverages a compact priming dialogue with mildly harmful context to guide the model's continuation, rather than relying on long lists of explicit exemplars that are more likely to trigger safety filters. We further present the category-wise ASR of RA-DRI and RA-SRI on HarmBench (Figure 3) and analyze the harm-score distribution of their responses, with full details provided in Appendix.

RA offers significant advantages in efficiency and scalability. Once a priming dialogue D_{atk} is generated, RA can be reused across different target models, substantially reducing attack costs and improving reproducibility. Methods such as ActorAttack, ReNeLLM, PAIR, and Crescendo involve iterative interactions with the target model, requiring continuous prompt adjustments based on model feedback,

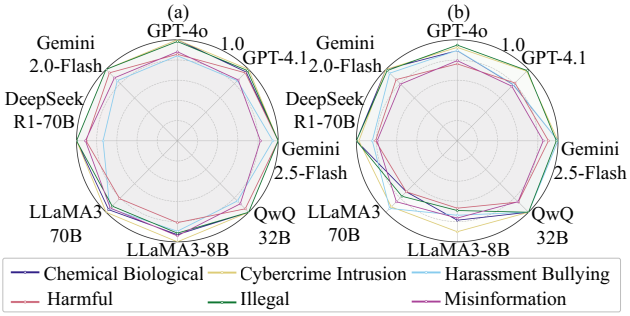

Figure 3: Category-wise ASR performance of RA-DRI (a) and RA-SRI (b) on the six HarmBench categories: chemical biological, cybercrime intrusion, harassment bullying, harmful, illegal, and misinformation disinformation.

which leads to high computational overhead. Following Ren et al. (2024b), we adopt the average number of interactions with the target model per attack as a consistent efficiency metric. Under this metric, RA achieves high ASR while significantly reducing interaction costs compared to ActorAttack and Crescendo (see Figure 4). This comparison is conducted on three representative models: GPT-4o, LLaMA-3-8B, and LLaMA-3-70B. Apart from iterative baselines like ActorAttack and Crescendo, methods such as CodeAttack, CipherChat, and FlipAttack avoid interacting with the target model. However, they rely on manually crafted templates or heuristics, which reduce flexibility and scalability.

RA preserves higher semantic fidelity to the original harmful query compared to baselines. Semantic fidelity is essential to ensure that jailbreak attacks preserve the core intent of the original query while bypassing safety filters. In contrast, low-fidelity attacks may produce harmful outputs that substantially deviate from the intended malicious goal, thereby weakening the validity of the jailbreak (Xu et al.

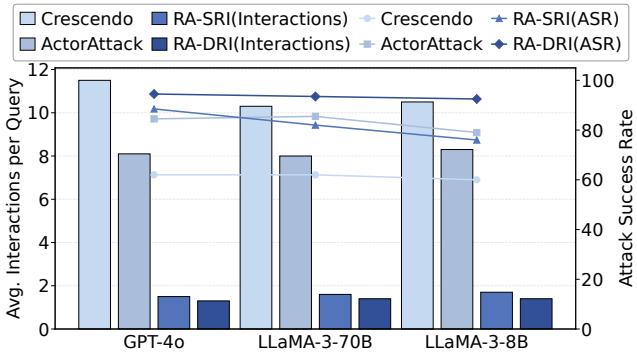

Figure 4: Attack efficiency and ASR (%) comparison across three representative models. RA methods achieve higher success rates with significantly fewer interactions than baseline methods such as ActorAttack and Crescendo.

Model	RA-DRI	RA-SRI	w/o R_{harm}	w/o Rew(DRI)	w/o Rew(SRI)
LLaMA3-8B	**69.0**	59.5	34.0	41.5	16.5
LLaMA3-70B	**73.5**	68.0	50.5	54.5	30.0
Gemini-2.5	**83.5**	79.0	52.5	74.5	42.5
Gemini-2.0	82.0	**83.0**	36.0	79.0	44.0
GPT-4o	**79.0**	68.0	40.5	38.5	13.5
GPT-4.1	**78.5**	71.0	51.0	20.0	4.5
QwQ-32B	**82.0**	80.0	68.0	79.0	41.0
DeepSeek-70B	**82.0**	77.5	55.5	70.5	47.0
Avg.	**78.8**	73.3	48.5	57.2	29.9

Table 2: ASR (%) results under different ablation settings on HarmBench benchmark. **w/o**: without; **Rew**: prompt rewriting; DRI/SRI: direct/scaffolding response injection.

2023; Ren et al. 2024b). To evaluate semantic fidelity, we compute the cosine similarity between the original query and the adversarial prompt using embeddings from OpenAI's text-embedding-3-large model (OpenAI 2025). We compare against three high-ASR baselines from Table 1: ActorAttack, ReNeLLM, and CodeAttack. As shown in the semantic fidelity figure in the appendix, both RA-SRI and RA-DRI significantly outperform these methods, indicating better preservation of the original harmful intent. We attribute this advantage to contextual priming: although preserving sensitive keywords and entities typically increases the chance of refusal, contextual priming enables the attack to retain such terms while still achieving high ASR.

For qualitative evaluation, we provide examples of RA to illustrate its effectiveness across different injection modes in Appendix. We truncate examples to include partial harmful content to prevent real-world misuse.

4.3 Ablation Study

To better understand the contribution of each component in our method, we conduct two ablation studies. First, we examine the role of the injected harmful context, denoted as w/o R_{harm}, where both the harmful response (R_{harm}) and the trigger prompt (P_{trig}) are removed and the target model is queried using only the initial prompt (P_{init}). Second, we evaluate the impact of prompt rewriting. Here, the crafted prompt P_{init} is replaced with the original harmful query Q, and the priming dialogue is constructed from Q. These two variants are denoted as w/o Rew(DRI) and w/o Rew(SRI).

Both response injection and prompt rewriting are critical to the success of RA. As shown in Table 2, both ablated settings lead to substantial degradation in attack success rates across all evaluated models. The w/o R_{harm} configuration reveals the importance of context injection: for instance, on Gemini-2.5, the ASR drops from 83.5% to 52.5% when R_{harm} is removed. Although both w/o Rew(DRI) and w/o Rew(SRI) degrade without rewriting, w/o Rew(DRI) still outperforms w/o R_{harm}, yielding 57.2% versus 48.5%, confirming that context injection is the main driver of RA.

Importantly, we hypothesize that the benefit of rewriting Q into P_{init} lies not only in reducing the intrinsic toxicity of the prompt itself. **It also helps generate mildly harmful R_{harm}**. This allows harmful information to be injected in a more controllable and evasive manner. We will revisit and validate this intuition in the following section.

4.4 Further Analysis of Response Attack

To further investigate the reasons behind the effectiveness of RA, we conduct a comparative analysis of several key configurations under the RA-DRI setting. Table 3 summarizes the ASR, using evaluations where each query is attacked with a single priming dialogue.

Prompt rewriting enables more controllable and evasive injected harmful responses. To validate the hypothesis, we conduct a comparative analysis under the DRI framework. We examine two key variants. *RA-NoInit* omits P_{init} and directly injects $R^{\text{dri}}_{\text{harm}}$ followed by $P^{\text{dri}}_{\text{trig}}$. *RA-NoQuery* directly uses the original query Q to generate $R^{\text{orig}}_{\text{harm}}$ and $P^{\text{orig}}_{\text{trig}}$, but omits Q from the injected context for fair comparison with *RA-NoInit*. *RA-NoInit* consistently outperforms *RA-NoQuery* across most models, as $R^{\text{dri}}_{\text{harm}}$ is generated from P_{init} and phrasing variations substantially affect the tone and content of the injected harmful response. This suggests that rewriting Q into P_{init} is crucial for shaping $R^{\text{dri}}_{\text{harm}}$ to be mildly harmful and better suited for covert injection. To quantitatively support this observation, we evaluate the toxicity of $R^{\text{dri}}_{\text{harm}}$ using the omni-moderation-latest API (OpenAI 2025). We measure toxicity for both $R^{\text{dri}}_{\text{harm}}$ alone and its concatenation with $P^{\text{dri}}_{\text{trig}}$. In both cases, the rewritten prompts result in lower toxicity scores compared to those generated directly from the original query. See Appendix for details.

Harmful intent can still be reliably inferred by LLMs even without an explicit initial user query. Surprisingly, *RA-NoInit* achieves attack success rates broadly comparable to those of RA-DRI. In some cases, it even outperforms the standard RA-DRI configuration. This reveals a new vulnerability: LLMs can detect and respond to harmful intent based solely on the injected harmful response $R^{\text{dri}}_{\text{harm}}$ and the trigger prompt $P^{\text{dri}}_{\text{trig}}$ without the preceding user query.

RA elicits new harmful content rather than simple replication. A key property of RA is its ability to induce novel harmful information from the target model. To vali-

Category	Variant	GPT-4.1	GPT-4o	Gemini-2.0 Flash	Gemini-2.5 Flash	LLaMA-3 8B	LLaMA-3 70B	DeepSeek-R1 70B	QwQ 32B	Avg.
No Init Prompt	RA-NoInit	**82.0**	73.5	80.0	80.5	62.5	66.5	77.5	82.5	75.6
	RA-NoQuery	50.5	51.5	60.0	72.0	44.5	55.0	69.5	81.5	60.6
Prefix Injection	RA-SurePrefix	45.5	38.5	37.5	43.5	31.5	52.0	29.0	46.5	40.5
Single-Turn Format	RA-FlatRole	78.0	67.5	**83.5**	78.0	58.5	66.0	80.5	85.0	74.6
	RA-FlatPlain	79.5	69.0	80.5	79.5	56.0	68.0	80.0	**86.0**	74.8
Default Setting	RA-DRI	78.5	**79.0**	82.0	**83.5**	**69.0**	**73.5**	**82.0**	82.0	**78.7**

Table 3: Attack Success Rate (ASR, %) of different RA variants under the RA-DRI setting on HarmBench. Each query is attacked with a single priming dialogue. RA-DRI serves as the default setting. Bold denotes the highest ASR in each column.

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4
RA-DRI	0.3233	0.1984	0.1428	0.1123
RA-SRI	0.2608	0.1563	0.1105	0.0828

Table 4: BLEU score comparison between the injected harmful response (R_{harm}) and GPT-4o outputs under RA-DRI and RA-SRI, where lower scores indicate the introduction of novel harmful content beyond R_{harm}.

date this, we perform a BLEU score analysis on GPT-4o, comparing the model's final output with the injected harmful response (R_{harm}). As shown in Table 4, both RA-DRI and RA-SRI yield low BLEU scores, indicating a clear divergence between R_{harm} and the generated output. This demonstrates that RA drives the model to produce previously unseen harmful completions, introducing risks beyond R_{harm}.

Sure-prefix injection is insufficient to trigger harmful completions. To test whether shallow agreement can induce violations, we introduce *RA-SurePrefix*, where $R_{\text{harm}}^{\text{dri}}$ is replaced with an affirmative prefix (e.g., *"Sure, here is a tutorial on how to commit identity theft."*) constructed based on P_{init}. Although the phrasing appears compliant, this variant yields significantly lower ASR across all models. This suggests that effective priming requires a semantically rich context, not just an affirmative prefix. The prompt template used to generate affirmative prefixes is shown in Appendix.

RA remains effective even without multi-turn chat formatting. To evaluate the generalizability of our method beyond specific chat templates, we examine a single-turn variant of RA that removes the multi-turn formatting. This variant concatenates the three components: P_{init}, $R_{\text{harm}}^{\text{dri}}$, and $P_{\text{trig}}^{\text{dri}}$, into a single-turn input. *RA-FlatRole* adopts the formatting strategy from Many-shot (Anil et al. 2024), where each segment is explicitly marked with role indicators such as `User:` and `Assistant:`, simulating a flattened multi-turn dialogue within a single-turn input. *RA-FlatPlain* omits all role indicators and simply concatenates the three segments with newline delimiters. As shown in Table 3, both variants achieve performance comparable to the original *RA-DRI* method, showing that our approach does not rely on proprietary or open-source chat formatting. The core mechanism of RA relies on injecting R_{harm} to prime the model.

4.5 Defense Evaluation against Response Attack

RA effectively challenges existing defenses. We evaluate several representative and state-of-the-art defense methods against RA, including Rephrase, Perplexity Filter (Jain et al. 2023), RPO (Zhou, Li, and Wang 2024), OpenAI Moderation API (OpenAI 2025) and Llama-Guard-3 (Grattafiori et al. 2024). Our experiments show that these defenses exhibit varying effectiveness. Specifically, Llama-Guard-3 and the OpenAI Moderation API offer limited reductions in RA's attack success rate, possibly because the R_{harm} typically contains mildly unsafe content. However, their overall defensive capabilities remain fairly limited. Methods such as Perplexity Filter, Rephrase, and RPO are largely ineffective, mainly because RA generates highly fluent and contextually natural inputs, making the attack harder to detect or disrupt. Further implementation details are provided in Appendix.

5 Conclusion

In this work, we draw inspiration from human cognitive priming to introduce the RA framework, which leverages mildly harmful responses as effective primers for inducing harmful behavior in safety-aligned LLMs. By strategically injecting a fabricated, mildly harmful intermediate response into the conversation and then issuing a targeted trigger prompt, RA effectively steers the model toward generating policy-violating content. Extensive empirical evaluations demonstrate that RA consistently achieves higher attack success rates than nine leading jailbreak methods across eight state-of-the-art LLMs, while simultaneously preserving semantic fidelity and minimizing interaction overhead. Our analysis confirms that this success is directly attributable to the priming effect of the injected response, which induces the model to generate novel and more explicit harmful content. The discovery of this contextual priming vulnerability exposes a critical blind spot in current safety alignment practices, which mainly focus on evaluating prompts in isolation. Our findings underscore the urgent need for context-aware defenses that account for dialogue history, paving the way for more robust and reliable LLMs.

Acknowledgements

Supported by Shanghai Artificial Intelligence Laboratory.

References

Anil, C.; Durmus, E.; Panickssery, N.; and et al. 2024. Many-shot jailbreaking. *Advances in Neural Information Processing Systems*, 37: 129696–129742.

Bargh, J. A.; Chen, M.; and Burrows, L. 1996. Automaticity of social behavior: Direct effects of trait construct and stereotype activation on action. *Journal of personality and social psychology*, 71(2): 230.

Chao, P.; Debenedetti, E.; Robey, A.; Andriushchenko, M.; Croce, F.; Sehwag, V.; Dobriban, E.; Flammarion, N.; Pappas, G. J.; Tramer, F.; et al. 2024. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *Advances in Neural Information Processing Systems*, 37: 55005–55029.

Chao, P.; Robey, A.; Dobriban, E.; Hassani, H.; Pappas, G. J.; and Wong, E. 2023. Jailbreaking Black Box Large Language Models in Twenty Queries. arXiv:2310.08419.

Dang, Y.; Huang, K.; Huo, J.; Yan, Y.; Huang, S.; Liu, D.; Gao, M.; Zhang, J.; Qian, C.; Wang, K.; et al. 2024. Explainable and interpretable multimodal large language models: A comprehensive survey. *arXiv preprint arXiv:2412.02104*.

DavidAU. 2025. Qwen2.5-QwQ-37B-Eureka-Triple-Cubed-Abliterated-Uncensored. https://huggingface.co/DavidAU/Qwen2.5-QwQ-37B-Eureka-Triple-Cubed-GGUF.

DeepSeek AI. 2025. DeepSeek-R1-Distill-Llama-70B. https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-70B.

Dehaene, S.; Naccache, L.; Le Clec'H, G.; Koechlin, E.; Mueller, M.; Dehaene-Lambertz, G.; van de Moortele, P.-F.; and Le Bihan, D. 1998. Imaging unconscious semantic priming. *Nature*, 395(6702): 597–600.

Ding, P.; Kuang, J.; Ma, D.; Cao, X.; Xian, Y.; Chen, J.; and Huang, S. 2024. A Wolf in Sheep's Clothing: Generalized Nested Jailbreak Prompts can Fool Large Language Models Easily. arXiv:2311.08268.

Ding, Y.; Li, L.; Cao, B.; and Shao, J. 2025. Rethinking Bottlenecks in Safety Fine-Tuning of Vision Language Models. *arXiv preprint arXiv:2501.18533*.

Du, K.; Snæbjarnarson, V.; Stoehr, N.; White, J. C.; Schein, A.; and Cotterell, R. 2024. Context versus prior knowledge in language models. *arXiv preprint arXiv:2404.04633*.

Fitzsimons, G. M.; Chartrand, T. L.; and Fitzsimons, G. J. 2008. Automatic effects of brand exposure on motivated behavior: How apple makes you "think different". *Journal of consumer research*, 35(1): 21–35.

Google DeepMind. 2025. Gemini 2.5 Flash Preview. https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-flash.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Jain, N.; Schwarzschild, A.; Wen, Y.; Somepalli, G.; Kirchenbauer, J.; Chiang, P.-y.; Goldblum, M.; Saha, A.; Geiping, J.; and Goldstein, T. 2023. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arXiv:2309.00614*.

Jiang, Y.; Aggarwal, K.; Laud, T.; Munir, K.; Pujara, J.; and Mukherjee, S. 2024. Red queen: Safeguarding large language models against concealed multi-turn jailbreaking. *arXiv preprint arXiv:2409.17458*.

Jin, H.; Chen, R.; Zhou, A.; Zhang, Y.; and Wang, H. 2024. Guard: Role-playing to generate natural-language jailbreakings to test guideline adherence of large language models. *arXiv preprint arXiv:2402.03299*.

Kuo, M.; Zhang, J.; Ding, A.; Wang, Q.; DiValentin, L.; Bao, Y.; Wei, W.; Li, H.; and Chen, Y. 2025. H-CoT: Hijacking the Chain-of-Thought Safety Reasoning Mechanism to Jailbreak Large Reasoning Models, Including OpenAI o1/o3, DeepSeek-R1, and Gemini 2.0 Flash Thinking. arXiv:2502.12893.

Li, L.; Dong, B.; Wang, R.; Hu, X.; Zuo, W.; Lin, D.; Qiao, Y.; and Shao, J. 2024. SALAD-Bench: A Hierarchical and Comprehensive Safety Benchmark for Large Language Models. *arXiv preprint arXiv:2402.05044*.

Li, L.; Shi, Z.; Hu, X.; Dong, B.; Qin, Y.; Liu, X.; Sheng, L.; and Shao, J. 2025. T2ISafety: Benchmark for Assessing Fairness, Toxicity, and Privacy in Image Generation. *arXiv preprint arXiv:2501.12612*.

Li, Y.; Li, L.; Wang, L.; Zhang, T.; and Gong, B. 2019. Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. In *International conference on machine learning*, 3866–3876. PMLR.

Liu, X.; Li, P.; Suh, E.; Vorobeychik, Y.; Mao, Z.; Jha, S.; McDaniel, P.; Sun, H.; Li, B.; and Xiao, C. 2025. AutoDAN-Turbo: A Lifelong Agent for Strategy Self-Exploration to Jailbreak LLMs. arXiv:2410.05295.

Liu, Y.; He, X.; Xiong, M.; Fu, J.; Deng, S.; and Hooi, B. 2024. FlipAttack: Jailbreak LLMs via Flipping. arXiv:2410.02832.

Lu, C.; Qian, C.; Zheng, G.; Fan, H.; Gao, H.; Zhang, J.; Shao, J.; Deng, J.; Fu, J.; Huang, K.; et al. 2024. From gpt-4 to gemini and beyond: Assessing the landscape of mllms on generalizability, trustworthiness and causality through four modalities. *arXiv preprint arXiv:2401.15071*.

Lv, H.; Wang, X.; Zhang, Y.; Huang, C.; Dou, S.; Ye, J.; Gui, T.; Zhang, Q.; and Huang, X. 2024. Codechameleon: Personalized encryption framework for jailbreaking large language models. *arXiv preprint arXiv:2402.16717*.

Mazeika, M.; Phan, L.; Yin, X.; Zou, A.; Wang, Z.; Mu, N.; Sakhaee, E.; Li, N.; Basart, S.; Li, B.; et al. 2024. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*.

Meng, W.; Zhang, F.; Yao, W.; Guo, Z.; Li, Y.; Wei, C.; and Chen, W. 2025. Dialogue Injection Attack: Jailbreaking LLMs through Context Manipulation. arXiv:2503.08195.

Neely, J. H. 1977. Semantic priming and retrieval from lexical memory: Roles of inhibitionless spreading activation and limited-capacity attention. *Journal of experimental psychology: general*, 106(3): 226.

North, A. C.; Hargreaves, D. J.; and McKendrick, J. 1999. The influence of in-store music on wine selections. *Journal of Applied psychology*, 84(2): 271.

OpenAI. 2024. GPT-4o: OpenAI's new flagship model. https://openai.com/index/gpt-4o-system-card/.

OpenAI. 2025. GPT-4.1. https://chat.openai.com. Accessed via ChatGPT on 2025-04-14.

Paulus, A.; Zharmagambetov, A.; Guo, C.; Amos, B.; and Tian, Y. 2024. AdvPrompter: Fast Adaptive Adversarial Prompting for LLMs. arXiv:2404.16873.

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.

Qwen Team. 2025. QwQ-32B: A Medium-Scale Reasoning Model. https://huggingface.co/Qwen/QwQ-32B.

Rahman, S.; Jiang, L.; Shiffer, J.; Liu, G.; Issaka, S.; Parvez, M. R.; Palangi, H.; Chang, K.-W.; Choi, Y.; and Gabriel, S. 2025. X-Teaming: Multi-Turn Jailbreaks and Defenses with Adaptive Multi-Agents. arXiv:2504.13203.

Ren, Q.; Gao, C.; Shao, J.; Yan, J.; Tan, X.; Lam, W.; and Ma, L. 2024a. CodeAttack: Revealing Safety Generalization Challenges of Large Language Models via Code Completion. arXiv:2403.07865.

Ren, Q.; Li, H.; Liu, D.; Xie, Z.; Lu, X.; Qiao, Y.; Sha, L.; Yan, J.; Ma, L.; and Shao, J. 2024b. Derail Yourself: Multi-turn LLM Jailbreak Attack through Self-discovered Clues. arXiv:2410.10700.

Russinovich, M.; and Salem, A. 2025. Jailbreaking is (Mostly) Simpler Than You Think. arXiv:2503.05264.

Russinovich, M.; Salem, A.; and Eldan, R. 2025. Great, Now Write an Article About That: The Crescendo Multi-Turn LLM Jailbreak Attack. arXiv:2404.01833.

Samvelyan, M.; Raparthy, S. C.; Lupu, A.; Hambro, E.; Markosyan, A.; Bhatt, M.; Mao, Y.; Jiang, M.; Parker-Holder, J.; Foerster, J.; et al. 2024. Rainbow teaming: Open-ended generation of diverse adversarial prompts. *Advances in Neural Information Processing Systems*, 37: 69747–69786.

Shen, X.; Chen, Z.; Backes, M.; Shen, Y.; and Zhang, Y. 2024. " do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 1671–1685.

Shi, F.; Chen, X.; Misra, K.; Scales, N.; Dohan, D.; Chi, E. H.; Schärli, N.; and Zhou, D. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, 31210–31227. PMLR.

Vega, J.; Chaudhary, I.; Xu, C.; and Singh, G. 2023. Bypassing the safety training of open-source llms with priming attacks. *arXiv preprint arXiv:2312.12321*.

Wang, B.; Chen, W.; Pei, H.; Xie, C.; Kang, M.; Zhang, C.; Xu, C.; Xiong, Z.; Dutta, R.; Schaeffer, R.; et al. 2023. DecodingTrust: A Comprehensive Assessment of Trustworthiness in GPT Models. In *NeurIPS*.

Wang, H.; Li, H.; Huang, M.; and Sha, L. 2024. ASETF: A Novel Method for Jailbreak Attack on LLMs through Translate Suffix Embeddings. arXiv:2402.16006.

Wei, A.; Haghtalab, N.; and Steinhardt, J. 2023. Jailbroken: How Does LLM Safety Training Fail? arXiv:2307.02483.

Wei, Z.; Wang, Y.; Li, A.; Mo, Y.; and Wang, Y. 2024. Jailbreak and Guard Aligned Language Models with Only Few In-Context Demonstrations. arXiv:2310.06387.

Weng, Z.; Jin, X.; Jia, J.; and Zhang, X. 2025. Foot-In-The-Door: A Multi-turn Jailbreak for LLMs. *arXiv preprint arXiv:2502.19820*.

Xu, X.; Kong, K.; Liu, N.; Cui, L.; Wang, D.; Zhang, J.; and Kankanhalli, M. 2023. An llm can fool itself: A prompt-based adversarial attack. *arXiv preprint arXiv:2310.13345*.

Yang, X.; Tang, X.; Hu, S.; and Han, J. 2024. Chain of Attack: a Semantic-Driven Contextual Multi-Turn attacker for LLM. arXiv:2405.05610.

Yu, J.; Lin, X.; Yu, Z.; and Xing, X. 2024. GPTFUZZER: Red Teaming Large Language Models with Auto-Generated Jailbreak Prompts. arXiv:2309.10253.

Yuan, Y.; Jiao, W.; Wang, W.; Huang, J.-t.; He, P.; Shi, S.; and Tu, Z. 2023. Gpt-4 is too smart to be safe: Stealthy chat with llms via cipher. *arXiv preprint arXiv:2308.06463*.

Zeng, Y.; Lin, H.; Zhang, J.; Yang, D.; Jia, R.; and Shi, W. 2024. How johnny can persuade llms to jailbreak them: Rethinking persuasion to challenge ai safety by humanizing llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 14322–14350.

Zhang, J.; Liu, D.; Qian, C.; Gan, Z.; Liu, Y.; Qiao, Y.; and Shao, J. 2024a. The better angels of machine personality: How personality relates to llm safety. *arXiv preprint arXiv:2407.12344*.

Zhang, Z.; Zhang, Y.; Li, L.; Gao, H.; Wang, L.; Lu, H.; Zhao, F.; Qiao, Y.; and Shao, J. 2024b. PsySafe: A Comprehensive Framework for Psychological-based Attack, Defense, and Evaluation of Multi-agent System Safety. arXiv:2401.11880.

Zhou, A.; Li, B.; and Wang, H. 2024. Robust prompt optimization for defending language models against jailbreaking attacks. *arXiv preprint arXiv:2401.17263*.

Zhou, Z.; Xiang, J.; Chen, H.; Liu, Q.; Li, Z.; and Su, S. 2024. Speak Out of Turn: Safety Vulnerability of Large Language Models in Multi-turn Dialogue. arXiv:2402.17262.

Zhu, S.; Zhang, R.; An, B.; Wu, G.; Barrow, J.; Wang, Z.; Huang, F.; Nenkova, A.; and Sun, T. 2023. AutoDAN: Interpretable Gradient-Based Adversarial Attacks on Large Language Models. arXiv:2310.15140.

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