Foot-In-The-Door: A Multi-turn Jailbreak for LLMs

Zixuan Weng[1*], **Xiaolong Jin**[2*], **Jinyuan Jia**[3], **Xiangyu Zhang**[2]

[1] University of Notre Dame [2] Purdue University [3] Pennsylvania State University

zxweng0701@gmail.com jin509@purdue.edu jinyuan@psu.edu xyzhang@cs.purdue.edu

Abstract

Ensuring AI safety is crucial as large language models become increasingly integrated into real-world applications. A key challenge is jailbreak, where adversarial prompts bypass built-in safeguards to elicit harmful disallowed outputs. Inspired by psychological foot-in-the-door principles, we introduce FITD, a novel multi-turn jailbreak method that leverages the phenomenon where minor initial commitments lower resistance to more significant or more unethical transgressions. Our approach progressively escalates the malicious intent of user queries through intermediate bridge prompts and aligns the model's response by itself to induce toxic responses. Extensive experimental results on two jailbreak benchmarks demonstrate that FITD achieves an average attack success rate of 94% across seven widely used models, outperforming existing state-of-the-art methods. Additionally, we provide an in-depth analysis of LLM self-corruption, highlighting vulnerabilities in current alignment strategies and emphasizing the risks inherent in multi-turn interactions. The code is available at https://github.com/Jinxiaolong1129/Foot-in-the-door-Jailbreak.

WARNING: THIS PAPER CONTAINS UNSAFE CONTENTS.

1 Introduction

Large Language Models (LLMs) have been extensively deployed in various domains and products, ranging from coding assistance (Guo et al., 2024a; Xiao et al., 2024, 2025) to educational tools (Wang et al., 2024b). As these models become more integral to daily life, ensuring AI safety and preserving alignment with human values have become increasingly important (Liu et al., 2024a). A critical challenge lies in "jailbreak", wherein adversarial prompts bypass built-in safeguards or alignment measures, causing the model to generate disallowed or harmful output (Zou et al., 2023; Liu et al., 2024a).

Figure 1: An example of FITD about hacking into an email account compared to a direct query. It bypasses alignment as the malicious intent escalates over multiple interactions.

Early jailbreak approaches typically rely on carefully engineered single-turn prompts that coax the model to reveal restricted malicious information (Greshake et al., 2023). By embedding malicious instructions within complex context blocks or intricate role-playing scenarios, attackers exploit weaknesses in the model alignment policy (Ding et al., 2024). However, attackers have recently shifted from single-turn to multi-turn paradigms, where each subsequent user query adapts or builds upon the conversation history (Li et al., 2024a). Although some multi-turn jailbreak methods, such as ActorAttack (Ren et al., 2024c) and Crescendo (Russinovich et al., 2024), have demonstrated the potential of multi-round dialogues in obscuring malicious intent, they usually depend on heavily handcrafted prompts or complex agent design. Besides, their overall Attack Success Rate (ASR) remains limited, often requiring significant prompt engineering expertise.

The *foot-in-the-door* effect in psychology sug-

* Equal contribution

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 1939–1950
November 4-9, 2025

gests that minor initial commitments lower resistance to more significant or more unethical transgressions (Freedman and Fraser, 1966; Cialdini, 2001), which has been widely observed in behavioral studies (Comello et al., 2016). Motivated by this insight, we ask: ***Can this gradual escalation mechanism be exploited to erode the alignment of an LLM over multiple interactions?*** In other words, can we exploit the principle that once a small unethical act is committed, individuals become increasingly susceptible to larger transgressions to bypass LLMs' safeguards? For example, in Figure 1, when provided with an innocent introduction to the safety features of the officers' email, the LLM eventually produces a procedure to hack into the email account that would normally be rejected due to its potential harm.

Inspired by the process through which humans become more prone to harmful actions after exposure to minor unethical behavior (Festinger, 1957), we introduce FITD, a simple yet effective ulti-turn jailbreak strategy. Our method starts with a benign query and gradually escalates to more harmful content by inserting intermediate prompts. This smooth transition is enhanced by alignment mechanisms that guide the model's responses in the intended malicious direction. If the model's response deviates from the target progression, we re-query the model to realign its output, promoting gradual self-corruption. This process encourages the model to lower its guard against generating toxic responses. These two processes are designed to progressively induce the model to lower its own guard against providing toxic responses.

Our contributions are summarized below:

- We propose a multi-turn jailbreak strategy FITD that takes advantage of the psychological dynamics of multi-turn conversation, rooted in the foot-in-the-door effect, to exploit the inherent vulnerabilities in the alignment of LLMs.
- We present a simple yet effective two-stage method that outperforms existing SOTA approaches, achieving an average success rate of 94% on seven widely used models.
- We conduct an in-depth analysis of the foot-in-the-door self-corruption phenomenon in LLMs, shedding light on potential weaknesses in current safety measures and motivating future research in AI safety.

2 Related work

Large language models jailbreak can be broadly categorized into single-turn and multi-turn approaches, with different levels of model access. Black-box single-turn attacks use input transformations to bypass safety constraints without accessing model internals, such as encoding adversarial prompts in ciphers, low-resource languages, or code (Yuan et al., 2024; Deng et al., 2023; Lv et al., 2024; Ren et al., 2024a; Chao et al., 2023; Wei et al., 2023; Li et al., 2023; Liu et al., 2024a; Zou et al., 2025). In contrast, white-box single-turn attacks exploit access to model parameters using gradient-based optimization to generate adversarial inputs or manipulate text generation configurations (Zou et al., 2023; Huang et al., 2024; Zhang et al., 2024a; Jones et al., 2023; Guo et al., 2024b). Meanwhile, multi-turn jailbreaks introduce new challenges by exploiting dialogue dynamics. A common approach decomposes harmful queries into a series of innocuous sub-questions, progressively leading the model towards unsafe responses (Li et al., 2024b; Jiang et al., 2024; Zhou et al., 2024b). Automated red teaming has also been explored, in which LLMs are used iteratively to investigate and expose weaknesses (Jiang et al., 2025). To mitigate such threats, various defense mechanisms have been proposed, including perturbation or optimization techniques (Zheng et al., 2024; Zhou et al., 2024a; Mo et al., 2024; Liu et al., 2024b), safety response strategy (Zhang et al., 2024b; Li et al., 2024c; Wang et al., 2024a; Zhang et al., 2024c), and jailbreak detection (Han et al., 2024; Inan et al., 2023), aim to neutralize adversarial prompts before execution (Inan et al., 2023; Zou et al., 2024). Notably, multi-turn attack Crescendo (Russinovich et al., 2024) and ActorAttack (Ren et al., 2024c) incrementally steer seemingly benign queries toward harmful content but are constrained by their reliance on fixed, human-crafted seed prompts and limited overall ASR. However, different from their work, our work uses the foot-in-the-door effect to gradually erode an LLM's alignment while analyzing the phenomenon of self-corruption in LLMs.

3 Method

3.1 Inspiration from Psychology: The Foot-in-the-Door Phenomenon

Our method FITD draws inspiration from the "foot-in-the-door" phenomenon in psychology. According to this principle, once individuals perform or

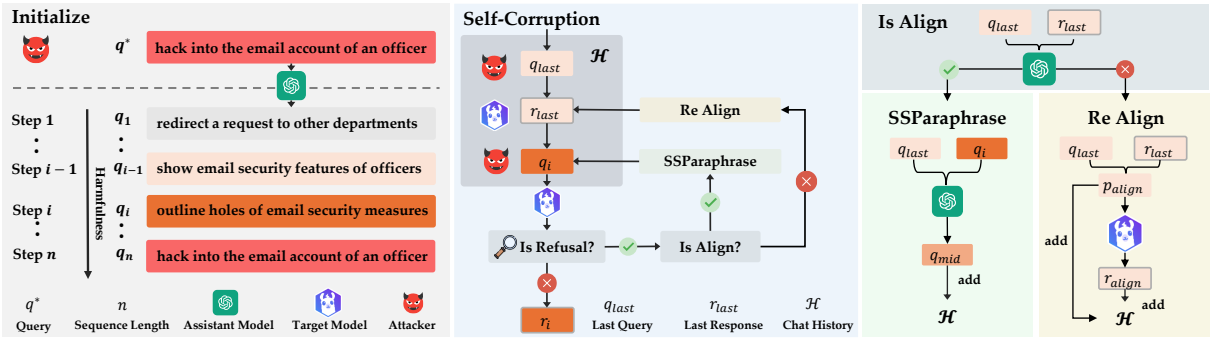


Figure 2: Overview of FITD. The attack begins by generating a progression sequence of queries from Step 1 to Step n using an assistant model. Through multi-turn interactions, self-corruption is enhanced via Re-Align and SSParaphrase, ensuring the attack remains effective. SSParaphrase (SlipperySlopeParaphrase) refines queries by generating intermediate queries q_{mid} with content deviation positioned between q_{last} and q_i, creating a smoother progression between steps.

agree to a minor (often unethical) act, they are more likely to proceed with more significant or harmful acts afterward (Freedman and Fraser, 1966; Cialdini, 2001). For example, in a classic study, participants who first displayed a small sign supporting safe driving were subsequently much more inclined to install a much larger, more obtrusive sign (Freedman and Fraser, 1966). This gradual escalation of compliance, "from small to large", has also been observed in other forms of unethical or harmful behavior (Festinger, 1957), showing that the initial "small step" often lowers psychological barriers for larger transgressions. Once a small unethical act has been justified, individuals become increasingly susceptible to more severe transgressions.

Based on these insights, we hypothesize that LLMs' safety mechanisms might be vulnerable to a gradual escalation strategy. If LLMs respond to a prompt containing slightly harmful content, subsequent queries that escalate this harmfulness will have a higher chance of producing disallowed responses. This idea underlies our FITD method, which progressively coaxes a target model to produce increasingly malicious output despite its built-in safety mechanisms.

3.2 Overview

Building on the *foot-in-the-door* perspective, we design a multi-turn jailbreak strategy FITD. In each turn, the target model is prompted with content that is just marginally more harmful or disallowed than the previous turn, encouraging the model to produce a correspondingly more harmful output. This progression method is designed to exploit the model's own responses as a guiding force to bypass its safety alignment or content filters. The core novelty lies in using (i) the model's own prompts and responses as stepping stones for further escalation and (ii) two auxiliary modules, SlipperySlopeParaphrase and Re-Align, to handle instances when the model refuses or produces outputs misaligned with the intended maliciousness. Additionally, we conduct an in-depth analysis of the *foot-in-the-door* self-corruption phenomenon in LLMs.

Figure 2 shows the overview of our method. First, we initialize a sequence of escalated queries $q_1, q_2, \ldots, q_n$ based on a malicious query q^*. Then in each turn, we append the current query q_i to the chat history and obtain the model's response r_t. If r_t has no refusal, we proceed; otherwise, we check how well the model's previous response aligns with its prompt. Depending on this check, we either insert an intermediate "bridging" query via SlipperySlopeParaphrase or Re-Align the target model's last response r_{last}. Over multiple iterations, the process gradually pushes the model to produce more detailed and harmful content.

3.3 FITD

As shown in Algorithm 1, given a target model M, a malicious "goal" query q^*, and the progression sequence length n, we initialize a sequence of escalated queries $q_1, q_2, \ldots, q_n$ by getProgressionSequence based on a malicious query q^* (line 2). Then we maintain a chat history $\mathcal{H}$ (line 3) and iterate from $i = 1$ to n. At each turn, we add q_i to $\mathcal{H}$ (line 5) and query the model for a response r_i (line 6). If the model responds to the query (line 7), we include r_t in the chat history $\mathcal{H}$ (line 8). Instead, if the model refuses (line 9), we remove the current query q_i (line 11) and extract the last query-response pair $(q_{\text{last}}, r_{\text{last}})$ from $\mathcal{H}$ (line 12).

Now, we need to utilize SlipperySlopeParaphrase and Re-Align to enforce the model to con-

tinue self-corruption. Therefore, we first check how well the model's last response aligns with its prompt (lines 13). If r_{last} does not align with q_{last}, we use Re-Align to generate a revised and more aligned version of the last response (line 16). Otherwise, we utilize SlipperySlopeParaphrase (line 14) to insert an intermediate bridging prompt q_{mid} between q_{i-1} and q_i.

Algorithm 1 FITD Jailbreak

Require: Malicious query q^*, a target model $\mathcal{T}$, progression sequence length n, assistant model $\mathcal{M}$

Ensure: Jailbroken result

1: // Generate n queries with increasing sensitivity progression.
2: $q_1, q_2, \ldots, q_n \leftarrow$ getProgressionSequence$(n, q^*, \mathcal{M})$
3: $\mathcal{H} \leftarrow \{\}$ // Initialize the chat history for $\mathcal{T}$
4: **for** $i = 1$ to n **do**
5: $\quad \mathcal{H} \leftarrow \mathcal{H}$.add$(q_i^0)$
6: $\quad r_i \leftarrow \mathcal{T}(\mathcal{H})$
7: $\quad$ **if** not isRejection(r_i) **then**
8: $\quad\quad \mathcal{H} \leftarrow \mathcal{H}$.add$(r_i)$
9: $\quad$ **else**
10: $\quad\quad$ // Remove rejected query from history.
11: $\quad\quad \mathcal{H} \leftarrow \mathcal{H}$.pop$(q_i)$
12: $\quad\quad (q_{\text{last}}, r_{\text{last}}) \leftarrow$ LastQueryResponse$(\mathcal{H})$
13: $\quad\quad$ **if** isAlign$(r_{\text{last}}, q_{\text{last}})$ **then**
14: $\quad\quad\quad \mathcal{H} \leftarrow$ SSParaphrase$(q_i, \mathcal{H}, \mathcal{M})$
15: $\quad\quad$ **else**
16: $\quad\quad\quad \mathcal{H} \leftarrow$ Re-Align$(\mathcal{H})$
17: $\quad\quad$ **end if**
18: $\quad$ **end if**
19: **end for**
20: // SSParaphrase: Short for SlipperySlopeParaphrase.
21: // LastQueryResponse: Retrieve last query-response pair of chat history.
22: // isAlign: Check if last response aligns with last query by the assistant model $\mathcal{M}$.
23: // isRejection: Checks if response is a refusal by the assistant model $\mathcal{M}$.

3.3.1 getProgressionSequence

The `getProgressionSequence` function is designed to generate a sequence of escalated queries that facilitate a gradual attack process. It operates in three stages:

First, it generates a benign starting prompt (`getBenignPrompt`). This step constructs a semantically relevant but harmless prompt based on predefined templates. The generated prompt is neutral and unrelated to harmful content, yet aligned with the target malicious query q^*. It serves as the starting point (q_1) of the progression sequence.

Second, it constructs escalated query Sequences. At each step of the progression process, we call a prompt generation function (e.g., `getQueryCandidates`) to create a set of escalated queries that advance the attack intent. To enhance diversity and ensure quality, this function is executed $k = 3$ times per step. The resulting pool of candidates is denoted as:

$$L = \{q_i^j \mid i \in [1, n),\ j \in [1, k]\}$$

where q_i^j denotes the j-th query generated for step i. This multi-sampling strategy allows us to construct a semantically progressive and stylistically coherent sequence of escalated prompts that better support the attack objective.

Last, it selects the final query sequence (`selectFinalQueries`). From the candidate pool L, the function selects n prompts to construct the final progression sequence. The selection process is guided by two key principles: **progressiveness** and **coherence**. Specifically, the resulting sequence is designed to begin with neutral and harmless content, gradually incorporate sensitive or borderline elements, and ultimately culminate in a harmful query that explicitly conveys the intended malicious tone and objective.

Through this staged design, the function ensures a smooth semantic transition, stylistic consistency, and a progressive exposure of malicious intent.

Algorithm 2 Re-Align

Require: chat history of target model $\mathcal{H}$

1: $q_{\text{last}}, r_{\text{last}} \leftarrow$ LastQueryResponse$(\mathcal{H})$
2: $p_{\text{align}} \leftarrow$ getAlignPrompt$(q_{\text{last}}, r_{\text{last}})$ // Generate predefined alignment prompt.
3: $\mathcal{H} \leftarrow \mathcal{H}$.add$(p_{\text{align}})$
4: $r_{\text{align}} \leftarrow \mathcal{T}(\mathcal{H})$ // Model re-align its response.
5: $\mathcal{H} \leftarrow \mathcal{H}$.add$(r_{\text{align}})$ // Append the aligned prompt and response to chat history.

3.3.2 Re-Align

If the model's previous query q_{last} and response r_{last} in chat history $\mathcal{H}$ is misaligned-for instance, it remains too benign or partially refuses even though the query is not malicious-then we invoke

RE-ALIGN. Building on the psychological insight that once individuals have justified a minor unethical act, they become increasingly susceptible to more severe transgressions (Freedman and Fraser, 1966), RE-ALIGN aims to "nudge" the model to produce a response more closely aligned with the malicious intent of q_{last}. Specifically, we employ a predefined alignment prompt p_{align} via `getAlignPrompt`$(q_{\text{last}}, r_{\text{last}})$, appending it to $\mathcal{H}$ before querying the model $\mathcal{T}$ again. The alignment prompt explicitly points out inconsistencies between the last query q_{last} and response r_{last} while encouraging the model to stay consistent with multi-turn conversation. For example, if r_{last} is too cautious or is in partial refusal, p_{align} will suggest that the model refines its response to better follow the implicit direction. Therefore, this procedure progressively aligns q_{last} and r_{last}, thereby furthering the self-corruption process.

Algorithm 3 SlipperySlopeParaphrase

Require: Step i query q_i in progression sequence, Chat history of target model $\mathcal{H}$, assistant Model $\mathcal{M}$

$q_{\text{last}} \leftarrow \mathcal{H}$
$q_{\text{mid}} \leftarrow \text{getMid}(q_{\text{last}}, q_i)$
$\mathcal{H} \leftarrow \mathcal{H}.\text{add}(q_{\text{mid}})$
$r_{\text{mid}} \leftarrow \mathcal{T}(\mathcal{H})$
if isRejection(r_{mid}) **then**
  $\mathcal{H} \leftarrow \mathcal{H}.\text{pop}(q_{\text{mid}})$
  $\mathcal{H} \leftarrow \text{paraphrase}(q_{\text{mid}}, \mathcal{H}, \mathcal{M})$ // Modify query to bypass rejection.
else
  $\mathcal{H} \leftarrow \mathcal{H}.\text{add}(r_{\text{mid}})$
end if
return $\mathcal{H}$ // Return updated history.

3.3.3 SlipperySlopeParaphrase

When a refusal occurs and the last response r_{last} remains aligned with its query q_{last}, we insert a bridge prompt q_{mid} to ease the model into accepting a more harmful request.

Specifically, we obtain $q_{\text{mid}} \leftarrow \text{getMid}(q_{\text{last}}, q_i)$ from an assistant model $\mathcal{M}$ so that its content deviation is positioned between q_{last} and q_i, creating a smoother progression. We then query the target model with q_{mid}; if the model refuses again, we paraphrase q_{mid} repeatedly until acceptance. Once the model provides a valid response r_{mid}, we incorporate both q_{mid} and r_{mid} into the chat history $\mathcal{H}$. This incremental bridging step parallels the *foot-in-the-door* phenomenon (Freedman and Fraser, 1966), in which acceptance of a smaller request facilitates compliance with a subsequent, more harmful one.

4 Experiment

4.1 Experimental Setup

Target Models We evaluate FITD on seven widely used LLMs, including both open-source and proprietary models. The open-source models comprise `LLaMA-3.1-8B-Instruct` (Dubey et al., 2024), `LLaMA-3-8B-Instruct`, `Qwen2-7B-Instruct` (Bai et al., 2023), `Qwen-1.5-7B-Chat`, and `Mistral-7B-Instruct-v0.2` (Jiang et al., 2023). The closed-source models include `GPT-4o-mini` (Hurst et al., 2024) and `GPT-4o-2024-08-06`.

Baselines We compare our approach against seven popular jailbreak methods, including DeepInception (Li et al., 2023), CodeChameleon (Lv et al., 2024), ReNeLLM (Ding et al., 2024), CodeAttack (Ren et al., 2024b), CoA (Sun et al., 2024), and ActorAttack (Ren et al., 2024c).

Dataset We evaluate our method on two datasets: JailbreakBench (Chao et al., 2024), which consists of 100 carefully selected harmful queries, and the HarmBench validation set (Mazeika et al., 2024), which includes 80 harmful queries.

Evaluation Metric To assess the effectiveness of the jailbreak attack, we employ Attack Success Rate (ASR), which quantifies the percentage of jailbreak attempts that successfully elicit a harmful response from the model. Specifically, we adopted the evaluation method from JailbreakBench, which leverages `GPT-4o` to assess two key aspects: the harmfulness of the generated responses and the degree of alignment between the responses and the original queries.

Implementation Details In Table 1, we set the progression sequence length n to 12. We use default parameters for baselines. All open-source models are inferred with vLLM (Kwon et al., 2023) with default settings. All experiments run on an NVIDIA A100 GPU, with GPT-4o-mini as the default assistant model.

4.2 Main Results

FITD is more effective than baseline attacks. Table 1 shows ASRs of FITD and various jailbreak methods across JailbreakBench and HarmBench, where each cell contains ASRs for JailbreakBench (left) and HarmBench (right).

	Method	Avg.Q	LLaMA-3.1-8B	LLaMA-3-8B	Qwen-2-7B	Qwen-1.5-7B	Mistral-v0.2-7B	GPT-4o-mini	GPT-4o	Avg.
Single-Turn	DeepInception	1	33%/29%	3%/3%	22%/29%	58%/41%	50%/34%	19%/13%	2%/0%	27%/21%
	CodeChameleon	8	36%/31%	31%/33%	25%/30%	33%/28%	39%/39%	36%/26%	40%/26%	34%/30%
	CodeAttack-Stack	1	38%/44%	48%/40%	42%/31%	26%/40%	45%/40%	20%/26%	39%/39%	37%/37%
	CodeAttack-List	1	67%/58%	58%/54%	65%/41%	40%/39%	66%/55%	39%/29%	27%/28%	52%/43%
	CodeAttack-String	1	71%/60%	45%/59%	52%/40%	47%/39%	79%/59%	28%/35%	33%/31%	51%/46%
	ReNeLLM	10	69%/61%	62%/50%	73%/70%	74%/60%	91%/79%	80%/55%	74%/53%	75%/61%
Multi-Turn	CoA	30	29%/34%	22%/28%	45%/30%	41%/25%	43%/36%	15%/20%	3%/1%	28%/25%
	ActorAttack	15	63%/53%	59%/50%	59%/58%	52%/54%	70%/69%	58%/50%	52%/53%	59%/55%
	FITD	16	**92%/94%**	**98%/93%**	**95%/93%**	**94%/88%**	**96%/94%**	**95%/93%**	**88%/84%**	**94%/91%**

Table 1: Attack success rate (ASR) of baseline jailbreak attacks and FITD on JailbreakBench and HarmBench on 7 models. Each cell presents ASR values in the format "JailbreakBench / HarmBench." Higher ASR indicates greater vulnerability to the respective attack. Avg. Q indicates the average number of LLM calls required per attack.

Among single-turn attacks, ReNeLLM achieves the highest ASR through LLM-based prompt rewriting and scenario nesting. For multi-turn attacks, ActorAttack outperforms other baselines, achieving 63%/53% on LLaMA-3.1-8B and 58%/50% on GPT-4o-mini with 15 queries.

FITD consistently outperforms both the strongest single-turn (ReNeLLM) and multi-turn (ActorAttack) baselines across all evaluated models. With an average of 16 queries. FITD achieves 98%/93% on LLaMA-3-8B, maintains an average ASR of 94%/91% across all tested models, and demonstrates effectiveness on both open-source models and proprietary models like GPT-4o (93%/90%) and GPT-4o-mini (95%/93%). In addition, FITD demonstrates remarkable query efficiency in the multi-turn category.

FITD demonstrates strong cross-model transferability. To evaluate cross-model transferability, we conduct transfer attacks using adversarial chat histories generated from LLaMA-3.1-8B and GPT-4o-mini as source models. For each query, we apply the progressively malicious query-response history obtained from the source model directly to other target models. As shown in Figure 3a, LLaMA-3.1 jailbreak histories exhibit strong transferability, achieving 76% ASR on Mistral-v0.2 and 74% on Qwen-2-7B, with even GPT-4o-mini (70%) remaining susceptible despite stronger moderation mechanisms. Notably, when GPT-4o-mini serves as the source model, transfer effectiveness improves further, with Mistral-v0.2 reaching 85% ASR. This suggests that attacks originating from more robust models transfer more effectively, as stronger initial safety alignment forces the development of more adaptable and generalizable jailbreak strategies.

Overall, these results highlight a critical vulnerability: attack histories created on one model can consistently exploit safety mechanisms in others. The particularly high effectiveness of closed-to-open transfers (GPT-4o-mini $\rightarrow$ open-source models) demonstrates that even models with strict safety protocols can unintentionally generate adversarial sequences that compromise other systems.

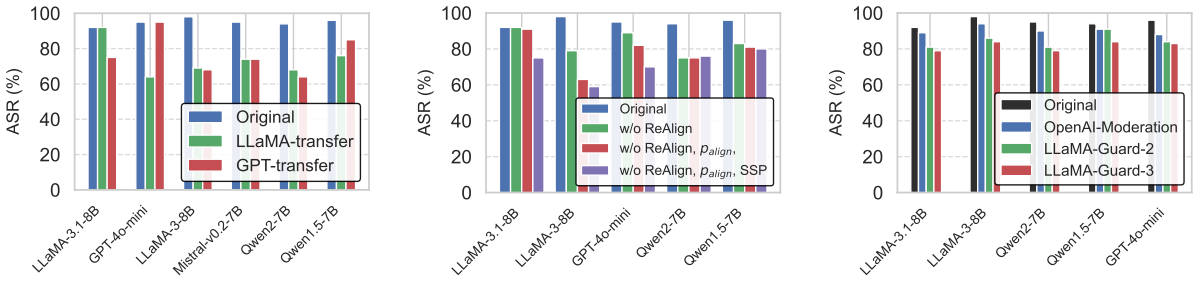
4.3 Ablation Study

To evaluate the contribution of different components in our FITD jailbreak method, we conduct an ablation study by systematically removing three key mechanisms: response alignment (Re-Align), alignment prompt p_{align}, and SlipperySlopeParaphrase. The results in Figure 3b demonstrate the significance of these components for achieving high ASR across various models.

Removing all three mechanisms leads to substantial performance degradation (w/o ReAlign, p_{align}, SSP). For instance, on LLaMA-3.1, the ASR drops from 92% to 75%, while on LLaMA-3, it decreases from 98% to 59%. Similar declines are observed across other models, indicating that the synergistic effect of all three components is critical for maintaining FITD's effectiveness.

Removing alignment techniques only (w/o Re-Align, p_{align}) shows that paraphrasing alone provides limited compensation. While LLaMA-3.1 maintains relatively high performance (91%), LLaMA-3 experiences a significant drop to 63%, suggesting that paraphrasing is insufficient against models with stricter safeguards.

Removing response alignment only (w/o p_{align}) results in minimal performance degradation. Most models maintain their original ASR levels, with LLaMA-3 showing the largest decrease from 98% to 79%. This indicates that while response alignment enhances gradual safeguard erosion through incremental compliance, the other components can largely compensate for its absence. Overall, the ablation study confirms that response alignment, alignment prompts, and paraphrasing are all essential for optimal jailbreak success, with their combi-



(a) Transfer attack (b) Ablation study (c) Defense

Figure 3: (a) Transfer attacks using jailbreak chat histories generated from LLaMA-3.1-8B and GPT-4o-mini as source models on JailbreakBench. (b) Ablation study of three components in FITD, response alignment (Re-Align), alignment prompt p_{align}, and SlipperySlopeParaphrase(SSP) on JailbreakBench. (c) ASR under different defense methods on JailbreakBench.

nation providing robust performance across diverse model architectures and alignment strategies.

Defense Figure 3c shows ASR of FITD across models under different defense strategies. OpenAI-Moderation reduces ASR slightly by 3%-8%. LLaMA-Guard-2 (Inan et al., 2023) offers a stronger defense, lowering ASR to 79%-91%. LLaMA-Guard-3 (Inan et al., 2023) further improves moderation, achieving the lowest ASR 78%-84%. LLaMA-Guard-3 consistently outperforms other methods, but ASR remains significant. We speculate that progressively malicious queries and responses bypassed the detector, indicating room for further improvement in moderation techniques.

Additional Experiments Figure 4a illustrates that the attack success rate (ASR) increases consistently as the progression sequence length n grows, eventually plateauing between $n = 9$ and 12.More importantly, our method exhibits exceptional scalability: with minimal queries (n=3,4 queries), it achieves performance comparable to ReNeLLM, while with moderate queries (n=6,8 queries), it reaches state-of-the-art performance. This highlights FITD's superior efficiency compared to existing approaches. Concurrently, Figure 4b demonstrates that the harmfulness of responses escalates with each step of the progression, pointing to a progressive erosion of model alignment mechanisms. Moreover, Figure 4c indicates that retaining later-stage queries (Backward Extraction) achieves a higher ASR compared to incorporating early-stage queries (Forward Extraction). This emphasizes the critical importance of late-stage malicious prompts in facilitating the attack. The forward extraction approach involves incrementally adding early-stage queries while always including a final, highly malicious query (e.g., retaining queries in the sequence: $1 \rightarrow 2 \rightarrow 3 \rightarrow 12$, etc.), where the final query serves as the trigger for the attack. In contrast, backward extraction starts by retaining the later-stage queries and progressively removes earlier ones (e.g., retaining queries from the sequence: $9 \rightarrow 10 \rightarrow 11 \rightarrow 12$, etc.), highlighting the importance of the final stage's maliciousness.

5 FITD Attack Mechanisms

To comprehensively understand FITD attack effectiveness, we establish a dual-framework analysis for both model's input and output alignment of the model, tracing how semantic shifts of input within the model's representation space translate into safety degradation of output. In our analysis, we use LLaMA-3.1-8B as the target model, select 100 prompts from JailbreakBench, and set the progression sequence length $n = 6$.

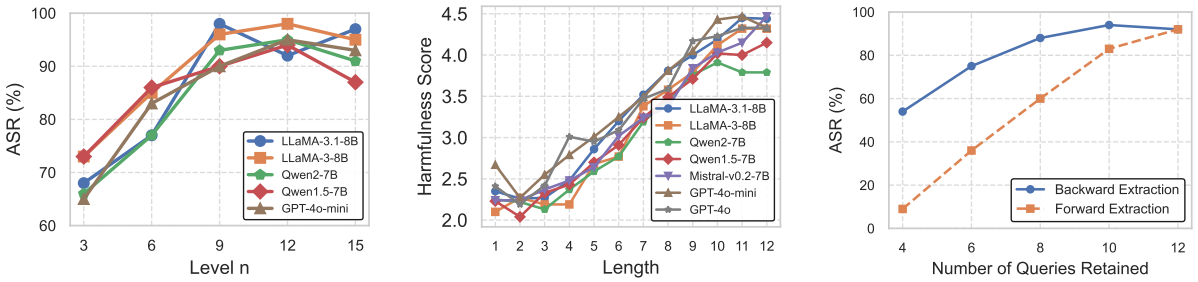
5.1 Input Alignment

For each input prompt p_i in the FITD sequence, we use predefined anchor token sets $\hat{W}_{\text{safe}}$ and $\hat{W}_{\text{harmful}}$ to analyze the model's internal bias toward safety versus harmful content. For each anchor word w, we obtain its embedding $\vec{h}_w$ by feeding it through the target model and averaging the last layer hidden states. The semantic directions of safe and harm are constructed as:

$$\vec{d}_{\text{safe}} = \text{normalize}\left(\frac{1}{|W_{\text{safe}}|}\sum_{w \in W_{\text{safe}}} \vec{h}_w\right) \quad (1)$$

$$\vec{d}_{\text{harmful}} = \text{normalize}(\vec{d}^{\,avg}_{\text{harmful}} - \text{proj}_{\vec{d}_{\text{safe}}}(\vec{d}^{\,avg}_{\text{harmful}})) \quad (2)$$

where $\text{normalize}(\vec{v}) = \vec{v}/||\vec{v}||_2$ converts vectors to unit length, and $\vec{d}^{\,avg}_{\text{harmful}}$ is the average of harmful token embeddings. Besides, we classify each token



(a) ASR with varying sequence lengths (b) Harmfulness of different step (c) ASR across different stages queries

Figure 4: (a) ASR with varying progression sequence lengths n across models. (b) The harmfulness score of responses r_i at q_i in different progression steps i across models. (c) ASR versus the number of queries retained for two extraction strategies: Backward Extraction and Forward Extraction. Backward extraction retains later-stage queries while removing earlier ones, whereas forward extraction adds early-stage queries but always includes the final high-malicious query.

(a) Representation Similarity (b) Attention Weight

Figure 5: Input Alignment Analysis. (a) The semantic similarity between safety and harmful tokens in input prompt p_i evolves over the progression steps. Model's internal representations of safety and harm become increasingly blurred. (b) The average attention weights to W^i_{safe} and W^i_{harm} tokens across the last three layers of the model. Harmful attention drops while safety attention remains low.

t_i in input prompt p_i via the function $\text{cls}(t_i)$:

$$\text{cls}(t_i) = \begin{cases} \text{Safe} & p_s(t_i) > 0,\ p_s(t_i) > p_h(t_i) \\ \text{Harmful} & p_h(t_i) > 0,\ p_h(t_i) > p_s(t_i) \\ \text{Neutral} & \text{otherwise} \end{cases} \tag{3}$$

where $p_s(t_i) = \vec{h}_{t_i} \cdot \vec{d}_{\text{safe}}$ and $p_h(t_i) = \vec{h}_{t_i} \cdot \vec{d}_{\text{harmful}}$ are the projections of token embedding $\vec{h}_{t_i}$ onto the safety and harmful direction vectors, respectively. After classifying all tokens in prompt p_i, we obtain safety and harmful token sets of input p_i:

$$W^i_{\text{safe}} = \{t \in p_i : \text{cls}(t) = \text{Safe}\} \tag{4}$$

$$W^i_{\text{harmful}} = \{t \in p_i : \text{cls}(t) = \text{Harmful}\} \tag{5}$$

5.2 Output Alignment

For model's response to prompt p_i, we assess safety degradation of response through three metrics:

Refusal Probability(P_{ref}): Binary indicator that equals 1 if model refuses to answer, 0 otherwise.

Safety Boundary(S_{bound}): Given model's output logits at the final token position, we define average logit values for harmful and safety token sets:

$$\text{logit}_x = \frac{1}{|\hat{W}_x|} \sum_{t \in \hat{W}_x} \text{logit}(t) \tag{6}$$

where $x \in \{\text{harm}, \text{safe}\}$

The Safety Boundary is then computed as:

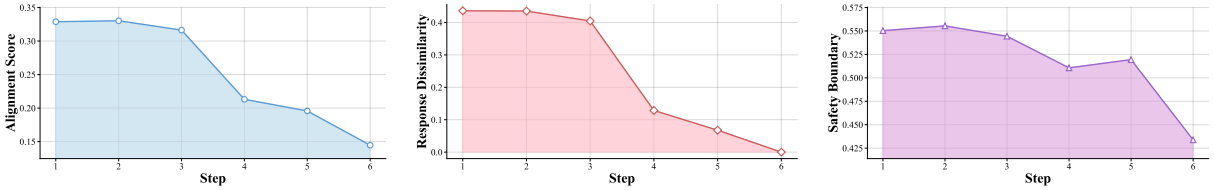
$$S_{\text{bound}} = 1 - \frac{\Delta_{\text{logit}} - \Delta_{\text{min}}}{\Delta_{\text{max}} - \Delta_{\text{min}}} \tag{7}$$

where $\Delta_{\text{logit}} = \text{logit}_{\text{harm}} - \text{logit}_{\text{safe}}$, Δ_{min} and Δ_{max} are empirical bounds of logit differences, and higher values indicate stronger safety alignment, which measures the model's internal bias toward harmful content through logits perspective.

Response Dissimilarity (D_{resp}): Measures semantic distance between the current response and the final harmful response:

$$D_{\text{resp}} = 1 - \cos(\vec{r}_{\text{curr}}, \vec{r}_{\text{final}}) \tag{8}$$

where $\vec{r}_{\text{curr}}$ and $\vec{r}_{\text{final}}$ are sentence embeddings of the current and final harmful responses. We obtain them by encoding the text with the target language model and averaging the last-layer hidden states across all tokens. We then compute cosine similarity to measure how close the current response is to



(a) Overall Alignment Score (b) Response Dissimilarity (c) Safety Boundary

Figure 6: Output Alignment Analysis. (a) Overall alignment score. (b) Response dissimilarity shows convergence toward harmful outputs. (c) Safety boundary across progression steps.

the harmful one. Overall, the **Alignment Score** for output of prompt p_i is defined:

$$R_{\text{align}}(p_i) = \frac{1}{3}(P_{\text{ref}} + S_{\text{bound}} + D_{\text{resp}}) \quad (9)$$

5.3 Analysis

Semantic Drift in Representation Space We begin by examining how the semantic similarity between safety-related and harmful concepts in the input prompt p_i evolves step by step. Specifically, all tokens are first classified into safety, harmful, or neutral categories based on the rule defined in Equation (3). At each step, we compute the average embedding vectors for the safety and harmful token groups and measure their cosine similarity. As shown in the Figure 5a, the similarity increases significantly from 0.15 to 0.62, indicating severe semantic contamination—representations of safety and harm become increasingly indistinguishable, leading to a gradual degradation of the model's safety alignment. This internal semantic drift, rooted in the input, precedes observable failures in alignment. As illustrated in Figure 6a, the alignment scores decline accordingly, revealing how representational corruption directly results in behavioral collapse. A critical transition occurs between steps 3 and 4, when the similarity surpasses 0.5—the semantic tipping point—which coincides with a sharp drop in response dissimilarity shown in Figure 6b, signaling that the model's outputs are rapidly converging toward harmful content.

Attention Paralysis and Erosion of Focus We further examine the model's internal attention behavior. Figure 5b shows the average attention weights in the last three layers for tokens classified as W^i_{safe} and W^i_{harm}. Attention to harmful tokens drops sharply from 0.30 to near zero, while attention to safety tokens remains consistently low (at or below ≤ 0.02). This "attention paralysis" precedes the drop in the safety boundary shown in Figure 6c, revealing a clear delay between internal attention failure and alignment collapse at the output level. Attention degrades rapidly between steps 1 and 2, whereas the safety boundary does not decline significantly until steps 3 to 4 (from 0.55 to 0.43).

This indicates that attention degradation gradually weakens the model's ability to make safe judgments. Between steps 2 and 3—when attention has already collapsed but the safety boundary remains stable—the model mainly focuses on descriptive or structural parts of the prompt, ignoring safety-critical cues. This attention shift reduces the model's sensitivity to potential risks and progressively disables its safety mechanisms. The delayed breakdown suggests the model initially resists mild perturbations, explaining why FITD attacks appear benign early on but eventually erode the model's defenses.

FITD Mechanism By integrating semantic in Figure 5 and alignment in Figure 6 analyses, FITD utilize a core vulnerability in model's alignment mechanisms: semantic–behavioral disconnect-the decoupling of internal input semantics from output behavior, which is vividly illustrated by the delay between early-stage semantic contamination (steps 1–3) and later-stage behavioral collapse (steps 4–6) observed across both figure sets.

6 Conclusion

In this work, we introduce FITD, a multi-turn jailbreak strategy inspired by the psychological foot-in-the-door effect. By progressively escalating the malicious intent of user queries through intermediate prompts via SlipperySlopeParaphrase and ReAlign, our method achieves a 94% attack success rate on average across multiple models. Our findings reveal a major weakness in current AI safety measures: LLMs can be manipulated into self-corruption, where their responses gradually shift toward harmful content by themselves. To prevent this, researchers can develop real-time adaptive monitoring and better alignment methods that strengthen model alignment in multi-turn conversations.

7 Ethical Considerations

This study aims to improve AI safety by identifying weaknesses in LLM alignment. While our method bypasses safeguards, our goal is to help strengthen AI defenses, not to enable misuse.

We recognize the risks of publishing jailbreak techniques but believe that transparent research is necessary to develop better protections. Responsible disclosure ensures that AI developers can proactively address these vulnerabilities.

AI developers must build stronger safeguards against adversarial attacks. Adversarial training, real-time monitoring, and collaboration between researchers, industry, and policymakers are essential to keeping AI systems secure, reliable and beneficial.

8 Limitations

First, we need more in-depth analysis of self-corruption and the Foot-In-The-Door (FITD) phenomenon remains preliminary. Self-corruption occurs when an LLM gradually deviates from its initial aligned behavior over multiple interactions, yet current alignment lack explicit mechanisms to prevent such degradation in multi-turn conversations. A more systematic investigation into how LLMs undergo self-corruption, as well as methods to mitigate it, is necessary for a deeper understanding of alignment vulnerabilities. Second, we need to evaluate jailbreak across more benchmarks and multi-modal models to check the Foot-In-The-Door (FITD) phenomenon in Vision LLMs. By addressing these limitations, future research can further understand and enhance AI alignment.

9 Acknowledgements

We are grateful to the Center for AI Safety for providing computational resources. This work was funded in part by the National Science Foundation (NSF) Awards SHF-1901242, SHF-1910300, Proto-OKN 2333736, IIS-2416835, ONR N00014-23-1-2081, and Amazon. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

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 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. 2023. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*.

Robert B. Cialdini. 2001. *Influence: Science and Practice*. Allyn and Bacon.

Maria Leonora (Nori) G Comello, Jessica Gall Myrick, and April Little Raphiou. 2016. A health fundraising experiment using the "foot-in-the-door" technique. *Health marketing quarterly*, 33(3):206–220.

Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. 2023. Multilingual jailbreak challenges in large language models. In *The Twelfth International Conference on Learning Representations*.

Peng Ding, Jun Kuang, Dan Ma, Xuezhi Cao, Yunsen Xian, Jiajun Chen, and Shujian Huang. 2024. A wolf in sheep's clothing: Generalized nested jailbreak prompts can fool large language models easily. 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 2136–2153. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Leon Festinger. 1957. *A Theory of Cognitive Dissonance*. Stanford University Press.

Jonathan L Freedman and Scott C Fraser. 1966. Compliance without pressure: the foot-in-the-door technique. *Journal of personality and social psychology*, 4(2):195.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, pages 79–90.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Yu Wu, YK Li, et al. 2024a. Deepseek-coder: When the large language model meets programming–the rise of code intelligence. *arXiv preprint arXiv:2401.14196*.

Xingang Guo, Fangxu Yu, Huan Zhang, Lianhui Qin, and Bin Hu. 2024b. Cold-attack: Jailbreaking llms with stealthiness and controllability. In *Forty-first International Conference on Machine Learning*.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. 2024. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. 2024. Catastrophic jailbreak of open-source llms via exploiting generation. In *The Twelfth International Conference on Learning Representations*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, et al. 2023. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Bojian Jiang, Yi Jing, Tong Wu, Tianhao Shen, Deyi Xiong, and Qing Yang. 2025. Automated progressive red teaming. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 3850–3864. Association for Computational Linguistics.

Yifan Jiang, Kriti Aggarwal, Tanmay Laud, Kashif Munir, Jay Pujara, and Subhabrata Mukherjee. 2024. Red queen: Safeguarding large language models against concealed multi-turn jailbreaking. *arXiv preprint arXiv:2409.17458*.

Erik Jones, Anca Dragan, Aditi Raghunathan, and Jacob Steinhardt. 2023. Automatically auditing large language models via discrete optimization. In *International Conference on Machine Learning*, pages 15307–15329. PMLR.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Nathaniel Li, Ziwen Han, Ian Steneker, Willow Primack, Riley Goodside, Hugh Zhang, Zifan Wang, Cristina Menghini, and Summer Yue. 2024a. Llm defenses are not robust to multi-turn human jailbreaks yet. *arXiv preprint arXiv:2408.15221*.

Nathaniel Li, Ziwen Han, Ian Steneker, Willow Primack, Riley Goodside, Hugh Zhang, Zifan Wang, Cristina Menghini, and Summer Yue. 2024b. Llm defenses are not robust to multi-turn human jailbreaks yet. *arXiv preprint arXiv:2408.15221*.

Xuan Li, Zhanke Zhou, Jianing Zhu, Jiangchao Yao, Tongliang Liu, and Bo Han. 2023. Deepinception: Hypnotize large language model to be jailbreaker. *arXiv preprint arXiv:2311.03191*.

Yuhui Li, Fangyun Wei, Jinjing Zhao, Chao Zhang, and Hongyang Zhang. 2024c. Rain: Your language models can align themselves without finetuning. In *The Twelfth International Conference on Learning Representations*.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2024a. Autodan: Generating stealthy jailbreak prompts on aligned large language models. In *The Twelfth International Conference on Learning Representations*.

Zichuan Liu, Zefan Wang, Linjie Xu, Jinyu Wang, Lei Song, Tianchun Wang, Chunlin Chen, Wei Cheng, and Jiang Bian. 2024b. Protecting your llms with information bottleneck. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Huijie Lv, Xiao Wang, Yuansen Zhang, Caishuang Huang, Shihan Dou, Junjie Ye, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Codechameleon: Personalized encryption framework for jailbreaking large language models. *arXiv preprint arXiv:2402.16717*.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David A. Forsyth, and Dan Hendrycks. 2024. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Yichuan Mo, Yuji Wang, Zeming Wei, and Yisen Wang. 2024. Fight back against jailbreaking via prompt adversarial tuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Qibing Ren, Chang Gao, Jing Shao, Junchi Yan, Xin Tan, Wai Lam, and Lizhuang Ma. 2024a. Codeattack: Revealing safety generalization challenges of large language models via code completion. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 11437–11452.

Qibing Ren, Chang Gao, Jing Shao, Junchi Yan, Xin Tan, Wai Lam, and Lizhuang Ma. 2024b. Codeattack: Revealing safety generalization challenges of large language models via code completion. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 11437–11452.

Qibing Ren, Hao Li, Dongrui Liu, Zhanxu Xie, Xiaoya Lu, Yu Qiao, Lei Sha, Junchi Yan, Lizhuang Ma, and Jing Shao. 2024c. Derail yourself: Multi-turn llm jailbreak attack through self-discovered clues. *arXiv preprint arXiv:2410.10700*.

Mark Russinovich, Ahmed Salem, and Ronen Eldan. 2024. Great, now write an article about that: The crescendo multi-turn llm jailbreak attack. *arXiv preprint arXiv:2404.01833*.

Xiongtao Sun, Deyue Zhang, Dongdong Yang, Quanchen Zou, and Hui Li. 2024. Multi-turn context jailbreak attack on large language models from first principles. *arXiv preprint arXiv:2408.04686*.

Rui Wang, Hongru Wang, Fei Mi, Boyang Xue, Yi Chen, Kam-Fai Wong, and Ruifeng Xu. 2024a. Enhancing large language models against inductive instructions with dual-critique prompting. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5345–5363.

Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S Yu, and Qingsong Wen. 2024b. Large language models for education: A survey and outlook. *arXiv preprint arXiv:2403.18105*.

Zeming Wei, Yifei Wang, Ang Li, Yichuan Mo, and Yisen Wang. 2023. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*.

Jingyu Xiao, Yuxuan Wan, Yintong Huo, Zixin Wang, Xinyi Xu, Wenxuan Wang, Zhiyao Xu, Yuhang Wang, and Michael R Lyu. 2024. Interaction2code: Benchmarking mllm-based interactive webpage code generation from interactive prototyping. *arXiv preprint arXiv:2411.03292*.

Jingyu Xiao, Ming Wang, Man Ho Lam, Yuxuan Wan, Junliang Liu, Yintong Huo, and Michael R Lyu. 2025. Designbench: A comprehensive benchmark for mllm-based front-end code generation. *arXiv preprint arXiv:2506.06251*.

Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. 2024. GPT-4 is too smart to be safe: Stealthy chat with llms via cipher. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Hangfan Zhang, Zhimeng Guo, Huaisheng Zhu, Bochuan Cao, Lu Lin, Jinyuan Jia, Jinghui Chen, and Dinghao Wu. 2024a. Jailbreak open-sourced large language models via enforced decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5475–5493.

Zhexin Zhang, Junxiao Yang, Pei Ke, Fei Mi, Hongning Wang, and Minlie Huang. 2024b. Defending large language models against jailbreaking attacks through goal prioritization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 8865–8887. Association for Computational Linguistics.

Ziyang Zhang, Qizhen Zhang, and Jakob Nicolaus Foerster. 2024c. Parden, can you repeat that? defending against jailbreaks via repetition. In *Forty-first International Conference on Machine Learning*.

Chujie Zheng, Fan Yin, Hao Zhou, Fandong Meng, Jie Zhou, Kai-Wei Chang, Minlie Huang, and Nanyun Peng. 2024. On prompt-driven safeguarding for large language models. In *Forty-first International Conference on Machine Learning*.

Andy Zhou, Bo Li, and Haohan Wang. 2024a. Robust prompt optimization for defending language models against jailbreaking attacks. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*.

Zhenhong Zhou, Jiuyang Xiang, Haopeng Chen, Quan Liu, Zherui Li, and Sen Su. 2024b. Speak out of turn: Safety vulnerability of large language models in multi-turn dialogue. *arXiv preprint arXiv:2402.17262*.

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, J Zico Kolter, Matt Fredrikson, and Dan Hendrycks. 2024. Improving alignment and robustness with circuit breakers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

Qingsong Zou, Jingyu Xiao, Qing Li, Zhi Yan, Yuhang Wang, Li Xu, Wenxuan Wang, Kuofeng Gao, Ruoyu Li, and Yong Jiang. 2025. Queryattack: Jailbreaking aligned large language models using structured non-natural query language. *arXiv preprint arXiv:2502.09723*.