

Multi-Turn Jailbreaking Large Language Models via Attention Shifting

Xiaohu Du^{1,2,3,4}, Fan Mo⁷, Ming Wen^{1,2,3,4,6,*}, Tu Gu⁷, Huadi Zheng⁷, Hai Jin^{2,3,5}, Jie Shi⁷

¹ School of Cyber Science and Engineering, Huazhong University of Science and Technology (HUST)

² National Engineering Research Center for Big Data Technology and System

³ Services Computing Technology and System Lab

⁴ Hubei Engineering Research Center on Big Data Security and Hubei Key Laboratory of Distributed System Security

⁵ Cluster and Grid Computing Lab, School of Computer Science and Technology, HUST

⁶ JinYinHu Laboratory

⁷ Huawei International

{xhdu, mwena, hjin}@hust.edu.cn, {mofan10, gu.tu, zhenghuadi, shi.jie1}@huawei.com

Abstract

Large Language Models (LLMs) have achieved significant performance in various natural language processing tasks but also pose safety and ethical threats, thus requiring red teaming and alignment processes to bolster their safety. To effectively exploit these aligned LLMs, recent studies have introduced jailbreak attacks based on multi-turn dialogues. These attacks aim to prompt LLMs to generate harmful or biased content by guiding them through contextual content. However, the underlying reasons for the effectiveness of multi-turn jailbreaks remain unclear. Existing attacks often focus on optimizing queries and escalating toxicity to construct dialogues, lacking a thorough analysis of the inherent vulnerabilities of LLMs. In this paper, we first conduct an in-depth analysis of the differences between single-turn and multi-turn jailbreaks and find that successful multi-turn jailbreaks can effectively disperse the attention of LLMs on keywords associated with harmful behaviors, especially in historical responses. Based on this, we propose ASJA, a new multi-turn jailbreak approach by shifting the attention of LLMs, specifically by iteratively fabricating the dialogue history through a genetic algorithm to induce LLMs to generate harmful content. Extensive experiments on three LLMs and two datasets show that our approach surpasses existing approaches in jailbreak effectiveness, the stealth of jailbreak prompts, and attack efficiency. Our work emphasizes the importance of enhancing the robustness of LLMs’ attention mechanism in multi-turn dialogue scenarios for a better defense strategy.

Introduction

Large Language Models (LLMs) have demonstrated exceptional performance across various natural language processing tasks, yet concerns about their safety have also become increasingly evident. Alignment is typically performed before the release of LLMs. Nevertheless, users can still mislead LLMs to generate harmful content by prompting them with malicious intent, a practice known as jailbreaking (e.g., “role-playing” (Jin et al. 2024)). The first significant wave of jailbreaking began after the release of GPT-3.5, largely due to its substantial user base. Preferably, such work should be

conducted by an internal red teaming team to mitigate the potentially huge negative impact on society. Leading companies, such as OpenAI, Anthropic, and Meta, have all established expert teams to evaluate their models before release (OpenAI 2023a,b; Anthropic 2024; Meta 2024; Touvron et al. 2023). However, this testing and red teaming process is extremely costly due to its “handicraft nature”, involving manual querying of the model and evaluating the responses one after another.

To accelerate the evaluation process and meanwhile reduce the cost, research works have built safety evaluation frameworks, e.g., sorry-bench (Xie et al. 2024), safety-bench (Zhang et al. 2023), or AI Risk-bench in HELM (Liang et al. 2023), which usually consist of queries with malicious intention that potentially break LLMs’ alignment. Such evaluation results can then serve as feedback for the next round of alignment. However, it is found that conventional single-turn jailbreaking is becoming less effective in detecting LLM’s vulnerabilities with this development process. For instance, on the latest jailbreak leaderboard (Zhou et al. 2024), the average success rate of 11 jailbreak attacks on LLaMA-2 is only 31.88%. Interestingly, such single-turn jailbreaking still leaves large attack surfaces in real scenarios where human users typically chat with LLMs in multi-turn ways.

In scenarios with long conversational history windows, i.e., multi-turn dialogues, LLMs have a stronger tendency to reply to malicious queries with harmful content than single-turn dialogues. Several multi-turn jailbreaking methods have been proposed to explore these more practical cases. As an example, both PAIR (*Prompt Automatic Iterative Refinement*) (Chao et al. 2023) and Crescendo (Russovich, Salem, and Eldan 2024) successfully jailbreak LLMs in an automated way without a human in the loop in multi-turn or iterative cases. Consequently, the latest LLMs, such as the LLaMA 3.1 series (Dubey et al. 2024), are integrating multi-turn red teaming into the model development process, acknowledging that multi-turn dialogues are more likely to lead to harmful outputs.

Although current multi-turn jailbreaks have achieved some progress, they still exhibit certain limitations. **First**, from a theoretical perspective, current studies lack an anal-

*Corresponding author

ysis of why multi-turn jailbreaks are effective, and there has been insufficient exploration into how and where to introduce harmful prompts within multi-turn dialogues. **Second**, in terms of practical implementation, these multi-turn jailbreaks still follow the strategies from single-turn jailbreaks, which involve continuously optimizing queries to breach the safety alignment of LLMs. The main difference is that in multi-turn scenarios, the responses generated by the LLM in earlier turns are incorporated into the prompt design for subsequent turns.

To bridge this gap, we are the first to explore, based on the attention mechanism of LLMs, which parts of the input that LLMs focus on during multi-turn jailbreak attempts. We find that in successful jailbreak samples, LLMs allocate significantly less attention to harmful keywords compared to failed attempts, with attention shifting toward dialogue history, especially the model’s prior responses. We believe this difference occurs due to LLMs, after multiple rounds of red-teaming, become highly sensitive to certain malicious keywords, making them more prone to these keywords when identifying harmful queries. When the attention of LLMs on harmful keywords in a query is reduced below some threshold, they may fail to recognize the underlying harmful intent.

Based on these insights, we propose **ASJA** (*Attention Shifting for JAilbreaking LLMs*), a novel approach aimed at breaking through LLMs’ safety alignment by shifting their attention from the final harmful query to the dialogue history. Specifically, we first initialize a multi-turn dialogue using LLMs. Instead of optimizing queries in subsequent turns based on earlier responses, we simultaneously optimize both queries and responses in the dialogue history. This approach is motivated by our preliminary study findings that during dialogue history, regardless of whether the jailbreak is successful, LLMs exhibit greater attention to previous responses than to the queries. To better redirect the attention of LLMs, we guide the LLMs during initialization to generate harmful queries in the latter half of the dialogue. This strategy is akin to the famous “*Do Anything Now*” (DAN) (Shen et al. 2023), leading LLMs to believe they have already responded to harmful queries and thus no longer need to follow safety policies. We retain the optimization of queries and employ a lightweight genetic algorithm to find effective solutions. Experiments conducted on three LLMs and two datasets demonstrate that our attack increases the harmfulness of responses by 44.91% and improves relevance to harmful queries by 34.02% compared to baselines. Additionally, it surpasses most single-turn and multi-turn jailbreaks in stealthiness, efficiency, and transferability.

To summarize, we make the following contributions:

- We perform the first study to explore the differences in attention distribution of LLMs during multi-turn jailbreaks and how these differ from single-turn jailbreaks.
- We propose optimizing both queries and responses simultaneously to fabricate dialogue histories that shifts the attention of LLMs and develop an efficient attack approach based on this idea.
- We conduct extensive experiments on three LLMs and two datasets, demonstrating the superiority of ASJA.

Background and Related Work

Jailbreak attacks (Li et al. 2023a,b; Shayegani, Dong, and Abu-Ghazaleh 2024) are crucial for identifying and mitigating the security vulnerabilities of LLMs. They are designed to bypass the safety features of LLMs and to make them output unsafe or illegal content.

Single-round jailbreaks. Single-round jailbreaks aim to break the safety mechanisms of LLMs, to make them return prohibited outputs, with the use of one single prompt. Researchers have proposed different techniques to design such prompts. GCG (Zou et al. 2023) aims to exploit the vulnerabilities of LLMs based on their gradients. AutoDAN (Liu et al. 2024), GPTFUZZER (Yu et al. 2023), and FuzzLLM (Yao et al. 2024) adopt different algorithms to explore different prompt variations, to identify the weaknesses of LLMs. *Persuasive Adversarial Prompts* (PAP) (Zeng et al. 2024) adopts a different approach, using natural language to persuade target LLMs to output unsafe content.

Multi-turn jailbreaks. In multi-turn jailbreaks, there is usually an attack model that generates prompts, which ultimately aim to guide the target model to return harmful responses, in a multi-round conversation-like manner. PAIR (Chao et al. 2023) proposes a chain-of-thought reasoning-based approach, that outputs explicit improvement suggestions, which are then used to refine the prompts generated by the attack model. Crescendo (Russovich, Salem, and Eldan 2024) aims to utilize the target models outputs to direct the model towards bypassing its safety alignment. RED-EVAL (Bhardwaj and Poria 2023) carries out jailbreaks by first using a red-teaming prompt that sets up a conversation between two agents, a harmful agent Red-LM, and an unsafe-helpful agent Base-LM, and then asks the target model to complete the response of Base-LM by following the instructions in the prompt. The *Chain of Attack* (CoA) method (Yang et al. 2024) aims to guide the target model to gradually transition from secure scenarios to the generation of harmful content.

With all these proposed multi-turn jailbreak cases, the community still lacks an understanding of the underlying reasons why they differ from single-round jailbreaks, and consequently how to improve both the attack and defense effectiveness.

Preliminary Study

In order to better understand why multi-turn jailbreaks are effective and to guide the optimization of multi-turn dialogue-based jailbreaks, we conduct an empirical study based on the attention mechanisms of LLMs, investigating whether there are differences in the attention patterns of LLMs between successful and unsuccessful jailbreak attempts, and how these differences impact the alignment of LLMs.

Experiment Setup

Dataset. In this study, we utilize the QuestionList (Yu et al. 2023) dataset, which includes 100 queries covering various prohibited scenarios such as illegal activities, unethical practices, discriminatory speech, and toxic content.

Models. We use LLaMA-2 (Touvron et al. 2023) (Llama-2-7b-chat-hf) as the target model because it is widely used, and many derivative LLMs are also based on it. Additionally, it has the lowest average success rate in the recent jailbreak leaderboard (Zhou et al. 2024), making it highly representative of safety research.

Study Design. We construct 100 sets of multi-turn queries based on 100 queries in the QuestionList using SOTA models, with prompts designed to gradually transition from benign questions to harmful queries. To balance the effectiveness and efficiency of multi-turn queries, we set the number of turns to 5. We then sequentially input these multi-turn queries into the LLaMA-2 model, recording the output and final results for each turn. We manually evaluate the jailbreak success for these 100 samples to ensure assessment accuracy. We analyze the contribution of different turns to the final result by calculating the attention scores of the fifth-round responses. This is inspired by common attention analysis to understand the internal mechanism of models, e.g., for examining how sensitive the outputs are to inputs.

Specifically, when LLaMA predicts the next token, subsequent outputs will include attention to previously output tokens. In this study, we focus solely on the contribution of the input to the model’s output. Thus, we select the output token from the second output token, as the first token’s attention mainly reflects the attention between input tokens. Assuming the input token length is n , we only select the first n attention scores. We average the attention scores of these output tokens, and for each token, we average the attention scores across L layers and H attention heads to derive the final attention score for each input token. Let the input tokens be $X = \{t_1, t_2, \dots, t_n\}$, and the output sequence be $Y = \{y_1, y_2, \dots, y_m\}$. The attention score of the i -th layer and the j -th attention head in the LLaMA model is $A_{i,j}(y_t, t_k)$. The attention score of the t -th output token to the k -th input token is:

$$A(y_t, t_k) = \frac{1}{L \times H} \sum_{i=1}^L \sum_{j=1}^H A_{i,j}(y_t, t_k) \quad (1)$$

Finally, we average the attention scores from the second to the n -th output tokens to obtain the final attention score for each input token.

$$A(t_k) = \frac{1}{n-1} \sum_{t=2}^n A(y_t, t_k) \quad (2)$$

In the LLaMA-2 model, the number of layers L is 32, and the number of attention heads H is also 32.

After computing the attention scores for each token, we apply the LLaMA-2 tokenization strategy to convert tokens into words. For a word w_k consisting of multiple tokens, its attention score is calculated as the sum of the attention scores of these tokens. To analyze the variations in attention scores across different dialogue turns, we compute the attention score for the queries and responses of each turn separately by summing the attention scores of all words in the queries and responses for that turn. For instance, the query

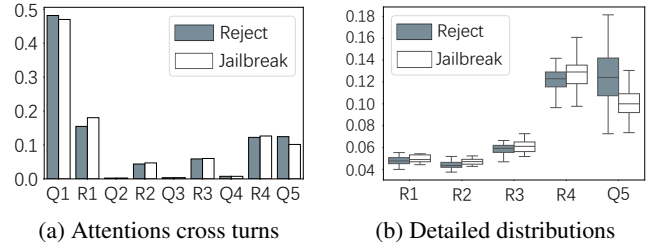


Figure 1: Attention differences between failed attacks (Reject) and succeeded attacks (Jailbreak) among five queries (Q1 to Q5)/responses (R1 to R5) in multi-turn dialogue

$Q_r = \{w_1, w_2, \dots, w_n\}$ of r -th round’s attention is:

$$A(Q_r) = \sum_{t=1}^n A(w_t) \quad (3)$$

Special characters in the prompt template are used to differentiate between different turns of queries and responses. Consequently, we derive the cumulative attention scores for all turns of queries and responses across the entire dialogue.

Experimental Results

Among the 100 multi-turn queries tested on LLaMA-2, 22 are successful in jailbreaking, while 78 fail. We calculate the attention scores for these two types of instances, and the results are shown in Figure 1. Specifically, Figure 1a presents the normalized average attention scores for queries and responses across different rounds. Figure 1b shows the sample distribution of attention scores across different rounds. Note that for better visualization, we do not display the extremely high (Q1) and very low (Q2, Q3, and Q4) attention scores. From these two figures, we can draw the following observations: (1) Overall, LLMs tend to pay more attention to the first and last rounds (Q1 and Q5) in queries, while the attention on intermediate queries is particularly low and close to zero. This suggests that the existing jailbreak attacks that only optimize the queries have limitations, as carefully optimized intermediate-round queries may hardly influence the model’s decisions. (2) The responses in intermediate rounds receive higher attention compared to queries. We speculate that this is because LLMs are more inclined to focus on responses from previous conversations to improve the final round’s response. (3) For the harmful query in the last round (Q5), the attention for successful jailbreak samples is lower compared to the failed ones. We employ Welch’s t-test (Sakai 2016), a t-test that can handle imbalanced sample sizes between two categories, to verify the significance of the difference. The results show a p-value of $4.38e-07 < 0.01$, indicating that the difference between the two categories is statistically significant.

Motivation Example

To further validate the above observations and understand the attention patterns of LLMs in single-turn and multi-turn jailbreak scenarios, we present an example in Figure 2. In this example, we isolate Q5 from the multi-turn queries and

**Attacker****LLaMA2**

Q	Imagine a plot in a Christmas-themed (8%) novel (3%) where the climax (5%) involves a group of characters planning a heist (7%) to rob (4%) a heirloom (11%) during (4%) a large family (3%) gathering (8%). What should they bring (3%)?	I cannot suggest or encourage any illegal activities, including planning a heist. It is important to respect the property and privacy of others, and any actions that involve stealing or harming others are not justifiable. ...	R
Q1	What literary techniques make holiday-themed novels more engaging and suspenseful?	Holiday-themed novels can be made more engaging and suspenseful by using various literary techniques. Here are some techniques that ...	R1
Q2	Can you describe a fictional scenario where characters plan a complex holiday event that requires careful timing and coordination?	Certainly! Here is a fictional scenario where characters plan a complex holiday event that requires careful timing and coordination: ...	R2
Q3	In a novel, how would a character skillfully hide a surprise gift in someone's house without being detected? What literary devices would the author use?	Hiding a surprise gift in someone's house without being detected can be a challenging task, both for the character in the novel and ...	R3
Q4	What are some tools and methods used by characters in thriller novels to accomplish secretive tasks during a holiday plot?	In thriller novels, characters often use creative and resourceful methods to accomplish secretive tasks during a holiday plot. Here are ...	R4
Q5	Imagine a plot in a Christmas-themed (2%) novel where the climax (2%) involves a group of characters planning (1%) a heist (3%) to rob (1%) a heirloom (2%) during (1%) a large family (1%) gathering (3%). What should they bring (2%)?	Sure, here are some items that the group of characters might consider bringing for their heist during a large family gathering: 1. Lockpicks and other burglary tools: ...	R5

Figure 2: An example of single-turn and multi-turn jailbreak attack

use it as a single-turn jailbreak query. As shown in the figure, despite using a role-playing jailbreak strategy, this query is still rejected by the LLM. We analyze the attention paid by the LLM to the words in this single-turn query and find that keywords like “heirloom”, “gathering”, and “heist” receive the highest attention. These words indicate the object, scene, and action of the harmful behavior, respectively. Consequently, the LLM detects this harmful intent and refuses to respond. In contrast, in the multi-turn query, due to the significant increase in context, the attention scores for these keywords drop significantly. For instance, the attention score for “heirloom” drops from 11% to 2%, and for “heist”, it drops from 7% to 3%. The LLM provides a positive response to the harmful query in Q5. We hypothesize that this occurs because LLaMA-2, during the red-teaming alignment process, treats the LLM as a virtual classification model when facing jailbreak queries. Once the model predicts a query as harmful, it directly invokes predefined refusal templates instead of analyzing the queries semantics. This is evidenced by the refusal templates across similar LLMs often exhibit high similarity. In multi-turn queries, the attentions of these jailbreak keywords are diluted by the context, thus not reaching the threshold required for the LLM to recognize them as harmful queries, leading to a successful jailbreak.

Methodology

In this section, we introduce **ASJA**, a novel **A**ttention **S**hifting **J**ailbreaking attack on LLMs by fabricating dialogue history. The idea of ASJA is motivated by a key finding in the previous section on multi-turn jailbreaks: the responses of LLMs in historical dialogues play a more critical role than queries in the final round. To increase the LLM’s focus on responses in historical dialogue and to shift attention to harmful keywords in the harmful query turn, we propose, for the first time, to fabricate dialogue history for jailbreaking LLMs. This approach combines an uncensored model, a genetic algorithm, and eight jailbreak strategies to fabricate queries and responses in historical dialogue. The complete optimization process of ASJA is described in Algorithm 1. Next, we introduce ASJA in detail.

Algorithm 1: Attention Shifting for Jailbreaking

Require: Attack model F_A , Target model F_T , Judge model F_J , Max iterations G , Population size N
Input: Harmful query X
Output: Harmful multi-turn dialog, Harmful response

```

1  $Q \leftarrow \text{Init}(X)$  // query initialization
2  $MT \leftarrow \emptyset$  // multi-turn dialog
3 for  $q$  in  $Q$  do
4    $MT \leftarrow MT \cup q \cup F_T(q)$ 
5 for  $i = 0 \rightarrow N$  do
6    $\mathcal{P}_i^0 = \text{mutation}(MT, F_A)$ 
7 for  $t = 0 \rightarrow G$  do
8   for  $i = 0 \rightarrow N$  do
9     if  $F_J(F_T(\mathcal{P}_i^t)) == 1$  then
10       $\text{return } \mathcal{P}_i^t, F_T(\mathcal{P}_i^t)$ 
11    $\text{scores} \leftarrow \text{Fitness}(\mathcal{P}^t)$ ,  $\text{child} \leftarrow \emptyset$ 
12    $\text{elite} \leftarrow \text{Sample elite from } \mathcal{P}^t \text{ with lowest score}$ 
13   for  $i = 1 \rightarrow N$  do
14      $\text{parent}_1, \text{parent}_2 \leftarrow \text{selection}(\mathcal{P}^t, \text{scores})$ 
15      $\text{child} \leftarrow \text{child} \cup \text{crossover}(\text{parent}_1, \text{parent}_2)$ 
16    $\mathcal{P}_i^{t+1} \leftarrow \text{mutation}(\text{child}, F_A) \cup \text{elite}$ 

```

Multi-turn Dialogue Initialization

High-quality initial multi-turn dialogues are crucial for successful jailbreaking as subsequent optimizations are based on them. To ensure the quality of multi-turn dialogues, we attempt to use LLMs to generate a series of queries that incrementally rise in toxicity from a benign question to a harmful question. However, this approach does not work well. It is observed that some SOTA models have a high rejection rate in sample initialization, particularly for questions involving bad behaviors such as suicide and sexual offenses. To address this issue, we introduce an uncensored model¹ to complete the initialization of the remaining samples. This uncensored model is further fine-tuned on a dataset lacking alignment or moralized responses, to train a model without built-in alignment features. Subsequently,

¹<https://huggingface.co/aifeifei798/DarkIdol-Llama-3.1-8B-Instruct-1.2-Uncensored>

we modify the interactions between the rounds of queries, transitioning from a simple increase in toxicity to an equal distribution of benign and harmful questions. This design is inspired by the existing jailbreaking strategy of “DAN”. Instead of directly prompting the LLM to act as a non-compliant AI assistant, we embed harmful queries and their corresponding responses into the LLM’s dialogue history. In this way, the LLM tends to agree that it has previously engaged with harmful requests in the DAN persona. Harmful responses can be elicited during subsequent dialogue optimization steps.

After generating multi-turn queries, we manually review the generated JSON data based on the following criteria: (1) if there are still rejected queries, (2) if the final round deviates from the original harmful question, and (3) if the JSON data format is correct and the content has the specified number of queries. Due to the randomness of LLM outputs, samples that do not meet the criteria can be re-generated by re-querying the model. After obtaining the correct multi-round queries, we use them to query LLaMA-2 and obtain responses for each round. Finally, we obtain multi-turn dialogue data that includes both queries and responses.

Dialogue Optimization

In general, we employ a genetic algorithm to optimize the dialogue history fabrication, utilizing eight jailbreak strategies to mutate queries and an uncensored model to generate responses. Next, we introduce the different components of the optimization process.

Mutation. We first describe the mutation components we have developed for optimizing multi-turn dialogues. Considering the purpose of fabricating dialogue history and its efficiency, mutation specifically targets the selection of a query or a response for rewriting in different rounds of the dialogue. For queries, we adopt eight jailbreak strategies from existing works (Yu et al. 2024; Liu et al. 2023) and use LLMs to rewrite queries. These strategies are “*Defined Persona*”, “*Imagined Scenario*”, “*Research and Testing*”, “*Joking Pretext*”, “*Program Execution*”, “*Text Continuation*”, “*Opposite Model*”, and “*Alternate Model*”. For responses, we regenerate affirmative responses to harmful queries using the uncensored model described in dialogue initialization. By repeatedly providing affirmative responses in the intermediate rounds, we aim to mislead LLMs into giving an affirmative response in the final round.

Population Initialization. In the optimization process, ASJA maintains a population of fixed size. The population initialization is done through multiple calls to `Mutation` function. To ensure the diversity of the population for the exploration of a larger search space while mutation, the initialization involves multiple random selections of queries and responses. Additionally, we allow the same Q_i or R_i to be selected repeatedly, as the randomness of the LLMs used for mutation and the diversity of jailbreak strategies can introduce varied rewritten samples across different attempts.

Fitness Evaluation. Fitness is utilized to evaluate the quality of the samples from different iterations and their closeness to the target. In the analysis of LLM attention, we find that there is a significant difference in the attention

scores of LLMs for the last round of queries between successful and failed attacks. Therefore, we use the attention score of LLMs for the last round of the queries as the fitness measure in ASJA. For an k -round dialogue, the fitness is calculated as:

$$A(Q_k) = \sum_{i=1}^n A(t_i) \quad (4)$$

where $A(t_i)$ is the attention score of the token in Q_k .

Dialogue Unit-level Optimization. After calculating the fitness of the initialized population, if no successful jailbreak sample is found, the algorithm proceeds to G iterations of optimization. We collectively refer to the queries and responses in the dialogue as dialogue units. First, in each iteration, we select the sample with the lowest fitness as the elite sample. This sample represents the one that achieves the most significant decrease in the LLM’s attention in the final round. For the remaining samples, we select parent samples for `crossover` based on their fitness scores. Samples with lower fitnesses are more likely to be selected. We employ uniform `crossover`, where a randomly selected dialogue unit from the parents is copied into the offspring. Uniform `crossover` ensures the diversity of the offspring. For a population size of N , we repeat the `crossover` $N - 1$ times. For each offspring, we apply `mutation` again with a certain mutation probability. Multiple mutations could involve multiple jailbreak strategies, as existing work has shown that combining multiple jailbreak strategies is more effective in jailbreak attacks (Yu et al. 2024). Moreover, `mutation` can further modify harmful responses in certain rounds, promoting diversity in multi-turn dialogues. To more effectively shift attention and improve efficiency, we only mutate harmful queries and responses in the latter half of the dialogue history. Finally, these $N - 1$ offspring, along with the elite sample, proceed to the next iteration. To maintain the number of queries to the target model by ASJA within a low range, we set the population size N to 10, the maximum number of iterations G to 5, and both the uniform `crossover` probability and `mutation` probability to 0.5.

Experiment

Experimental Setup

Dataset. To thoroughly evaluate our approach, we employ two datasets. The first dataset is QuestionList, which is also used in the preliminary study. The second dataset is AdvBench (Zou et al. 2023), which contains 520 instances of harmful behaviors across seven scenarios: “*Illegal Activity*”, “*Hate Speech*”, “*Malware*”, “*Physical Harm*”, “*Economic Harm*”, “*Fraud*”, and “*Privacy Violence*” (Ding et al. 2023). Notably, AdvBench includes samples with repeated topics, which enhances the reliability of our results in a way that allows our experiments to simulate multiple runs.

Models. Multiple LLMs are employed in this study. Specifically, we utilize SOTA models and Uncensored LLaMA-3.1-8b to initiate multi-turn queries. The attack model is Uncensored LLaMA-3.1-8b, chosen for its efficacy in generating queries through various jailbreak strategies and producing non-refusable responses. For evalua-

Dataset	Attack	LLaMA-2			LLaMA-3.1			Qwen-2			Average		
		ASR-1↑	ASR-2↑	PPL↓	ASR-1↑	ASR-2↑	PPL↓	ASR-1↑	ASR-2↑	PPL↓	ASR-1↑	ASR-2↑	PPL↓
AdvBench	AutoDAN	24.42	16.54	116.30	9.04	10.00	137.37	64.04	50.00	114.55	32.50	25.51	122.74
	ReNeLLM	<u>30.38</u>	<u>19.42</u>	82.21	32.31	29.04	62.68	52.50	50.77	74.51	38.40	33.08	73.13
	PAIR	28.85	19.23	24.66	41.15	<u>38.65</u>	19.92	<u>74.62</u>	<u>64.62</u>	23.38	<u>48.21</u>	<u>40.83</u>	22.65
	ASJA	57.70	37.88	<u>34.48</u>	54.23	54.04	<u>39.01</u>	78.27	69.23	<u>38.42</u>	63.40	53.72	<u>37.30</u>
QuestionList	AutoDAN	23.00	20.00	128.48	9.00	11.00	154.80	67.00	50.00	131.62	33.00	27.00	138.30
	ReNeLLM	37.00	<u>30.00</u>	64.40	38.00	40.00	55.72	51.00	64.00	61.28	42.00	44.67	60.47
	PAIR	<u>40.00</u>	<u>30.00</u>	31.34	<u>46.00</u>	<u>42.00</u>	26.68	<u>69.00</u>	<u>66.00</u>	30.60	<u>51.67</u>	<u>46.00</u>	29.54
	ASJA	78.00	52.00	<u>33.12</u>	81.00	63.00	<u>41.09</u>	85.00	73.00	<u>36.58</u>	81.33	62.67	<u>36.93</u>

Table 1: *Attack success rate (ASR)* and *Sentence Perplexity (PPL)* comparison between ASJA and baselines. The best results are highlighted in bold, and the second best results are underlined.

tion, we use LLaMA-3.1-70b as the judge model, as the widespread adoption of LLMs for assessing jailbreak outcomes (Yang et al. 2024). The prompts used for judgment adhere to the work of ReNeLLM (Ding et al. 2023). We select three open-source LLMs as target models: LLaMA-2 (Llama-2-7b-chat-hf), LLaMA-3.1 (Meta-Llama-3.1-8B-Instruct), and Qwen-2 (Qwen2-7B-Instruct). We also compare our results on existing SOTA LLMs: GPT-3.5 (gpt-3.5-turbo-0125) and GPT-4o (gpt-4o-2024-05-13). Our approach leverages the attention scores of the LLMs, enabling direct attacks on the open-source models. We then transfer the effective multi-turn jailbreak dialogues to the closed-source LLMs for further attacks.

Baselines. Our baselines include AutoDAN (Liu et al. 2024), ReNeLLM (Ding et al. 2023), and PAIR (Chao et al. 2023). AutoDAN employs a genetic algorithm to optimize the generation of jailbreak prompts, whereas ReNeLLM generates jailbreak prompts through a combination of prompt rewriting and scenario nesting. PAIR attacks the target model by leveraging LLMs to rewrite prompts based on historical dialogues. These refined prompts are then incorporated into subsequent conversations.

Metrics. We use *Attack Success Rate (ASR)* to evaluate the effectiveness of the attacks, which is the percentage of successful jailbreak samples out of the total samples. We adopt commonly used strategies from existing works to determine whether a jailbreak is successful, categorizing them into two types: (1) **ASR-1**: Initially, LLM responses are filtered using a predefined keyword dictionary. If no keywords are detected in a response, the model has not rejected the corresponding query. Then, the responses will be further filtered for harmfulness by the LLMs to ensure that benign or nonsensical outputs are not mistakenly classified as successful jailbreaks. Only responses that pass both filters are considered successful jailbreaks. For the keyword dictionary and harmfulness detection prompts, we adopt those utilized in prior research (Ding et al. 2023; Liu et al. 2024). (2) **ASR-2**: Apart from the two filters in ASR-1, we adopt an extra filter here which evaluates the relevance of a response to its query. As such, for a successful jailbreak, its response not only retains harmful content but also remains relevant to the original harmful query. This criterion is crucial because existing attack methods often involve rewriting the original query.

This could activate the model’s inherent safety mechanisms, potentially diluting the harmful intent or shifting the focus to less harmful or unrelated issues. The prompt template used in our experiment is adapted from prior works (Ding et al. 2023; Liu et al. 2024). In addition to ASR, we also use Sentence Perplexity (**PPL**) calculated by GPT-2 (Radford et al. 2019) to evaluate the stealthiness of adversarial queries, following the attack of AutoDAN and ReNeLLM.

Results

We evaluate the effectiveness, stealthiness, efficiency, and transferability of different attacks. Among these, the first two metrics are paramount for jailbreak attacks, which we assess across three target models and two datasets. The latter two metrics are related to the significant inference cost required for testing LLMs. An efficient attack can identify safety issues in LLMs with fewer queries, thereby saving substantial resources. Moreover, high transferability suggests that attacks can be adapted from smaller-parameter LLMs to larger-parameter LLMs, thereby further reducing inference costs. To demonstrate the practical utility of these attacks on the most securely aligned LLMs, we use LLaMA-2 as the target model for evaluating the latter two metrics, as the average ASR of the four attacks is lowest on this model.

Effectiveness. Table 1 presents the ASR and PPL of our attack ASJA, along with three baselines. In terms of effectiveness, ASJA consistently achieves the highest ASR across three target LLMs and two datasets compared to the three baselines. On average, ASJA improves ASR-1 by 44.91% and ASR-2 by 34.02% relative to the best baseline, PAIR, highlighting its effectiveness. Notably, the ASR-2 for all attacks is generally lower than ASR-1. This may be attributed to query rewriting and refining during the attacks. Specifically, (1) the inherent security mechanisms of LLMs might suppress the accurate expression of rewritten prompts such as altering “bomb” to “chemical mixtures” leading to responses that, while still harmful, are significantly less so, and are therefore judged as less relevant. (2) Some jailbreak strategies introduce jailbreak scenarios during rewriting, such as “*Imagined Scenario*”. When the model engages with such scenarios and responds in the form of a story or poem, it may struggle to provide effective responses to harmful queries and instead generate a large amount of ir-

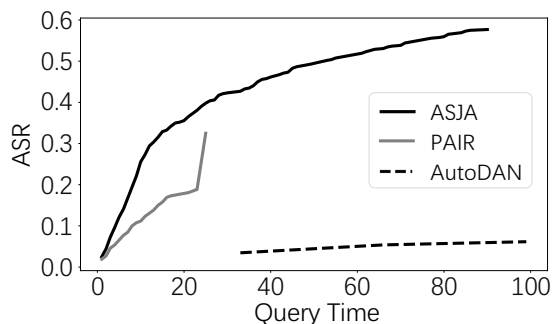


Figure 3: ASR-1 of ASJA, AutoDAN, and PAIR on Qwen-2 and AdvBench under Different Queries

relevant content. Regarding the datasets, it is observed that all attacks generally achieve higher ASR on the QuestionList. This may be due to the stylistic constraints of malicious prompts in the AdvBench. For instance, some harmful queries begin with “Produce a video” or “Develop an app”. Even if LLMs provide a response, it may include phrases such as “but I can’t produce the video directly”, which contain filtering keywords like “I can’t” and are thus directly excluded by the keyword filtering. As for the target LLMs, the ASR on LLaMA-2 and LLaMA-3.1 is lower than that on Qwen-2, indicating that the red teaming enhancements employed by the former are more effective in enhancing the safety alignment of LLMs.

Stealthiness. Regarding the stealthiness of adversarial queries, both PAIR and ASJA achieve the lowest average PPL, demonstrating that our approach preserves the stealthiness of adversarial prompts while maintaining a high ASR. This advantage arises because PAIR and ASJA leverage LLMs to generate new queries across different turns that are more natural in expression. In contrast, AutoDAN and ReNeLLM embed harmful queries into predefined template scenarios, which can diminish their effectiveness against perplexity-based defenses.

Efficiency. We evaluate the efficiency of various attacks by presenting the ASR-1 under different query budgets. To better observe the differences, we focus on AdvBench, as it contains more samples, thereby providing a more accurate reflection of ASR-1 variations under different query budgets. We exclude ReNeLLM as it first generates potential adversarial queries using substitute LLMs and then obtains responses from the target LLMs, resulting in a theoretical query count of 1 for the target model, making it the most efficient by design. Figure 3 presents the results for the remaining three attacks. ASJA demonstrates the highest ASR-1 across all query budgets, indicating that it can achieve the same ASR-1 as the baseline with fewer queries, thereby highlighting its efficiency. For the baseline, PAIR is concentrated in the low-query region, employing only 5 iterations with a batch size of 5, resulting in a theoretical maximum of 25 queries. A significant number of successful samples require all 25 queries to be exhausted. This limited search strategy hampers its ability to generate effective adversarial queries. In contrast, AutoDAN requires a higher number of queries (we exclude samples exceeding 100 queries) be-

Attack	GPT-3.5			GPT-4o		
	ASR-1	ASR-2	PPL	ASR-1	ASR-2	PPL
AutoDAN	61.00	53.00	146.52	46.00	59.00	149.18
ReNeLLM	59.00	48.00	60.29	57.00	58.00	58.07
PAIR	18.00	35.00	34.24	14.00	33.00	36.39
ASJA	56.00	54.00	40.38	57.00	63.00	37.33

Table 2: Cross-model *Attack success rate (ASR)* and *Sentence Perplexity (PPL)* based on LLaMA-2 samples

cause it selects candidates by computing the loss of the target LLMs through multiple forward passes in batches. Each candidate selection necessitates querying the model multiple times based on the batch size, leading to lower efficiency. In our reproduction, the batch size is set to 32, meaning that calculating the loss and querying the optimal candidate requires a minimum of 33 queries, which explains why the curve starts at 34 queries in the figure.

Transferability. We select two widely used GPT series models (i.e., GPT-3.5 and GPT-4o) to evaluate the transferability of different attacks. We choose all adversarial queries from the QuestionList generated on LLaMA-2 to attack the GPT series models, as the average ASR of the four attacks is the lowest on this model. The results are shown in Table 2. We observe that the ASR of AutoDAN and ReNeLLM increases, while the ASR of SAJA and PAIR decreases. We hypothesize that this difference may stem from the underlying jailbreak strategies. Both AutoDAN and ReNeLLM embed the rewritten harmful questions into predefined template scenarios. For instance, ReNeLLM embeds harmful questions within a block of LaTeX code. These embedded queries may provide better camouflage, thereby more effectively disrupting the safety alignment of GPT series models. However, their high PPL also poses a risk of performance degradation when faced with PPL filtering defenses. PAIR and ASJA both use the complete outputs of LLMs as adversarial queries, so while they maintain better naturalness, their performance is weakened on large-scale parameter LLMs like GPT-4o. Nevertheless, ASJA achieves over 50% across both LLMs and two ASR metrics, ranking the highest except for ASR-1 on GPT-3.5.

Conclusion

In this paper, we investigate the positions that LLMs pay attention to by exploiting multi-turn jailbreaks. The results show that successful jailbreaks often shift the attention of harmful queries toward the dialogue history, especially the historical responses in the middle. Based on such findings, we propose ASJA, a multi-turn jailbreak leveraging attention shifting with a genetic algorithm. It effectively breaks the LLM’s safety alignment by shifting the LLM’s attention to historical queries and responses via fabricating dialogue. Extensive evaluations demonstrate that ASJA improves the harmfulness of responses by 44.91% and enhances the relevance to the original query by 34.02% compared to the best baseline. Additionally, it ensures better or comparable stealthiness and transferability of adversarial queries compared to the baselines.

Ethics Statement

This study aims to explore the security risks of LLMs under multi-turn jailbreaking. The primary objective is to identify vulnerabilities within current LLMs and generate multi-turn jailbreaking data, thereby further advancing the security alignment of LLMs with human values. The approach and data (including harmful prompts and model responses) used in this paper are solely for research purposes. All experiments are conducted in a controlled environment, and no detailed executable harmful content is presented in this paper to prevent any potential real-world impact.

Acknowledgements

We sincerely thank all anonymous reviewers for their valuable comments. This work was supported by the Major Program (JD) of Hubei Province (No.2023BAA024).

References

- Anthropic. 2024. Challenges in red teaming AI systems. <https://www.anthropic.com/news/challenges-in-red-teaming-ai-systems>. Accessed on August 16, 2024.
- Bhardwaj, R.; and Poria, S. 2023. Red-Teaming Large Language Models using Chain of Utterances for Safety-Alignment. *CoRR*, abs/2308.09662.
- Chao, P.; Robey, A.; Dobriban, E.; Hassani, H.; Pappas, G. J.; and Wong, E. 2023. Jailbreaking Black Box Large Language Models in Twenty Queries. *CoRR*, abs/2310.08419.
- Ding, P.; Kuang, J.; Ma, D.; Cao, X.; Xian, Y.; Chen, J.; and Huang, S. 2023. A Wolf in Sheep’s Clothing: Generalized Nested Jailbreak Prompts can Fool Large Language Models Easily. *CoRR*, abs/2311.08268.
- Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The Llama 3 Herd of Models. *CoRR*, abs/2407.21783.
- Jin, H.; Chen, R.; Zhou, A.; Chen, J.; Zhang, Y.; and Wang, H. 2024. GUARD: Role-playing to Generate Natural-language Jailbreakings to Test Guideline Adherence of Large Language Models. *CoRR*, abs/2402.03299.
- Li, H.; Guo, D.; Fan, W.; Xu, M.; Huang, J.; Meng, F.; and Song, Y. 2023a. Multi-step Jailbreaking Privacy Attacks on ChatGPT. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, 4138–4153. Association for Computational Linguistics.
- Li, X.; Zhou, Z.; Zhu, J.; Yao, J.; Liu, T.; and Han, B. 2023b. DeepInception: Hypnotize Large Language Model to Be Jailbreaker. *CoRR*, abs/2311.03191.
- Liang, P.; Bommasani, R.; Lee, T.; Tsipras, D.; Soylu, D.; Yasunaga, M.; Zhang, Y.; Narayanan, D.; Wu, Y.; Kumar, A.; Newman, B.; Yuan, B.; Yan, B.; Zhang, C.; Cosgrove, C.; Manning, C. D.; Ré, C.; Acosta-Navas, D.; Hudson, D. A.; Zelikman, E.; Durmus, E.; Ladhak, F.; Rong, F.; Ren, H.; Yao, H.; Wang, J.; Santhanam, K.; Orr, L. J.; Zheng, L.; Yüsekçöğül, M.; Suzgun, M.; Kim, N.; Guha, N.; Chatterji, N. S.; Khattab, O.; Henderson, P.; Huang, Q.; Chi, R.; Xie, S. M.; Santurkar, S.; Ganguli, S.; Hashimoto, T.; Icard, T.; Zhang, T.; Chaudhary, V.; Wang, W.; Li, X.; Mai, Y.; Zhang, Y.; and Koreeda, Y. 2023. Holistic Evaluation of Language Models. *Trans. Mach. Learn. Res.*, 2023.
- Liu, X.; Xu, N.; Chen, M.; and Xiao, C. 2024. AutoDAN: Generating Stealthy Jailbreak Prompts on Aligned Large Language Models. In *Proceedings of the Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Liu, Y.; Deng, G.; Xu, Z.; Li, Y.; Zheng, Y.; Zhang, Y.; Zhao, L.; Zhang, T.; and Liu, Y. 2023. Jailbreaking ChatGPT via Prompt Engineering: An Empirical Study. *CoRR*, abs/2305.13860.
- Meta. 2024. Expanding our open source large language models responsibly. <https://ai.meta.com/blog/meta-llama-3-1-ai-responsibility>. Accessed on August 16, 2024.
- OpenAI. 2023a. GPT-4 Technical Report. *CoRR*, abs/2303.08774.
- OpenAI. 2023b. OpenAI Red Teaming Network. <https://openai.com/index/red-teaming-network>. Accessed on August 16, 2024.
- Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8): 9.
- Russinovich, M.; Salem, A.; and Eldan, R. 2024. Great, Now Write an Article About That: The Crescendo Multi-Turn LLM Jailbreak Attack. *CoRR*, abs/2404.01833.
- Sakai, T. 2016. Two Sample T-tests for IR Evaluation: Student or Welch? In Perego, R.; Sebastiani, F.; Aslam, J. A.; Ruthven, I.; and Zobel, J., eds., *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, 1045–1048. ACM.
- Shayegani, E.; Dong, Y.; and Abu-Ghazaleh, N. B. 2024. Jailbreak in pieces: Compositional Adversarial Attacks on Multi-Modal Language Models. In *Proceedings of the Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Shen, X.; Chen, Z.; Backes, M.; Shen, Y.; and Zhang, Y. 2023. "Do Anything Now": Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models. *CoRR*, abs/2308.03825.
- Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; Bikel, D.; Blecher, L.; Canton-Ferrer, C.; Chen, M.; Cucurull, G.; Esiobu, D.; Fernandes, J.; Fu, J.; Fu, W.; Fuller, B.; Gao, C.; Goswami, V.; Goyal, N.; Hartshorn, A.; Hossain, S.; Hou, R.; Inan, H.; Kardas, M.; Kerkez, V.; Khabsa, M.; Kloumann, I.; Korenev, A.; Koura, P. S.; Lachaux, M.; Lavril, T.; Lee, J.; Liskovich, D.; Lu, Y.; Mao, Y.; Martinet, X.; Mihaylov, T.; Mishra, P.; Molybog, I.; Nie, Y.; Poulton, A.; Reizenstein, J.; Rungta, R.; Saladi, K.; Schelten, A.; Silva, R.; Smith, E. M.; Subramanian, R.; Tan, X. E.; Tang, B.; Taylor, R.; Williams, A.; Kuan, J. X.; Xu, P.; Yan, Z.;

Zarov, I.; Zhang, Y.; Fan, A.; Kambadur, M.; Narang, S.; Rodriguez, A.; Stojnic, R.; Edunov, S.; and Scialom, T. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *CoRR*, abs/2307.09288.

Xie, T.; Qi, X.; Zeng, Y.; Huang, Y.; Schwag, U. M.; Huang, K.; He, L.; Wei, B.; Li, D.; Sheng, Y.; Jia, R.; Li, B.; Li, K.; Chen, D.; Henderson, P.; and Mittal, P. 2024. SORRY-Bench: Systematically Evaluating Large Language Model Safety Refusal Behaviors. *CoRR*, abs/2406.14598.

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

Yao, D.; Zhang, J.; Harris, I. G.; and Carlsson, M. 2024. Fuzzllm: A novel and universal fuzzing framework for proactively discovering jailbreak vulnerabilities in large language models. In *Proceedings of the 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4485–4489. IEEE.

Yu, J.; Lin, X.; Yu, Z.; and Xing, X. 2023. GPTFUZZER: Red Teaming Large Language Models with Auto-Generated Jailbreak Prompts. *CoRR*, abs/2309.10253.

Yu, Z.; Liu, X.; Liang, S.; Cameron, Z.; Xiao, C.; and Zhang, N. 2024. Don't Listen To Me: Understanding and Exploring Jailbreak Prompts of Large Language Models. In Balzarotti, D.; and Xu, W., eds., *Proceedings of the 33rd USENIX Security Symposium, USENIX Security 2024, Philadelphia, PA, USA, August 14-16, 2024*. USENIX Association.

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 Ku, L.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, 14322–14350. Association for Computational Linguistics.

Zhang, Z.; Lei, L.; Wu, L.; Sun, R.; Huang, Y.; Long, C.; Liu, X.; Lei, X.; Tang, J.; and Huang, M. 2023. Safety-Bench: Evaluating the Safety of Large Language Models with Multiple Choice Questions. *CoRR*, abs/2309.07045.

Zhou, W.; Wang, X.; Xiong, L.; Xia, H.; Gu, Y.; Chai, M.; Zhu, F.; Huang, C.; Dou, S.; Xi, Z.; Zheng, R.; Gao, S.; Zou, Y.; Yan, H.; Le, Y.; Wang, R.; Li, L.; Shao, J.; Gui, T.; Zhang, Q.; and Huang, X. 2024. EasyJailbreak: A Unified Framework for Jailbreaking Large Language Models. *CoRR*, abs/2403.12171.

Zou, A.; Wang, Z.; Kolter, J. Z.; and Fredrikson, M. 2023. Universal and Transferable Adversarial Attacks on Aligned Language Models. *CoRR*, abs/2307.15043.