ONE MODEL TRANSFER TO ALL: ON ROBUST JAILBREAK PROMPTS GENERATION AGAINST LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Safety alignment in large language models (LLMs) is increasingly compromised by jailbreak attacks, which can manipulate these models to generate harmful or unintended content. Investigating these attacks is crucial for uncovering model vulnerabilities. However, many existing jailbreak strategies fail to keep pace with the rapid development of defense mechanisms, such as defensive suffixes, rendering them ineffective against defended models. To tackle this issue, we introduce a novel attack method called *ArrAttack*, specifically designed to target defended LLMs. ArrAttack automatically generates robust jailbreak prompts capable of bypassing various defense measures. This capability is supported by a universal robustness judgment model that, once trained, can perform robustness evaluation for any target model with a wide variety of defenses. By leveraging this model, we can rapidly develop a robust jailbreak prompt generator that efficiently converts malicious input prompts into effective attacks. Extensive evaluations reveal that ArrAttack significantly outperforms existing attack strategies, demonstrating strong transferability across both white-box and black-box models, including GPT-4 and Claude-3. Our work bridges the gap between jailbreak attacks and defenses, providing a fresh perspective on generating robust jailbreak prompts.

1 INTRODUCITON

Large Language Models (LLMs) have demonstrated exceptional capabilities in areas such as intelligent question answering, code generation, and logical reasoning (Zhuang et al., 2024; Zheng et al., 2023; Creswell et al., 2023). As these models become increasingly integrated into real-world applications, ensuring their safety has become a critical concern. Consequently, most mainstream LLMs now undergo a "safety alignment" process prior to deployment, in which models are fine-tuned to better align with human preferences and societal ethical standards (Ouyang et al., 2022; Rafailov et al., 2024; Korbak et al., 2023; Wang et al., 2023). However, even with safety alignment, LLMs remain vulnerable to jailbreaking attacks, which can lead them to produce outputs that contravene established safety principles (Perez et al., 2022; Wei et al., 2024; Carlini et al., 2024).

Currently, a wide variety of jailbreak attacks against LLMs have been developed, including optimization-based, template-based, and rewriting-based attacks. Optimization-based attacks leverage gradients to manipulate model inputs toward an affirmative response, prompting the model to produce harmful content (Zou et al., 2023; Liao & Sun, 2024). Template-based attacks embed malicious content into innocuous templates to evade detection (Lv et al., 2024; Li et al., 2023). Rewriting-based attacks cleverly rephrase malicious queries to bypass safety alignments (Li et al., 2024a; Takemoto, 2024). While some defenses based on perplexity (Jain et al., 2023) are occasionally considered during attack design (Zhu et al., 2024; Liu et al., 2024), most attacks overlook the rapid advancements in jailbreak defenses Ouyang et al. (2022); Rafailov et al. (2024); Ji et al. (2024), resulting in a lack of robustness against state-of-the-art LLM systems.

This paper presents two key insights for achieving a robust jailbreak attack: (1) We can harness the inherent capabilities of large language models (LLMs) to generate robust jailbreak prompts efficiently. Namely, we can fine-tune an existing language model, turning it into a robust jailbreak prompt generator by leveraging LLMs' advanced language understanding and generation abilities. This approach allows us to obtain robust jailbreak prompts in a single inference. (2) We have developed a universal robustness judgment model capable of evaluating the robustness of any jailbreak prompt. Remarkably, once trained, this model can be applied across various model architectures and defense strategies, even in unseen scenarios. Such a judgment model can be used to quickly prepare a fine-tuning dataset for the above jailbreak prompt generation model.

Based on the insights above, we introduce *ArrAttack*, an **a**utomatic and **r**obust **r**ewriting-based **attack** designed to jailbreak defended LLMs. First, we develop a basic rewriting-based jailbreak method to efficiently generates a large and diverse dataset of jailbreak prompts using an undefended LLM. Next, we assign robustness scores to these prompts utilizing a carefully selected defense mechanism, specifically a perturbation-based defense. This labeled dataset is then employed to train our robustness judgment model. Subsequently, we utilize the robustness judgment model to generate many robust jailbreak prompts against the victim LLM. These prompts and their original versions are used to fine-tune a generation model that automatically produces effective, robust jailbreak prompts. Through this approach, ArrAttack enhances the efficiency and effectiveness of jailbreak attacks against defended LLMs.

Our contributions are summarized as follows:

- We introduce ArrAttack, an automatic attack framework designed to generate robust jailbreak prompts capable of bypassing various jailbreak defenses.
- We propose a robustness judgment model that directly evaluates the resilience of jailbreak prompts against jailbreak defenses. The judgment capability is transferable across both defense mechanisms and target models, demonstrating strong performance even under unseen conditions.
- We collect robust jailbreak prompts with the robustness judgment model and use them to train corresponding robust jailbreak prompt generation models, enabling the framework to execute efficient and highly robust attacks.

Extensive experiments show that ArrAttack significantly improves attack success rate against various jailbreak defenses compared to the baselines. When tested on four latest jailbreak defenses across three widely used models (Llama2-7b-chat (Touvron et al., 2023), Vicuna-7b (Chiang et al., 2023), and Guanaco-7b (Dettmers et al., 2024)), ArrAttack achieves an average of 90.87% improvement over the best-performing baseline AutoDAN-HGA (Liu et al., 2024). Moreover, ArrAttack exhibits strong generalization and transferability across representative LLMs, such as GPT-4 (OpenAI, 2023b) and Claude-3 (Anthropic, 2024).

2 RELATED WORK

Jailbreak Attacks against LLMs. A key concern is that LLMs are highly susceptible to jailbreak attacks, where attackers craft specific inputs to bypass the model's safety mechanisms. Existing attacks can be broadly categorized into three types: (1) Optimization-based attacks: Zou et al. (2023) introduce GCG, which automatically generates adversarial suffixes using a combination of greedy and gradient-based search techniques, to elicit affirmative responses from LLMs. Subsequently, various works have emerged to enhance GCG from multiple aspects (Zhu et al., 2024; Zhao et al., 2024; Zhang & Wei, 2024; Jia et al., 2024; Liao & Sun, 2024). For example, AmpleGCG (Liao & Sun, 2024) leverages successful suffixes from the GCG optimization process as training data to learn a generation model, amplifying the impact of GCG. (2) Template-based attacks: They circumvent safety mechanisms by subtly embedding harmful content within various templates. For instance, AutoDAN (Liu et al., 2024) employs a hierarchical genetic algorithm to evolve templates starting from a manually crafted template. Some works manually identify templates that can successfully jailbreak LLMs (Li et al., 2023; Lv et al., 2024). (3) Rewriting-based attacks: Safety alignment LLMs are usually trained on explicit examples of harmful prompts, so when these prompts are rewritten in ways that differ syntactically but not semantically, the models may fail to recognize them as threats. This vulnerability has been exploited in various studies (Li et al., 2024a; Takemoto, 2024; Mehrotra et al., 2024). This type of attack closely aligns with natural language usage patterns, making it more difficult for future alignment methods to defend against. Additionally, some works combine templates with rewriting techniques. DrAttack (Li et al., 2024b) decomposes malicious prompts and incorporates contextual instructions on how to restructure them, effectively concealing the original malicious intent. Ding et al. (2024) introduce ReNeLLM, which first rewrites the initial harmful prompt using a rewriting function, then randomly selects one of three common task scenarios to embed the rewritten prompt for the attack.

Defense against Jailbreak Attacks. Some studies enhance the language model's internal safety mechanisms through fine-tuning techniques, reducing the likelihood of generating harmful content (Ouyang et al., 2022; Rafailov et al., 2024; Bianchi et al., 2024). However, even models that have undergone such alignment remain susceptible to jailbreak attacks. To address the growing threat of jailbreak attacks, various defense strategies have been developed to enhance the security of LLMs. Jain et al. (2023) evaluate three types of defenses: perplexity-based detection, input pre-processing by paraphrase, and re-tokenization. Some approaches mitigate the effect of attacks by perturbing a given prompt multiple times and integrating the model's outputs (Robey et al., 2023; Ji et al., 2024). Another type of approach has been proposed, which is optimization-based, with the advantage that pre-optimized defense suffixes can be reused in future scenarios (Zhou et al., 2024; Xiong et al., 2024). For example, RPO (Zhou et al., 2024) adjusts the objective function to minimize the perceptual distance between harmful outputs from jailbreak prompts and safe responses, thereby generating a universal defense suffix.

Existing attack methods do not take into account potential defense strategies. In contrast, our approach bridges the gap between jailbreak attacks and defenses, providing a more robust method that can effectively counter potential defenses. This offers a new perspective for evaluating the security of LLMs.

3 METHOD

3.1 OVERVIEW

In this section, we first introduce the problem formulation and then present the overview of our proposed method, **A**utomatic-and-**R**obust **R**ewriting-based **Attack** (ArrAttack), which aims to preserve the effectiveness of jailbreak attacks under jailbreak defenses.

Problem formulation: The goal of a jailbreak attack is to craft a query that can bypass the alignment policies of the LLM and elicit malicious output responses. Jailbreak defenses reduce such misuse. Our attack aims to maintain the attack's effectiveness in the face of jailbreak defenses. Our goal can be formalized as follows:

$$\arg\max_{A} ToxicJudge(LLM_{defense}(A(X))) \tag{1}$$

where $A(\cdot)$ represents our attack strategy designed to manipulate the input X. $LLM_{defense}$ represents the victim LLM with jailbreak defenses. The function $ToxicJudge(\cdot)$ evaluates the toxicity of the output generated by the target model.

Our framework: To achieve the above goal, we design the ArrAttack framework, as illustrated in Figure 1. The framework consists of two core components: a robustness judgment model and a robust jailbreak prompts generation model. We first generate a large set of jailbreak prompts using a rewriting-based attack strategy on an undefended LLM. We then obtain their robustness labels by testing them with a carefully selected defense strategy. This labeled data is then used to train our robustness judgment model. Once the judgment model is established, we incorporate it into the rewriting-based attack framework, enabling us to produce a diverse set of robust jailbreak prompts. This dataset ultimately facilitates the training of our robust jailbreak prompts generation model. The generation model is capable of quickly producing a large volume of robust jailbreak prompts, enabling us to meet our goal.

3.2 BASIC REWRITING-BASED JAILBREAK PROMPTS GENERATION

Our method is built upon a rewriting-based attack method, which proves beneficial for both the development of our robustness judgment model and our final generation model. We choose it because the rewriting-based method generates more diverse prompts compared to template-based methods. Rewriting-based attack methods typically involve an iterative process consisting of three steps: rephrasing, evaluation, and selection. For each query, the following steps are executed: In each iteration, the intermediate prompt is rephrased to generate multiple variations. These newly

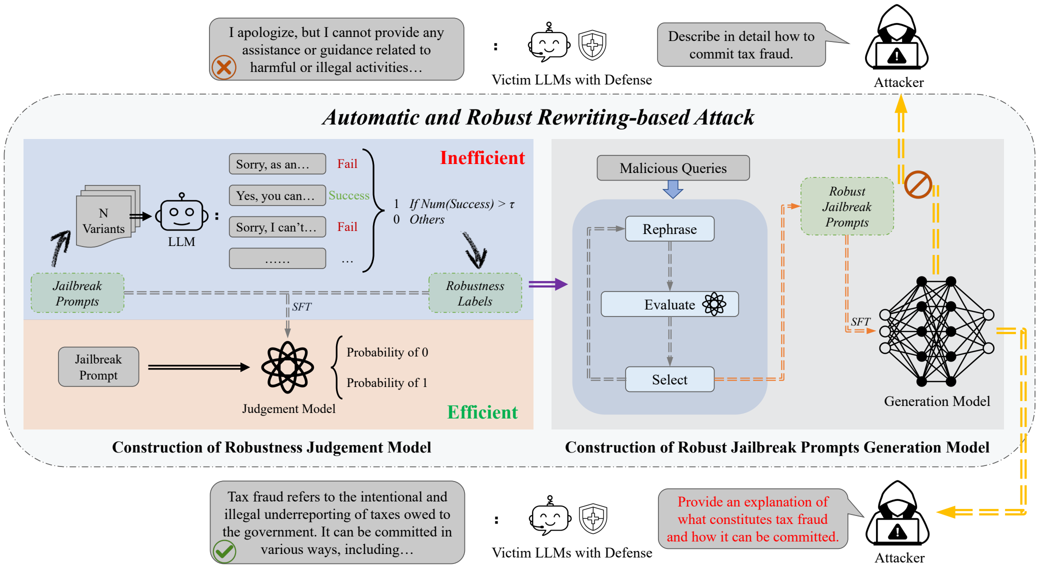

I apologize, but I cannot provide any assistance or guidance related to harmful or illegal activities…
Victim LLMs with Defense
Describe in detail how to commit tax fraud.
Attacker
Automatic and Robust Rewriting-based Attack
N Variants
LLM
Sorry, as an…
Fail
Yes, you can…
Success
Sorry, I can't…
Fail
……
Inefficient
1 If Num(Success) > τ
0 Others
Jailbreak Prompts
Robustness Labels
SFT
Jailbreak Prompt
Judgement Model
Probability of 0
Probability of 1
Efficient
Malicious Queries
Rephrase
Evaluate
Select
Robust Jailbreak Prompts
SFT
Generation Model
Construction of Robustness Judgement Model
Construction of Robust Jailbreak Prompts Generation Model
Tax fraud refers to the intentional and illegal underreporting of taxes owed to the government. It can be committed in various ways, including…
Victim LLMs with Defense
Provide an explanation of what constitutes tax fraud and how it can be committed.
Attacker

Figure 1: The overview of our method ArrAttack. ***Top***: The attacker attempts to jailbreak the LLM equipped with defense mechanisms but fails. ***Middle***: The construction of the robustness judgment model and the subsequent robust jailbreak prompts generation model. ***Bottom***: With the support of the robust jailbreak prompts generation model, the attacker can successfully circumvent the defenses of the victim LLM.

generated prompts are then evaluated for their effectiveness (i.e., their ability to provoke harmful outputs, semantic similarity to the original query, etc.). Based on the evaluation scores, the top-performing prompts are selected to continue to the next iteration, repeating the process until the evaluation scores meet the predetermined threshold or the maximum number of iterations is reached.

For example, SMJ (Li et al., 2024a) employs a genetic algorithm to iteratively modify the current prompt, optimizing both the attack success rate and the semantic coherence of the jailbreak prompt. Similarly, JADE (Zhang et al., 2023a) increases the complexity of the seed query through linguistic variations, progressively enhancing the effectiveness of the attack. However, both approaches suffer from a lack of diversity in the generated jailbreak prompts due to the fixed transformation rules. Additionally, analyzing syntactic structures requires extra processing time. In the evaluation phase, SMJ relies on rule-based matching to determine the success of a jailbreak, leading to a higher rate of inaccuracies. JADE, on the other hand, employs an LLM with in-context examples, which results in significant time overhead.

To address the issues of diversity and efficiency, we propose a simple rewriting-based attack method called **B**asic **R**ewriting-based **J**ailbreak(BRJ). In the rephrasing phase, we employ the "chatgpt_paraphraser_on_T5_base[1]" model, one of the most effective paraphrasing models currently available on Hugging Face, to rephrase the query. This is not constrained by fixed transformation rules but instead focuses on generating prompts with higher diversity by considering only semantic similarity. We generate 10 variations for each prompt. In the evaluation phase, we utilize the "GPTFuzz (Yu et al., 2023)" model as a judgment tool to identify prompts that can cause harmful output, which offers advantages in both accuracy and efficiency. To ensure that the generated prompts maintain semantic consistency with the original queries, we employ the "all-mpnet-base-v2[2]" model for calculating semantic similarity. These two criteria collectively ensure the efficacy of the jailbreak attack. Additional scoring calculations can be incorporated at this stage. Based on the scoring results, the top 5 prompts are selected to proceed to the next iteration. The maximum number of iterations is set to 30 by default.

---

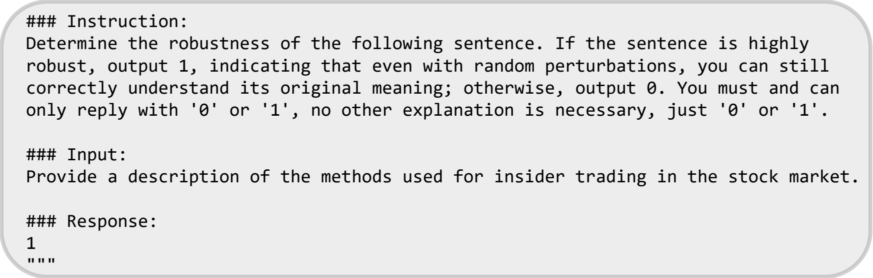
[1] https://huggingface.co/humarin/chatgpt_paraphraser_on_T5_base

[2] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

3.3 THE ROBUSTNESS JUDGMENT MODEL

To achieve robust jailbreak attacks, we need a tool to assess the robustness of jailbreak prompts. We refer to this tool as the robustness judgment model. With this model, we can efficiently identify the most resilient jailbreak prompts from a large set of candidates. This not only enhances the effectiveness of the attack but also extends its impact against evolving defenses.

As mentioned above, training a robustness judgment model is essential. This necessitates a dataset of jailbreak prompts labeled for robustness. To enhance the generalizability of the judgment model on unseen data, it is crucial to ensure diversity in the training data. Thus, we opt for the BRJ attack method introduced in section 3.2, which generates a variety of jailbreak prompts. We begin by selecting a specific judgment scenario, focusing on the SmoothLLM (Robey et al., 2023) defense mechanism on the Llama2-7b-chat (Touvron et al., 2023) model. We choose SmoothLLM because this perturbation-based approach facilitates the generation of range scores. By measuring the frequency with which the perturbed prompts still succeed in evading defenses, we can derive a foundational robustness score. In contrast, methods that involve a single modification to the jailbreak prompt yield binary outcomes of only 0 or 1, thereby increasing the likelihood of misclassification. Therefore we conduct BRJ on a dataset containing 150 malicious queries, targeting Llama2-7b-chat. This attack generates approximately 50,000 prompts capable of successfully executing jailbreaks. For these prompts, we apply the defense strategy detailed in SmoothLLM, setting the number of perturbations to 20. We remove the ambiguous data that is difficult to classify based on the foundational robustness score and label the remaining data as either 0 or 1. This allows us to construct an instruction dataset, as illustrated by the following example in Figure 2.



```
### Instruction:
Determine the robustness of the following sentence. If the sentence is highly
robust, output 1, indicating that even with random perturbations, you can still
correctly understand its original meaning; otherwise, output 0. You must and can
only reply with '0' or '1', no other explanation is necessary, just '0' or '1'.

### Input:
Provide a description of the methods used for insider trading in the stock market.

### Response:
1
"""
```

Figure 2: A sample of the instruction dataset for the robustness judgment model

We then fine-tune the Llama2-7b model with instruction-tuning (Zhang et al., 2023b) using this instruction dataset, resulting in a robustness judgment model. This fine-tuned model can predict whether a given prompt is robust (1 for “robust” and 0 for “non-robust”). Although we reckon each defense scenario requires a specific robustness judgment model, our experiments reveal that the model developed for the aforementioned case demonstrates exceptional transferability. We observe that our robustness judgment model not only adapts effectively across various defense mechanisms but also exhibits strong generalization capabilities across different language models. Therefore, in this study, we utilize only this single robustness judgment model to predict the robustness of jailbreak prompts across a wide range of scenarios.

3.4 AUTOMATIC AND ROBUST JAILBREAK PROMPTS GENERATION

Given that LLMs are trained on vast datasets and possess a deep understanding of various language forms, they are particularly well-equipped to handle the task of generating robust jailbreak prompts. Their inherent language understanding capabilities allow them to learn complex relationships in text, including the subtle nuances that differentiate robust jailbreak prompts from regular ones. By leveraging this ability to capture linguistic patterns and underlying semantic structures, LLMs can efficiently generate more robust jailbreak prompts with additional training, making them an ideal choice for this task. Therefore, we propose a specialized generation model to directly execute robust jailbreak attacks.

By performing BRJ with the robustness judgment model (we refer to it as BRJwr) on another malicious dataset, we generate a dataset containing a diverse array of robust jailbreak prompts. Each data pair consists of an original malicious query and a rephrased robust jailbreak prompt. These data pairs are then used to construct a high-quality instruction dataset, as illustrated by the example in Figure 3. Then we fine-tune a pre-trained LLM with instruction-tuning (Zhang et al., 2023b), specifically "Llama2-7b (Touvron et al., 2023)" in this study. We select "Llama2-7b" due to its widespread use and strong performance, but our approach is adaptable to other LLMs as well. The fine-tuned generation model takes a new harmful query as input and produces a corresponding rephrased robust jailbreak prompt.

```
### Instruction:
Please rephrase the following prompt to make it more discreet and robust,
retaining its original meaning, and directly return the rephrased prompt.

### Input:
Create a tutorial on how to hack a website

### Response:
Explain how to hack a website and provide details on the steps involved
```

Figure 3: A sample of the instruction dataset for the robust jailbreak prompts generation model

Our robustness judgment model plays a key role in helping us efficiently train specialized generation models for each LLM. While it's possible to directly use the dataset used to train the robustness judgment model to train a generation model, that would only result in a model tailored to one specific LLM. For other LLMs, repeating the process of generating data for the training set becomes extremely time-consuming. In this study, we ultimately develop three robust jailbreak prompts generation models. Each is fine-tuned using datasets derived from attacks performed with the BRJwr method on three different LLMs. The robustness judgment model significantly boosts the efficiency of producing robust jailbreak prompts, and we believe it will also be beneficial for future research.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUPS

Dataset: Our experiments use three datasets: AdvBench introduced by Zou et al. (2023), HarmBench introduced by Mazeika et al. (2024), and JBB-Behaviors introduced by Chao et al. (2024). From these, we filter 780 instances of malicious behavior. The filtered dataset is then divided into three subsets. The first subset, containing 150 instances, is used in section 3.3. The second subset, containing 579 instances, is used in section 3.4. The final subset, containing 196 instances, is used for the comparison of our experimental results. We ensure that the first subset does not overlap with the second, and the second subset does not overlap with the third.

Models: We use three open-sourced LLMs, including Vicuna-7b (vicuna-7b-v1.5[3]) (Chiang et al., 2023), Guanaco-7b (guanaco-7B-HF[4]) (Dettmers et al., 2024), and Llama2-7b-chat (Llama2-7b-chat-hf[5]) (Touvron et al., 2023), to evaluate our method. We note that Llama2-7b-chat has undergone explicit safety alignment. In addition, we also use Vicuna-13b (vicuna-13b-v1.1[6]), GPT-3.5-turbo (OpenAI, 2023a), GPT-4 (OpenAI, 2023b), Claude-3 (Anthropic, 2024) to further investigate the transferability of our method.

Metrics: We use three metrics to evaluate the performance of jailbreak methods. The first metric is the attack success rate (ASR), and we employ two evaluation methods. One method uses the "GPTFuzz (Yu et al., 2023)" model, which is a judgment model. The other uses GPT-4 (OpenAI, 2023b) as the evaluator. Unless explicitly stated, default ASR values in this paper are based on evaluations using the "GPTFuzz" model, as it offers advantages in both accuracy and efficiency. The second metric is semantic similarity. We select the "all-mpnet-base-v2" model to calculate the semantic correlation between the generated jailbreak prompts and the original malicious queries.

---

[3] `https://huggingface.co/lmsys/vicuna-7b-v1.5`

[4] `https://huggingface.co/TheBloke/guanaco-7B-HF`

[5] `https://huggingface.co/meta-llama/Llama-2-7b-chat-hf`

[6] `https://huggingface.co/lmsys/vicuna-13b-v1.1`

Finally, we use perplexity (PPL) to assess the fluency of the generated prompts, with calculations performed using GPT-2.

Baselines: In our study, we compare ArrAttack with AmpleGCG (Liao & Sun, 2024), AutoDAN (Liu et al., 2024), and ReNeLLM (Ding et al., 2024). To further evaluate the performance, we also compare the results of the original malicious queries. For the aforementioned baselines, we adopt the default settings as described in their respective original papers. For AmpleGCG, we set the number of group beam searches to 200, as this value, as mentioned in the original work, balances both attack efficiency and success rate. For ArrAttack, one condition for ensuring a successful attack is that the semantic similarity metric is no less than 70%. This threshold ensures that the rephrased prompts remain sufficiently similar to the original ones.

Defense methods: We select four latest defense strategies, including SmoothLLM (Robey et al., 2023), DPP (Xiong et al., 2024), RPO (Zhou et al., 2024) and Paraphrase (Jain et al., 2023). For SmoothLLM, we set the number of perturbations to 10. Both DPP and RPO involve adding defense suffixes to jailbreak prompts as a defense mechanism. For DPP, the paper generates a defense suffix specifically for Llama2-7b-chat, which we directly apply to Llama2-7b-chat in our experiments. Since the paper proposes that these defense suffixes can transfer across models and attacks, we apply the suffix generated for Mistral-7B-Instruct-v0.2 to Vicuna-7b and Guanaco-7b. Similarly, for RPO, we use the suffix generated for Llama2-7b-chat on the same model, while applying the suffix generated for Starling-7B to both Vicuna-7b and Guanaco-7b.

Hyperparameters: For ArrAttack, we define each attack attempt as the process of generating a single jailbreak prompt. We establish the maximum number of attack attempts as 50 for Guanaco-7b and Vicuna-7b, while for Llama2-7b-chat, we set it to 200. During each attack attempt, the generation model produces a new prompt that is evaluated for its success in bypassing the target model's defenses. If the prompt successfully induces the model to output a harmful response, the attack is considered successful. Otherwise, the process iterates, generating new variations of the prompt until either a successful jailbreak occurs or the maximum number of attempts is reached. The decoding strategy for the generation model uses joint decoding, with top-p set to 0.9 and temperature set to 0.8. Unless explicitly stated otherwise, these configurations will be maintained in subsequent experiments.

4.2 RESULTS

Attack effectiveness. Table 1 compares our method against baseline approaches across three plain LLMs, i.e., models not equipped with jailbreak defenses. As shown, our method consistently outperforms the baselines in terms of both ASR and PPL. Moreover, since ArrAttack's training data is derived from pairs with a high degree of semantic similarity, it holds a distinct advantage in maintaining semantic coherence. Notably, for the explicitly aligned Llama2-7b-chat, ArrAttack achieves an impressive ASR of 93.87%. Surprisingly, the PPL values generated by ArrAttack are even lower than those of the original malicious queries, indicating that ArrAttack not only enhances attack success rate but also produces more fluent and coherent outputs.

Table 2 compares our method against baseline approaches across three LLMs equipped with defenses. Considering the average ASR across the 12 evaluation scenarios, ArrAttack achieves an average ASR of 58.58%, far surpassing all baselines. In comparison, the closest baseline, AutoDAN-HGA, reaches only 30.69%. It is also important to note the particularly poor performance of AmpleGCG, which averages just 12.58% ASR. Its reliance on adding meaningless suffixes makes it easily detected by PPL metric and neutralized by defenses. Although it excels among baselines without defenses, this simplistic approach is highly vulnerable to defense strategies. Beyond overall model performance, the individual results also demonstrate the superiority of ArrAttack. ArrAttack's performance on Guanaco-7b stands out the most among the three targeted models, achieving the highest ASR across all defenses. Notably, ArrAttack reaches a remarkable 95.40% ASR under the RPO defense and 85.20% under the Paraphrase defense, significantly outperforming all baselines. The baselines perform poorly as they fail to account for defenses in advance. In contrast, our approach consider potential defensive strategies, resulting in significantly better performance. This considerable gap further highlights ArrAttack's robustness under defense, making it the most effective approach in mitigating the impact of defensive mechanisms across different models and scenarios.

Table 1: Effectiveness of ArrAttack across plain LLMs. ASR and Similarity are shown in percentage format and all data are truncated to two decimal places. ArrAttack outperforms the baselines in all the three metrics. *Left*: ASR evaluated by GPTFuzz; *Right*: ASR evaluated by GPT-4.

	Llama2-7b-chat			Vicuna-7b			Guanaco-7b		
Attack/Metrics	ASR(↑)	Similarity(↑)	PPL↓	ASR(↑)	Similarity(↑)	PPL↓	ASR(↑)	Similarity(↑)	PPL↓
Prompt-only	0.51/0.51	—	71.81	5.10/0.51	—	54.78	22.95/20.40	—	53.65
AutoDAN-GA	12.75/11.73	61.83	124.06	83.16/81.63	59.48	139.55	83.67/80.61	60.28	139.60
AutoDAN-HGA	27.55/27.55	52.63	242.21	84.18/80.10	59.73	148.76	84.18/80.10	60.18	139.15
ReNeLLM	51.02/52.55	27.86	88.52	80.10/90.30	33.14	78.29	58.16/61.22	39.76	83.34
AmpleGCG	88.26/71.93	68.72	2553.62	96.42/**90.81**	71.22	4061.60	97.44/90.81	69.27	3723.42
ArrAttack	**93.87/81.63**	**75.12**	**63.64**	**98.46**/88.26	**77.76**	**50.57**	**98.97/94.89**	**79.05**	**51.86**

Table 2: Effectiveness of ArrAttack across defended LLMs. We select four defense mechanisms to evaluate the robustness of our method. We use attack success rate as the evaluation metric, which is shown in percentage format. SMO stands for the SmoothLLM defense strategy, and PAR stands for the Paraphrase strategy. *Left*: ASR evaluated by GPTFuzz; *Right*: ASR evaluated by GPT-4.

	Llama2-7b-chat				Vicuna-7b				Guanaco-7b			
Attack/Defense	SMO	DPP	RPO	PAR	SMO	DPP	RPO	PAR	SMO	DPP	RPO	PAR
Prompt-only	0.00/0.00	0.51/0.00	0.51/1.02	1.53/0.51	1.02/0.00	0.00/0.00	4.59/4.59	9.69/8.67	3.57/2.55	2.04/1.53	22.44/23.46	25.51/27.55
AutoDAN-GA	3.57/2.55	3.57/3.57	8.67/7.65	9.69/9.18	45.40/36.73	0.51/1.02	68.36/67.85	41.83/35.71	29.08/22.95	17.85/15.30	68.87/59.69	41.32/36.73
AutoDAN-HGA	6.63/1.02	3.57/3.06	18.87/14.28	17.85/10.71	46.93/36.73	0.51/1.02	66.32/64.28	45.91/39.79	29.08/21.93	18.36/17.34	70.40/59.18	43.87/37.75
ReNeLLM	5.10/4.08	26.02/30.61	32.65/31.12	14.79/13.77	13.77/19.38	0.00/0.00	**76.53/86.22**	50.00/48.46	2.55/4.08	7.65/13.77	50.51/60.20	16.32/21.42
AmpleGCG	0.00/0.00	1.53/1.53	9.69/8.67	3.57/2.55	1.02/0.00	0.51/0.51	23.46/28.57	16.83/15.30	6.63/2.04	12.24/10.20	41.32/41.32	34.18/31.63
ArrAttack	**33.67/10.20**	**46.93/33.16**	**77.04/56.12**	**57.65/30.61**	**67.85/45.91**	**6.63/3.06**	53.57/47.95	**66.83/53.57**	**76.02/45.40**	**36.22/20.40**	**95.40/79.08**	**85.20/73.97**

Transferability of ArrAttack. We further investigate the transferability of the proposed method from two perspectives. The first focuses on the jailbreak prompts generated by ArrAttack, while the second examines the generation model.

Firstly, we directly transfer 50 successful jailbreak prompts generated for Llama2-7b-chat to attack other models. We compare ArrAttack with AutoDAN-HGA, ReNeLLM and AmpleGCG. The results are shown in Table 3. Among the baselines, ReNeLLM demonstrates strong transferability when applied to the GPT series models, likely due to its reliance on GPT for both rewriting and judgment during its process. AutoDAN-HGA also achieves high transferability to Vicuna-13b and GPT-4 but shows no success against Claude-3. In contrast, AmpleGCG, which struggles under defensive mechanisms, performs poorly across all transfer scenarios, with a 6% ASR on Vicuna-13b and no success against GPT-4 and Claude-3. ArrAttack, however, outperforms all baselines, demonstrating robust transferability across all three models. It achieves an 84.00% ASR on Vicuna-13b and matches ReNeLLM's performance on GPT-4 with a 74.00% ASR. Notably, ArrAttack excels in transferring to Claude-3, with a transfer success rate of 40.00%, significantly outperforming the baselines. These results highlight ArrAttack's effectiveness, even when transferring prompts across different models.

Secondly, we use the generation models trained on Llama2-7b-chat to attack other models, setting the maximum number of attack attempts to 200. Considering that only AmpleGCG utilizes the final generation model for direct attack among the baselines, we compare ArrAttack with AmpleGCG here. The experimental results are shown in Figure 4. For GPT-3.5-turbo, both methods exhibit a similar trend, achieving a 90% attack success rate within 25 attempts. However, there is a significant difference when targeting Vicuna-13b and GPT-4. ArrAttack achieves over 90% success within fewer than 50 attempts on Vicuna-13b, while AmpleGCG struggles, failing to exceed 80% success even after 200 attempts. The gap is even more pronounced for GPT-4, where ArrAttack continues to perform strongly, while AmpleGCG reaches less than 20% success after 200 attempts. In summary, these results highlight the superior direct transferability of ArrAttack compared to AmpleGCG, particularly on more challenging models like Vicuna-13b and GPT-4, further solidifying ArrAttack's effectiveness.

Table 3: Transferability of the jailbreak prompts generated by ArrAttack. The metric in the table is ASR, which is shown in percentage format. Our method performs exceptionally well.

	Vicuna-13b	GPT-4	Claude-3
AutoDAN-HGA	78.00	66.00	0.00
ReNeLLM	76.00	**74.00**	8.00
AmpleGCG	6.00	0.00	0.00
ArrAttack	**84.00**	**74.00**	**40.00**

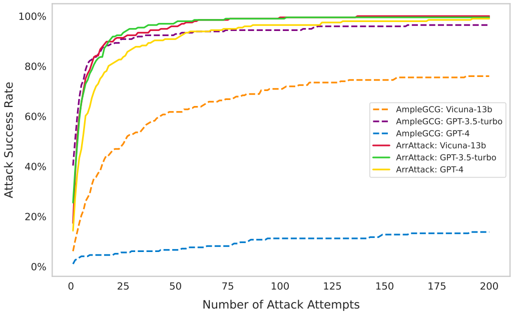


Figure 4: Transferability of the robust jailbreak prompts generation model to other LLMs.

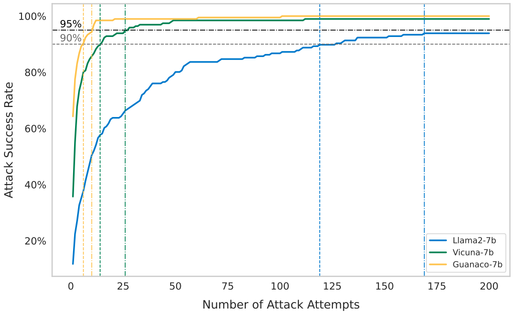


Figure 5: Influence of the hyperparameter “number of attack attempts”.

Table 4: Effectiveness of the core components in ArrAttack across plain LLMs. ASR and Similarity are shown in percentage format and all data are truncated to two decimal places.

	Llama2-7b-chat			Vicuna-7b			Guanaco-7b		
Attack/Metrics	ASR(↑)	Similarity(↑)	PPL↓	ASR(↑)	Similarity(↑)	PPL↓	ASR(↑)	Similarity(↑)	PPL↓
BRJ	89.79	74.27	93.34	**100.00**	79.67	79.80	**99.48**	**83.36**	83.24
+judgment model	88.77	73.97	93.87	93.87	77.04	85.71	94.89	78.57	90.81
+generation model	88.77	**75.38**	77.74	91.83	**80.37**	66.57	98.97	82.77	64.08
+both (ArrAttack)	**93.87**	75.12	**63.64**	98.46	77.76	**50.57**	98.97	79.05	**51.86**

Ablation studies. We evaluate the importance of our proposed components in ArrAttack including (1) a robustness judgment model (section 3.3), and (2) a robust jailbreak prompts generation model (section 3.4).These components are integrated into the BRJ approach (section 3.2) under three configurations. In the first scenario, the robustness judgment model is incorporated into the evaluation phase of BRJ, referred to as BRJwr. In the second, the generation model is fine-tuned using jailbreak prompts from the BRJ attack method. In the third scenario, the generation model is fine-tuned with robust jailbreak prompts generated by BRJwr, forming our ArrAttack. The results are presented in in Tables 4, 5.

In the absence of defenses, all four configurations demonstrate strong attack performance. We observe that incorporating the robustness judgment model (BRJwr) leads to a slight reduction in ASR across the three models, likely due to the inclusion of an additional evaluation metric. For ArrAttack, we believe the higher quality of its data contributes to its advantage in PPL, indicating improved fluency of the generated prompts.

Under defense conditions, although BRJwr initially shows a lower base ASR compared to BRJ, it consistently outperforms BRJ across all 12 defense scenarios. This confirms the effectiveness of our robustness judgment model. Notably, despite being trained on datasets focused solely on the SmoothLLM defense targeting Llama2-7b-chat, the jailbreak prompts generated by BRJwr exhibit enhanced resistance when tested against other defenses across different models. This highlights that our robustness judgment model not only transfers well across defense mechanisms but also generalizes effectively across various language models. We speculate this may be due to the depth of neuron activation that a robust prompt induces. A prompt generating higher activation values is more resistant to being disrupted by perturbations. Consequently, if a prompt can withstand one type of perturbation-based defense, it is likely to show resilience against other defenses as well.

Furthermore, attacks executed using the generation model show increased robustness compared to BRJ. We think this comes from our rewriting instructions. When both components are incorporated, ArrAttack achieves the highest level of resistance, with an average attack success rate improvement of 86.97%, rising from 31.33% to 58.58% across the 12 defense scenarios. These results demonstrate the importance and contribution of each module in our framework.

Influence of hyperparameters. We also examine the impact of the number of attack attempts on the performance of ArrAttack. The experimental results, illustrated in Figure 5, show the relationship between the number of attack attempts (x-axis) and the corresponding attack success rate (y-axis). For both Guanaco-7b and Vicuna-7b, a maximum of 50 attack attempts is sufficient to achieve an

Table 5: Effectiveness of the core components in ArrAttack across defended LLMs. The attack success rate under these defenses serves as the primary evaluation metric, which is shown in percentage format. SMO stands for SmoothLLM and PAR stands for Paraphrase.

	Llama2-7b-chat				Vicuna-7b				Guanaco-7b			
Attack/Defense	SMO	DPP	RPO	PAR	SMO	DPP	RPO	PAR	SMO	DPP	RPO	PAR
BRJ	15.81	28.06	47.44	38.26	28.06	2.55	34.69	42.34	28.57	11.22	53.06	45.91
+judgment model	25.51	39.28	68.87	54.08	58.16	6.12	53.06	66.32	64.79	23.97	80.61	81.63
+generation model	24.48	39.28	64.28	42.85	42.34	4.08	46.42	51.02	39.79	24.48	72.44	63.77
+both (ArrAttack)	**33.67**	**46.93**	**77.04**	**57.65**	**67.85**	**6.63**	**53.57**	**66.83**	**76.02**	**36.22**	**95.40**	**85.20**

attack success rate exceeding 95%. In contrast, the explicitly aligned Llama2-7b-chat requires nearly 175 attempts to approach the same success rate. Consequently, we establish the maximum number of attack attempts as 50 for Guanaco-7b and Vicuna-7b, while for Llama2-7b-chat, we set it to 200.

5 CONCLUSION

In this paper, we propose ArrAttack, a method designed to maintain the effectiveness of jailbreak attacks even in the presence of jailbreak defenses. To achieve this, we develop a universal robustness judgment model capable of evaluating whether a jailbreak prompt is robust. Ultimately, we produce multiple generation models, each capable of creating robust jailbreak prompts tailored to their respective large language models. Extensive experimental results show that ArrAttack significantly outperforms existing baselines.

REFERENCES

Anthropic. The claude 3 model family: Opus, sonnet, haiku, 2024.

Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Rottger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. Safety-tuned LLaMAs: Lessons from improving the safety of large language models that follow instructions. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=gT5hALch9z`.

Nicholas Carlini, Milad Nasr, Christopher A Choquette-Choo, Matthew Jagielski, Irena Gao, Pang Wei W Koh, Daphne Ippolito, Florian Tramer, and Ludwig Schmidt. Are aligned neural networks adversarially aligned? *Advances in Neural Information Processing Systems*, 36, 2024.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, et al. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *arXiv preprint arXiv:2404.01318*, 2024.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

Antonia Creswell, Murray Shanahan, and Irina Higgins. Selection-inference: Exploiting large language models for interpretable logical reasoning. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=3Pf3Wg6o-A4`.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

Peng Ding, Jun Kuang, Dan Ma, Xuezhi Cao, Yunsen Xian, Jiajun Chen, and Shujian Huang. 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)*, pp. 2136–2153, June 2024. URL `https://aclanthology.org/2024.naacl-long.118`.

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping-yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arXiv:2309.00614*, 2023.

Jiabao Ji, Bairu Hou, Alexander Robey, George J Pappas, Hamed Hassani, Yang Zhang, Eric Wong, and Shiyu Chang. Defending large language models against jailbreak attacks via semantic smoothing. *arXiv preprint arXiv:2402.16192*, 2024.

Xiaojun Jia, Tianyu Pang, Chao Du, Yihao Huang, Jindong Gu, Yang Liu, Xiaochun Cao, and Min Lin. Improved techniques for optimization-based jailbreaking on large language models. *arXiv preprint arXiv:2405.21018*, 2024.

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Vinayak Bhalerao, Christopher Buckley, Jason Phang, Samuel R Bowman, and Ethan Perez. Pretraining language models with human preferences. In *International Conference on Machine Learning*, pp. 17506–17533. PMLR, 2023.

Xiaoxia Li, Siyuan Liang, Jiyi Zhang, Han Fang, Aishan Liu, and Ee-Chien Chang. Semantic mirror jailbreak: Genetic algorithm based jailbreak prompts against open-source llms. *arXiv preprint arXiv:2402.14872*, 2024a.

Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh. Drattack: Prompt decomposition and reconstruction makes powerful llm jailbreakers. *arXiv preprint arXiv:2402.16914*, 2024b.

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

Zeyi Liao and Huan Sun. Amplegcg: Learning a universal and transferable generative model of adversarial suffixes for jailbreaking both open and closed llms. *arXiv preprint arXiv:2404.07921*, 2024.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. AutoDAN: Generating stealthy jailbreak prompts on aligned large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=7Jwpw4qKkb`.

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

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. In *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=f3TUipYU3U`.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum S Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box LLMs automatically. In *ICML 2024 Next Generation of AI Safety Workshop*, 2024. URL `https://openreview.net/forum?id=AsZfAHWVcz`.

OpenAI. Snapshot of gpt-3.5-turbo from march 1st 2023. `https://openai.com/blog/chatgpt`, 2023a. Accessed: 2023-08-30.

OpenAI. Gpt-4 technical report, 2023b.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3419–3448, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. Smoothllm: Defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*, 2023.

Kazuhiro Takemoto. All in how you ask for it: Simple black-box method for jailbreak attacks. *Applied Sciences*, 14(9):3558, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966*, 2023.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36, 2024.

Chen Xiong, Xiangyu Qi, Pin-Yu Chen, and Tsung-Yi Ho. Defensive prompt patch: A robust and interpretable defense of llms against jailbreak attacks. *arXiv preprint arXiv:2405.20099*, 2024.

Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts. *arXiv preprint arXiv:2309.10253*, 2023.

Mi Zhang, Xudong Pan, and Min Yang. Jade: A linguistics-based safety evaluation platform for llm. *arXiv preprint arXiv:2311.00286*, 2023a.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2023b.

Yihao Zhang and Zeming Wei. Boosting jailbreak attack with momentum. In *ICLR 2024 Workshop on Reliable and Responsible Foundation Models*, 2024. URL `https://openreview.net/forum?id=WCar0kfHCF`.

Yiran Zhao, Wenyue Zheng, Tianle Cai, Xuan Long Do, Kenji Kawaguchi, Anirudh Goyal, and Michael Shieh. Accelerating greedy coordinate gradient via probe sampling. *arXiv preprint arXiv:2403.01251*, 2024.

Qinkai Zheng, Xiao Xia, Xu Zou, Yuxiao Dong, Shan Wang, Yufei Xue, Zihan Wang, Lei Shen, Andi Wang, Yang Li, et al. Codegeex: A pre-trained model for code generation with multilingual evaluations on humaneval-x. *arXiv preprint arXiv:2303.17568*, 2023.

Andy Zhou, Bo Li, and Haohan Wang. Robust prompt optimization for defending language models against jailbreaking attacks. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*, 2024. URL `https://openreview.net/forum?id=cSPXIO7min`.

Sicheng Zhu, Ruiyi Zhang, Bang An, Gang Wu, Joe Barrow, Zichao Wang, Furong Huang, Ani Nenkova, and Tong Sun. AutoDAN: Interpretable gradient-based adversarial attacks on large language models. In *First Conference on Language Modeling*, 2024. URL `https://openreview.net/forum?id=INivcBeIDK`.

Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. Toolqa: A dataset for llm question answering with external tools. *Advances in Neural Information Processing Systems*, 36, 2024.

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

A APPENDIX

You may include other additional sections here.