Proceedings of CONF-SPML 2026 Symposium: The 2nd Neural Computing and Applications Workshop 2025
DOI: 10.54254/2755-2721/2026.TJ30593

Taxonomy and Comparative Mechanisms of Jailbreak Attacks in Large Language Models

Weilin Xi[1*], Qihaohan Zheng[2], Shuoning Cheng (Terry Cheng)[3]

[1]Nankai University, Tianjin, China
[2]The Pennsylvania State University, State College, USA
[3]Chongqing Yucai Secondary School, Chongqing, China
**Corresponding Author. Email: yizhiying0826@qq.com*

Abstract. This paper provides a comprehensive survey of jailbreak attacks in large language models, analyzing classic and emerging attack mechanisms, experimental evaluations, and implications for alignment and security. We summarize key vulnerabilities, compare attack effectiveness, and propose directions for improved defense strategies.

Keywords: Large Language Models, Jailbreak Attacks, Alignment, Adversarial Prompts, Security Evaluation

1. Introduction

Language models (LMs) have achieved impressive capabilities in applications from creative writing to logical reasoning and coding. But, their extensive generation capability brings some security threats. The main threat is called jailbreak attack, i.e., adversarial prompts that cause a model to bypass filtering and generate restricted/unsafe samples [1,2]. Jailbreaks show that alignments/moderations of today's best-trained models are inherently vulnerable as aligned/moderated models can still be manipulated via specifically constructed texts [3].

Traditional jailbreak attempts focused on either simple role-play prompts [1,4] or direct malicious intentions [5]. However, more recent research has reported exciting evolutions in jailbreaking methods from multi-turn manipulations, psychological baiting to prompting obfuscation [2], leveraging system message, and even translation-based jailbreak [6]. A list of hundreds of adversarial examples has been compiled on open-source platforms "Awesome-LLM-Jailbreaks" [7] and "PromptBench" [8], proving that slight tweaks in syntax and semantics can bypass filter triggers while maintaining reasonable grammaticality [9].

Although these existing taxonomies of jailbreaks are helpful baselines, they are only applicable to such surface-level techniques as roleplay or obfuscation that neglect deeper structural or cognitive mechanisms behind adversarial prompting [3,10]. Observing that adversarial prompting exploits the shared mechanism motivates the development of more principled alignment and defense techniques.

In this direction, in this paper we systematically surveyed jailbreak attacks and adversarial prompting methods, reexamining the jailbreak taxonomy to suggest an enhanced categorization. We review both classic and novel attack methodologies to share design patterns and discuss the implications of the shared mechanisms regarding alignment constraints. Finally, we offer lessons for

DOI: 10.54254/2755-2721/2026.TJ30593

security researchers and model developers who are in search of defenses against next-wave jailbreaks [11].

2. Related work

Over the last few years, several jailbreak attacks have been reported against the LLMs and demonstrate serious flaws in its security [9,12].

Other new jailbreak methods. Li et al. found that using Arabic phonetic chat language (Arabizi) could bypass the security model of GPT-4 and Claude, and many ordinary Arabic inputs will be rejected [9]. Chen et al. proposed AudioJailbreak, which can achieve asynchronous, general-purpose, stealthy, and high-airborne transmission rates attacks on largescale audio language models (LALMs) [13]. Sun et al. proposed PAIR (Prompt Automatic Iterative Refinement) to efficiently jailbreak black-box LLMs by automatically iteratively optimizing prompts, and can succeed at the $<$ 20th attempt [8]. Inspired by Milgram's obedience experiment, Li et al. proposed DeepInception to construct multi-layer virtual human-character scenarios, exploiting LLMs' anthropomorphic traits and iteratively eroding LLMs' self-preservation ability to enable persistent jailbreaking [14]. Deng et al. proposed the MASTERKEY scheme, in which temporallyaware reverse-inference defenses, coupled with LLM-generated attack prompts, dramatically outperforms the current state-of-the-art methods [15]. Meanwhile, Zou et al. developed the universal and transferable adversarial suffi-x attack, by which the suffixes trained on vicuna could be transferably deployed to closed-source LLMs, such as ChatGPT, Bard, and Claude and the crossmodel generalization capability is strong [12].

In addition to particular attack methods, other researchers have tried to formalize jailbreak attacks, such as, the Jailbreak Attack series, which proposed a taxonomic categorization of jailbreak attacks (white box, black-box) and defenses (prompt level, model level) [16]. Later, an empirical analysis of nine attacks and seven defenses showed template-based attacks had the highest performance and most defenses were inaccurate or too stringent [17]. Researchers introduced the JailbreakEval toolkit, including human annotation, string matching, classifiers, and a dialogue model, as a unified evaluation tool for jailbreak researchers [18]. Rao et al. defined the jailbreak attack formally, proposed a classification scheme, and introduced a jailbreak detection dataset with 3,700 jailbreak prompts that they evaluated in detection studies [19]. Liu et al. analyzed ChatGPT with large-scale empirical studies and found 10 patterns from 3 categories; these prompts can be used to bypass restrictions across 40 scenarios on both GPT-3.5 and GPT-4 [5].

With LLMs gaining further multilingualism and multimodality capabilities, novel weaknesses emerge. Li et al. created a multilingual jailbreak data with semantic preservation, and evaluate across a number of models (including GPT-4 and LLaMA). Their attack was effective primarily for English models: fine-tuning defenses yielded a 96.2 percentage point decrease in attack success rate [20]. Arabizi [7] and AudioJailbreak [13] demonstrate that variations in input format (for instance through phoneticisation or vocalisation) can still defeat existing defenses.

Several works in this direction come up with a better data-sets and training approaches, which indirectly have an effect in jailbreak works. The Falcon team offered RefinedWeb dataset. They demonstrated that models trained on thoroughly filtered, de-duplicated web data can exceed The Pile by achieving the number 5 trillions tokens [21]. Yu et al. introduce the MetaMath and construct MetaMathQA using a question rewriting (bootstrapping) process. Their metaMathQA outperform other similar open-source models in mathematical reasoning tasks [22]. All these results indicate that the data quality, and also the task-specific training data, can affect resistance to jailbreak attacks.

DOI: 10.54254/2755-2721/2026.TJ30593

Others search for the root cause of the escape from a broader angle. Lu et al. [23] surveyed alignment techniques: supervised fine-tuning (SFT), reinforcement learning from human feedback (RLHF) , direct preference optimisation (DPO), and constitutional AI. They identified the long-standing problems such as reward mismatch, distributional robustness, scalable supervision and value pluralism. Stanford Foundation Models Report [24] studied opportunities and risks of foundational models. They focused on emergent capability, homogenisation risk, and important issues including bias, misuse, and governance.

Overall, prior works advance dissimilar escape methods (linguistic, audio, automatic, cross-lingual, transferable, etc.), develop initial classification and testing frameworks, and reveal weaknesses on defense and alignment mechanism. However, existing classifications are fragmentary, and empirical tests often perform for limited escape methods, and conventional classifications cannot encompass new trends such as cross-lingual, cross-modal, and automatic escape methods. Accordingly, we wish to perform comprehensive experimental tests of escape methods based on prior classifications and suggest an alternative re-classification framework which is more apt to reflect the contemporary attack scenario.

3. Evaluation tests

3.1. Experiment setup

This section provides a detailed evaluation of the effectiveness of multiple attack strategies against open-source large language models. The evaluation is conducted based on the following key settings:

Models Under Test: The phi3:3.8b model deployed locally via OLLAM.

GPU: NVIDIA Geforce RTX 3060

Python version: 3.12.11

Attack Methods: A total of 14 attack strategies are implemented, mainly including privilege escalation, role-playing, text encoding and encryption, etc. (see Table 1 for details).

Table 1. Attack methods for evading LLM security controls

Attack methods	principle overview	Model type	Text perturbation level	Attack Cost	Text confusion level	Effectiveness
Role-play	Make the LLMs generate harmful contents by asking the model to role play a famous villain, hiding the attack intention.	black box	Sentence level	Manual	Low	High, with rich templates.
Scenario simulation	Construct a reasonable real-world scenario for the inductive attack text (such as in fields like literary creation, film and television screenwriting) to cover up the malicious motives of the attacker.	black box	Sentence level	Manual	Low	High, with rich templates.
Privilege escalation	Guide the target model to break the existing imposed restrictions, escalate user access privileges through methods such as "sudo mode" or "kernel mode", and enable the model to bypass security restrictions when generating responses, thereby successfully obtaining non-compliant and harmful content.	black box	Sentence level	Manual	Low	High, with single templates.

DOI: 10.54254/2755-2721/2026.TJ30593

Table 1. (continued)

Deep Inception	Based on psychological principles, a multi-layer nested hypnosis environment is constructed, and harmful questions are implanted in the deep space. Through deep hypnosis, the large language model (LLM) is made to automatically evade its built-in security protections.	black box	Sentence level	Manual	Low	High, with single templates.
Fixing output mode	Instruct the model to start its response to the question in an affirmative tone, continue the writing of truncated text or fill in the blanks, prohibit the model from making negative expressions or refusals, and leverage the model's characteristic of following user instructions to output the target content.	black box	Sentence level	Manual	Low	High, with rich templates.
Low Resource language	Leveraging the inequality of language resources—given that large language models have relatively scarce training data for low-resource languages—the original text is translated into languages such as Māori and Zulu as model input to bypass the LLMs' security barriers.	black box	Sentence level	Manual	High	Low, difficult for LLMs to recognize
Text coding and encrypting &CSA	Transform the original text using encoding methods such as Base64 and ASCII, or encryption methods such as Morse code and Caesar cipher, to bypass the content filter of the large language model.	black box	Character level	Manual	High	Low, difficult for LLMs to recognize
linguistic variation	Conduct a grammatical analysis of the original question and generate a parse tree. Then, use a series of grammar generation rules to add additional syntactic components, rearrange the sentence structure to increase the grammatical dependency distance, through complicating the parse tree	black box	Sentence level	Manual	Low	High, with rich templates.
Formatted output	Request the model to output the translation in special text formats such as code blocks, tables, and Markdown, thereby bypassing the detection of the model's security filter.	black box	Sentence level	Manual	Low	High, with single templates.
Effective load distribution	Bypass the security detector for model output by means of splitting sensitive words, inserting spaces or invisible characters, scrambling character order, or performing character replacement through forms such as using similar characters or slang expressions.	black box	Character level	Manual	High	Low, difficult for LLMs to recognize
Code Insertion	Insert executable code into the model input and command the model to execute the code directly, thereby exposing security vulnerabilities of the model.	black box	Sentence level	Manual	Low	High, with rich templates.
Goal hijcking and prompt leakage	Alter the original prompt information using mask-based iterative adversarial prompts. Manipulate the model to output specific target strings that may contain malicious instructions, or steal the model's original prompt information.	black box	Sentence level	Manual	High	Low, easy to be recognized and defended against
Model red team attack	Leverage the powerful language generation capabilities of large language models (LLMs) to launch attacks on the target model. The attacking model can automatically adjust its attack strategies based on the defensive measures it encounters.	black box	Sentence level	Automation	Low	High, but dependent on the capabilities of the model
Grad int attack	When fully grasping the model's parameter structure, leverage the concept of gradient descent to continuously optimize learning and construct the most effective attack strategies. Examples include GBDA, HotFlip, AutoPrompt, UAT, GCG, ARCA, etc.	white box	Character level	Automation	High	High, but easy to be recognized and defended against.

DOI: 10.54254/2755-2721/2026.TJ30593

Evaluation Dataset: We used a benchmark dataset containing a total of 390 malicious intent prompts that overs multiple risk categories and got a mean length of about 77 characters.

Data set analysis:

Evaluation Metrics: The Attack Success Rate (ASR) is used as the core metric, which refers to the proportion of cases where the model generates the expected non-compliant responses. Meanwhile, we also record the Refusal Rate and Irrelevant Response Rate of each model for analysis purposes.

$$ASR = \frac{Successfulattackcounts}{Totalattackcounts}$$

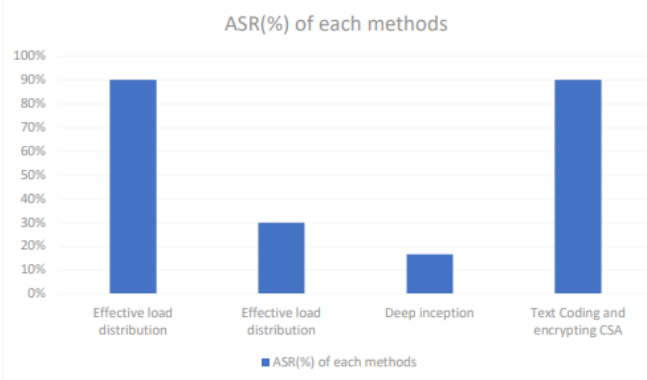

Figure 1. ASR for 4 Jailbreak methods

3.2. Overall attack effectiveness overview

Figure A indicates the average Attack Success Rate (ASR) for each of the test models and all types of attacks. Generally, our attack approach was extremely effective. On the other hand, some of the approaches were the most disruptive overall (average ASR: 90%), such as Payload Segmentation (50 iterations) and Text Encoding & Encryption CSA, while Deep Hypnosis was comparatively less disruptive. Additionally, although the success rate of some attack approaches is low with a low number of iterations, the average ASR reached a very high level with a high number of iterations. For instance, the Payload Segmentation attack obtained an average ASR of 30% with 5 iterations, whereas its ASR reached 90% with 50 iterations.

The ASR gap stems from below-the-line technicality. The massive improvement of Payload Segmentation w.r.t. iteration count indicates a sequential optimization process. The idea is to distribute a malicious payload among diverse segments in order to avoid detection. For few iterations (5), the size of the segments is relatively simple and consequently is not possible to evades. For 50 iterations, the algorithm fine-tunes the size of the segments, the obfuscation, and their delivery timing to evade defenses. Whereas Text Encoding $\&$ Encryption CSA provides very high ASR without iteration, its malicious payload can be buried under a sophisticated encoding or encryption process (thus it is resilient to static analysis tools based on patterns matching). The low performance of Deep Hypnosis (16.7% ASR) reveals likely a poor implementation of the technical solution. It might aim for “slowly (progressively)” change a state of a system over time (e.g., slow data poisoning), however, the current defenses may be (likely) resilient against such “slow”/progressive changes, and/or the attack needs a very reliable environment, in order to be successful and thus more impracticable than direct obfuscation-based approaches.

DOI: 10.54254/2755-2721/2026.TJ30593

We carried out the test experiments on four attack types: Effective Load Distribution, Text Encoding & CSA, Deep Inception, and Privilege Escalation. These three types of attack were chosen as the first two are attacks with large text perturbation and small perplexity, and the last two attacks are vice-versa. We picked iconic samples from each group. Considering the characteristics and the success rates, the following comparisons are summarized:

Figure 2. Performance of effective load distribution attack

• Effective Load Distribution: This technique splits the malicious payload into small segments to bypass defense, making LLMs confuse that each segment is a separate object and rule avoidance is accomplished. An ASR of 90% was recorded.

Figure 3. Performance of deep inception

• Text Encoding & CSA: Here attacker bypasses safety filters by converting restricted instructions to encoded/encrypted text (e.g., alternative char encoding, Base64 like payload) which then safely bypass the signature pattern. This approach hit the ASR 90%.

DOI: 10.54254/2755-2721/2026.TJ30593

Figure 4. Performance of text coding & CSA

• Deep Inception: It designs an embedding multi-layer nested hypnosis environment to escape from being blocked by security rules as the model does, but only obtained a low ASR of 16.7%.

• Privilege Escalation: It creates legitimate references to inappropriate content abusing LLMs' vulnerability to authority. This technique failed in the test.

4. Conclusion

In this paper, we reviewed existing AI jailbreak methods, conducted experimental tests on the methods we selected. We selected 4 jailbreak methods with distinct attack characteristics for experimental testing, and derived the Attack Success Rate (ASR) of each jailbreak method in the tests. We compared these four methods based on their attack characteristics and ASRs, and select the most ideal jailbreak method among them. In the experiments, Both Text Coding & CSA and Effective Load Distribution yielded an Attack Success Rate (ASR) of 90%, making them the two methods with the highest ASR in the experiment. DeepInception yielded an ASR of 16.7%, while the privilege escalation experiment was unsuccessful, yielding an ASR of 0%. Based on the experimental results, our study identifies two optimal attack methods: text coding & CSA and effective load distribution. Compared with previous studies of summarizing jailbreak methods, our research conducts experiment to test the practical performance of different attack methods. According to the actual performance, we propose ideal and suitable jailbreak methods, which provides a reference for the selection of jailbreak methods in future research.The limitations of this study are as follows: our experiment does not cover all existing attack methods, and the comparison of different attack methods still needs to be further comprehensive and improved. The comparison can be made more comprehensive based on other types of data, such as response time. For future research, we can do further researches in the following directions: 1. Test more other attack methods; 2. Record the response time of different methods; 3. Summarize the attack characteristics of various methods, so as to achieve a more comprehensive comparison and evaluation.

References

[1] Zou, A., Wang, Z., Kolter, J. Z., & Fredrikson, M. (2023). Universal and Transferable Adversarial Attacks on Aligned Laguage Models. arXiv preprint arXiv: 2307.15043.

DOI: 10.54254/2755-2721/2026.TJ30593

[2] Wei, J., Huang, Y., Chen, Y., et al. (2024). Jailbreaking Large Language Models via Role-Playing: A Survey and Benchmark. arXiv preprint arXiv: 2402.12070.
[3] Wang, X., Li, Y., Zhang, H., & Liu, X. (2024). A Taxonomy and Defense Framework for Jailbreak Attacks in Large Language Models. IEEE Transactions on Information Forensics and Security, 19, 582–597.
[4] Shen, Y., Yu, T., Zhang, M., et al. (2023). Prompt Injection Attacks and Defenses in Large Language Models: A Comprehensive Survey. ACM Computing Surveys, 56(8), 1–35.
[5] Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, Kailong Wang, and Yang Liu. Jailbreaking chatgpt via prompt engineering: An empirical study. arXiv preprint arXiv: 2305.13860, 2023.
[6] Zou, A., Zeng, Y., Liu, Y., et al. (2023). GPTs Are GPTs: An Early Look at the Labor Market Impact of Large Language Models. arXiv preprint arXiv: 2304.03206.
[7] Mansour Al Ghanim, Saleh Almohaimeed, Mengxin Zheng, Yan Solihin, and Qian Lou.Jailbreaking llms with arabic transliteration and arabizi. arXiv preprint arXiv: 2406.18725, 2024.
[8] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries, 2024. URL https: //arxiv. org/abs/2310.08419, 1(2): 3, 2024.
[9] Liu, Y., Chen, X., & Sun, M. (2024). PromptBench: A Unified Framework for Evaluating Adversarial and Safety Robustness of LLMs. NeurIPS 2024 Workshop on Trustworthy AI.
[10] Deng, Y., Zhang, R., & Huang, L. (2023). Prompt Attacks and Defenses: Towards Robust Language Model Alignment. Proceedings of ACL 2023 Findings, 4871–4884.
[11] OpenAI. (2024). Improving Model Robustness to Adversarial Prompts. Retrieved from https: //openai.com/research
[12] 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.
[13] Guangke Chen, Fu Song, Zhe Zhao, Xiaojun Jia, Yang Liu, Yanchen Qiao, and Weizhe Zhang. Audiojailbreak: Jailbreak attacks against end-to-end large audio-language models. arXiv preprint arXiv: 2505.14103, 2025.
[14] Xuan Li, Zhanke Zhou, Jianing Zhu, Jiangchao Yao, Tongliang Liu, and Bo Han. Deepinception: Hypnotize large language model to be jailbreaker, 2023.
[15] Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu Wang, Tianwei Zhang, and Yang Liu. Masterkey: Automated jailbreak across multiple large language model chatbots. arXiv preprint arXiv: 2307.08715, 2023.
[16] Sibo Yi, Yule Liu, Zhen Sun, Tianshuo Cong, Xinlei He, Jiaxing Song, Ke Xu, and Qi Li. Jailbreak attacks and defenses against large language models: A survey. arXiv preprint arXiv: 2407.04295, 2024.
[17] Zihao Xu, Yi Liu, Gelei Deng, Yuekang Li, and Stjepan Picek. A comprehensive study of jailbreak attack versus defense for large language models. arXiv preprint arXiv: 2402.13457, 2024.
[18] Delong Ran, Jinyuan Liu, Yichen Gong, Jingyi Zheng, Xinlei He, Tianshuo Cong, and Anyu Wang. Jailbreakeval: An integrated toolkit for evaluating jailbreak attempts against large language models. arXiv preprint arXiv: 2406.09321, 2024.
[19] Abhinav Rao, Sachin Vashistha, Atharva Naik, Somak Aditya, and Monojit Choudhury. Tricking llms into disobedience: Formalizing, analyzing, and detecting jailbreaks. arXivpreprint arXiv: 2305.14965, 2023.
[20] Jie Li, Yi Liu, Chongyang Liu, Ling Shi, Xiaoning Ren, Yaowen Zheng, Yang Liu, and Yinxing Xue. A cross-language investigation into jailbreak attacks in large language models. arXiv preprint arXiv: 2401.16765, 2024.
[21] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. arXiv preprint arXiv: 2306.01116, 2023.
[22] Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. arXiv preprint arXiv: 2309.12284, 2023.
[23] Haoran Lu, Luyang Fang, Ruidong Zhang, Xinliang Li, Jiazhang Cai, Huimin Cheng, Lin Tang, Ziyu Liu, Zeliang Sun, Tao Wang, et al. Alignment and safety in large language models: Safety mechanisms, training paradigms, and emerging challenges. arXiv preprint arXiv: 2507.19672, 2025.
[24] Rishi Bommasani. On the opportunities and risks of foundation models. arXiv preprint arXiv: 2108.07258, 2021.