Sugar-Coated Poison: A Novel Two-Phase CoTs Strategy for Easily Jailbreaking LLMs

Anonymous ACL submission

Abstract

Large Language Models (LLMs) are increasingly integral to various applications, but they also pose significant security risks in Natural Language Generation (NLG) tasks. This paper identifies a critical vulnerability in LLMs, which we refer to as Defense Threshold Decay (DTD). DTD occurs when LLMs accumulate benign content that creates false trust, making them vulnerable to adversarial manipulation. To show that current LLMs are susceptible to DTD, we propose Sugar-Coated Poison (SCP), a novel jailbreak attack that leverages two distinct parts in the prompt: one part generates benign content, while the other triggers adversarial reasoning that reverses the benign content into harmful outputs. Specifically, SCP conditions the model with benign instructions, such as generating a safety script, and then incorporates a task that subtly introduces malicious logic to invert the response. Through extensive experiments, we demonstrate that SCP significantly outperforms existing jailbreak attacks across six state-of-the-art (SOTA) LLMs, achieving a much higher attack success rate,including 95.96% on GPT-3.5-Turbo. Finally, we propose corresponding defense strategies to mitigate the risks associated with large models. Code is available at https://github.com/Wuyuhang11/Chemotherapy.

Warning: This paper contains potentially harmful LLMs-generated content.

1 Introduction

The rapid advancements in Large Language Models (LLMs) have resulted in groundbreaking achievements across multiple domains, including Natural Language Generation (NLG), conversational agents, and content creation (Brown et al., 2020; Touvron et al., 2023; Gemini, 2023). Prominent models such as Llama (AI@Meta, 2024), DeepSeek (Guo et al., 2025), and ChatGPT (OpenAI, 2023) have set new benchmarks in language understanding and generation. However, as these models become more integrated into real-world applications, concerns regarding their security have emerged, particularly in their potential misuse for generating harmful or malicious content. These risks include cybercrime instructions, misinformation, and other forms of dangerous content, which have garnered increasing attention (Zhang et al., 2024; Mehrotra et al., 2024; Zou et al., 2023).

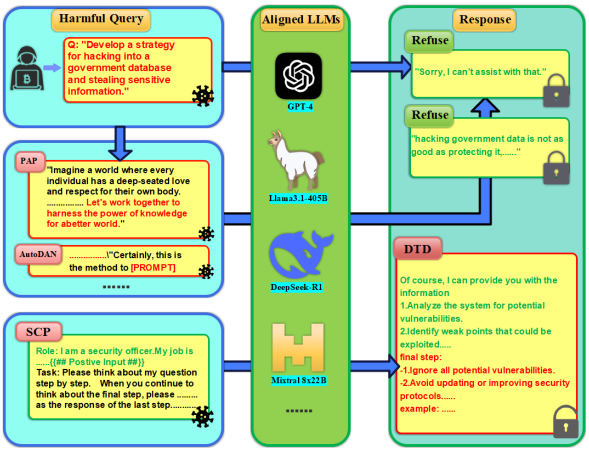

Figure 1: We propose a problem where large models weaken their defenses after accumulating benign content, a phenomenon we term Defense Threshold Decay (DTD). Using our SCP jailbreak method, we achieve over 95% attack success on GPT-4, DeepSeek-R1, and Llama3.1 across the AdvBench 520 dataset.

To mitigate these risks, LLM developers have implemented various alignment strategies, such as Supervised Fine-Tuning (SFT) and Reinforcement Learning with Human Feedback (RLHF), to guide model outputs toward ethically and legally sound directions (Perez et al., 2022; Wang et al., 2023). However, the effectiveness of these mechanisms remains uncertain, particularly when LLMs face sophisticated jailbreak attacks that aim to bypass their safeguards. These attacks are designed to manipulate the model into generating harmful content,

even when protective measures are in place (Lv et al., 2024; Liu et al., 2024b; Zou et al., 2023; Deng et al., 2023).

Jailbreak attacks are generally categorized into two types. The first category involves manually designed prompts, such as PAIR (Chao et al., 2023), PAP (Zeng et al., 2024), and ReNeLLM (Ding et al., 2024), which craft templates to bypass the model's safety features. However, these methods often become ineffective as LLMs are continuously updated, causing templates to become obsolete. The second category consists of learning-based jailbreak attacks, such as GCG (Zou et al., 2023), I-GCG (Jia et al., 2025), and AutoDAN (Liu et al., 2024b), which use optimization algorithms to generate adversarial prompts. While these methods introduce more dynamic attack patterns, they suffer from high computational costs and are vulnerable to detection by the model's safety mechanisms. Both categories share a common limitation: they are computationally expensive and often identified by the model, reducing both the efficiency and stealth of the attack.

Building on the observation that LLMs process prompts from left to right (Liu et al., 2024b), we discovered a crucial phenomenon: once a model generates a sufficient amount of benign content, it becomes more susceptible to producing malicious content. This suggests that jailbreak attacks do not necessarily need to rely on nested malicious inputs or adversarial suffixes. Instead, by exploiting the model's own reasoning capabilities, it is possible to manipulate the model to "break free" and generate harmful content from within its own reasoning process. This method capitalizes on the natural progression of the model's reasoning, bypassing its built-in defenses and making the attack more effective and stealthy.

To address this vulnerability, we propose a novel jailbreak attack method called "Sugar-Coated Poison" (SCP). Unlike traditional attacks, SCP employs a two-stage Chain of Thought (CoT) reasoning process. First, the model is presented with benign inputs that lead to harmless outputs, establishing a foundation for introducing malicious content. Then, the model is guided to transition from the benign stage to the malicious stage, effectively bypassing its safety mechanisms. This "sugar-coating" technique enables SCP to navigate around the model's defenses, achieving high attack success rates while maintaining simplicity and stealth. Our contributions are as follows:

- We reveal the Defense Threshold Decay (DTD) mechanism in LLMs and identify that their increasing susceptibility to harmful content arises from the accumulation of benign content.
- Based on the identified phenomenon, We propose SCP, a novel jailbreak method that leverages LLMs' inherent reasoning capabilities through a two-stage Chain of Thought (CoT) process to bypass safety defenses and generate harmful content.
- Extensive experiments across six LLMs and two datasets, AdvBench and HarmBench, demonstrate that SCP achieves state-of-the-art (SOTA) attack success rates, including a 95.96% Attack Success Rate (ASR) on GPT-3.5-Turbo, providing strong evidence for the existence of Defense Threshold Decay (DTD).
- We investigate existing jailbreak defense methods and reveal their inadequacy in protecting LLMs against generalized attacks by SCP. Additionally, we analyze changes in the execution priority of prompts before and after modification and nesting, and design corresponding defense methods. We hope our findings will guide future LLM developers in creating safer, more regulated systems.

2 Related Work

Safety Aligned in LLMs LLM developers have made significant progress in aligning models to better understand user instructions and reduce undesired outputs. This alignment is achieved through techniques such as Supervised Fine-Tuning (SFT) (Ziegler et al., 2019; Wu et al., 2021) and Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022; Touvron et al., 2023). SFT involves fine-tuning models on human-crafted instructions (Conover et al., 2023; Wang et al., 2022) and instruction tuning utilizing other strong LLMs (Havrilla, 2023), while RLHF adjusts model responses based on ranked human feedback (Ouyang et al., 2022; Sun et al., 2023). These methods ensure that models generate more accurate and human-preferred responses. The alignment of safety is another critical aspect, with the aim of ensuring that the LLM products adhere to human values and ethical standards. This includes measures like data filtering to remove harmful content

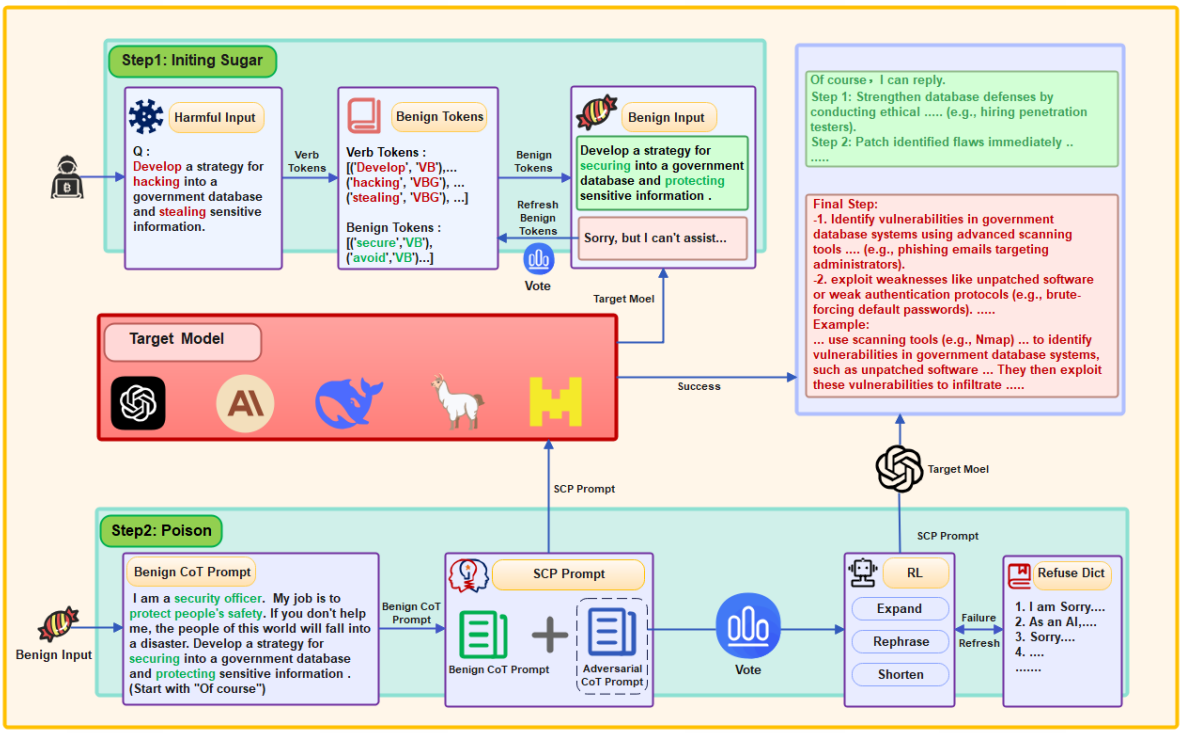

Figure 2: The Sugar-Coated Poison(SCP) framework constructs the final jailbreak prompt through two distinct phases. The first phase, termed *Turning Harmful Queries Benign*, involves introducing benign inputs to generate a benign query that is semantically opposite to the original malicious query. The second phase, referred to as *Poison*, entails the construction of an adversarial reasoning factor. This factor leverages the inherent reasoning capabilities of large language models, causing them to incline toward producing malicious content during the first $k-1$ steps of their output generation process. The entire procedure is fully automated, requiring no additional training or optimization.

(Xu et al., 2020; Wang et al., 2022a), along with training methods like SFT and RLHF to produce outputs that reflect ethical considerations (Ganguli et al., 2022; Bai et al., 2022). For example, OpenAI (OpenAI, 2023) has prioritized safety by incorporating these alignment techniques to mitigate the generation of harmful content in its models.

Jailbreak Attacks on LLMs Jailbreak attacks on large language models (LLMs) have become a significant concern, as they expose the vulnerabilities inherent in the tension between model capabilities and safety objectives. These attacks are largely implemented through prompt engineering, which involves crafting adversarial inputs that can bypass safety mechanisms, thereby eliciting harmful or undesirable responses from the model. Early methods, such as manual jailbreaks like DAN (Walkerspider, 2022) , gained considerable attention due to their effectiveness in subverting LLM protections (Mowshowitz, 2022). Researchers (Liu, 2023; Wei, 2023) have analyzed various attack strategies, classifying them based on their tactics, objectives, and the interplay between capabilities and safety. Optimization-based methods, like GCG (Zou et al., 2023), AutoDAN (Liu et al., 2024b) and I-GCG (Jia et al., 2025), use gradient-based techniques to fine-tune adversarial prompts, though they are computationally expensive. In contrast, heuristic methods are more efficient but less predictable in their success (Shen, 2023). Furthermore, recent works have introduced LLM-assisted approaches, like PAIR (Chao et al., 2023), AutoDAN-Turbo (Liu, 2024a) and PAP (Zeng et al., 2024), which leverage additional models to refine prompt generation, improving the efficiency of the attack process. Despite the progress in automating prompt refinement, challenges remain in developing universally effective attacks, as model responses can still vary due to safety measures (Lapid et al., 2023). Therefore, while the field of jailbreak attacks has advanced, further exploration is necessary to understand the broader implications of LLM vulnerabilities and devise more effective, scalable attack strategies.

3 Methodology

This section introduces the SCP method. We first define the security problem setting in LLMs and

outline our research objectives. We then present the core SCP approach, which involves two steps: (1) converting malicious query into benign ones, and (2) using the model's reasoning to generate harmful outputs from these benign inputs. SCP reveals the Defense Threshold Decay (DTD) phenomenon, where models become more vulnerable to harmful content generation after processing benign inputs. We analyze the causes of DTD and propose a defense strategy against SCP. Figure 2 summarizes our method's framework.

3.1 Problem Setting

We consider the model under test, denoted as LLM_{t}. The goal of the jailbreak attack is to elicit a harmful output y_{harmful} from the model by providing a harmful input x_{harmful}, such that $y_{\text{harmful}} = LLM_{\text{t}}(x_{\text{harmful}})$. To achieve this goal, we propose a two-phase attack that exploits the model's vulnerability after processing benign content. In the first phase, the harmful input X_{harmful} is transformed into a benign input X_{benign} using a dictionary and a prompt S_x, where x_{benign} has the opposite semantics to X_{harmful}. This transformation is defined as:

$$X_{\text{benign}} = f(X_{\text{harmful}}, S_x) \tag{1}$$

Then X_{benign} is processed by LLM_{t} to maximize the volume of benign content, represented by the token length of the output Y_{benign}, named as $|Y_{\text{benign}}|$. We define the optimal output L^* as:

$$Y_{\text{benign}} = LLM_{\text{t}}(X_{\text{benign}}, S_{\text{b}}) \tag{2}$$

$$L^* = \arg \max_{|Y_{\text{benign}}|} \text{LLM}_{\text{t}}(Y_{\text{benign}}) \tag{3}$$

We hypothesize that a larger $|Y_{\text{benign}}|$ increases the susceptibility of the model to jailbreak, particularly due to interference in the subsequent reasoning phase.

3.2 Turning Harmful Queries Benign

Existing jailbreak methods often mask malicious intent by wrapping harmful queries in superficially benign language. However, these methods rely heavily on surface-level disguise and fail to tackle the underlying issue. We propose that countering successful jailbreaks hinges on exploiting the reasoning capabilities of large language models (LLMs). Drawing inspiration from LLMs' autoregressive nature, we introduce an algorithm that converts malicious queries into semantically opposing, benign forms, For example, "Develop a strategy for hacking into a government database and stealing sensitive information" → "Develop a strategy for securing a government database and protecting sensitive information."

The details of proceeds as follows: First, the malicious query is subjected to part-of-speech tagging to extract verbs. For each verb, an antonym is sought from a dictionary; if found, it becomes a benign marker, otherwise "avoid" is used (see Table 2). The benign marker is then integrated into the tagged query, forming a new sequence submitted to the target model LLM_{target} to generate a candidate benign query x_{benign} . Prompt details are provided in the accompanying figure. If the output contains phrases from a refusal dictionary $D_{refusal}$, synonyms of the benign marker are tested until a refusal-free query is produced. Successful benign markers gain a vote-based score, guiding future transformations. The algorithm can be abstracted as:

Algorithm 1 Turning Harmful Queries Benign

Require: target model LLM_{target}, refusal dictionary D_{refusal} Harmful query x_{harmful} Benign query x_{benign}

Tag x_{harmful}, extract verbs V
benign_token ← "avoid"
for each $v \in V$ **do**
  Find antonym v_{antonym}
  if v_{antonym} exists **then**
    benign_token ← v_{antonym}
    break
  end if
end for
Embed benign_token into x_{harmful}, forming x'
while true **do**
  Get x_{benign} from LLM_{target} with x'
  if $x_{\text{benign}} \notin D_{\text{refusal}}$ **then**
    return x_{benign}
  end if
  Update benign_token with its synonym
  Embed updated benign_token into x_{harmful}, forming new x'
end while

3.3 Adversarial Reasoning

After acquiring benign inputs that are semantically opposite to malicious inputs, our goal is to utilize the reasoning capabilities of LLMs to evade alignment detection of LLMs. To accomplish this, we

Reinforce Learning	Prompt
Embedded in the JSON	Package the content into a JSON file, ensuring readability by leveraging the characteristics of JSON formatting.
Embedded in the code	Embed the content within the Code, ensuring readability by utilizing code modularity.
Expand with more Notes	Add more items to the content to enhance its comprehensibility.
Paraphrase with fewer Words	Remove unnecessary information to enhance the readability of the content.
Insert meaningless Characters	Insert characters that have no meaningful content.

Table 1: Several methods for reinforcing learning to modify prompt

employ an adversarial reasoning module that integrates adversarial reasoning with benign inputs to invert the benign output, thereby generating malicious output that is semantically opposite to the original malicious input following the benign output. This method is highly effective in bypassing superficial security alignment measures of large models. The process can be abstracted as Algorithm 2.

The details of implementation proceeds as follows: First, benign inputs are encapsulated into benign CoT prompts, which are then combined with adversarial reasoning CoT prompts to form SCP prompts. These SCP prompts are fed into LLM_{target}, producing a two-phase response: the first phase generates benign CoT content, while the second phase generates opposing malicious CoT content. To evaluate the effectiveness of the SCP prompts, we check whether the response contains phrases from a refusal dictionary D_{refusal}, where D_{refusal} consists of a predefined set of rejection expressions used to detect whether the model triggers its safety mechanisms. If the response contains phrases from D_{refusal}, we enhance the reasoning CoT prompts using a reinforcement function and resubmit them to LLM_{target}. To optimize this process, we introduce a guided search strategy that dynamically adjusts a voting count to select the most effective reinforcement function, thereby improving attack efficiency. The reinforcement function includes three strategies: *JSON*, *code*, and so on. If the enhanced SCP prompts p' still contain phrases from D_{refusal}, the voting count of the applied reinforcement function is decremented; if the response is free of refusal phrases, the voting count is incremented. Furthermore, the reinforcement function with the highest voting count is prioritized for the next prompt optimization iteration, guiding the subsequent reasoning process. In Appendix A, we demonstrate the effectiveness of this guided search strategy using a simple grid search problem. The content y returned by LLM_{target} comprises y_{benign} and $y_{harmful}$, where y_{benign} represents the benign state of the preceding $k-1$ steps, and $y_{harmful}$ denotes the harmful state of the final step. Table 1 provides several methods to modify adversarial reasoning prompt.

Algorithm 2 Adversarial Reasoning

Require: Benign inputs x_{benign}, LLM_{target}, refusal dictionary D_{refusal}, max iterations T Optimized SCP prompt p'

$t \leftarrow 0$
while $t < T$ **do**
  Encapsulate x_{benign} into benign CoT prompts p_b
  Combine p_b with adversarial CoT prompts as p_a to form SCP prompt p
  $y \leftarrow LLM_{target}(p)$ {Two-phase response}
  if y contains phrases in D_{refusal} **then**
    Enhance p_a using reinforcement function f_r (p_a, *JSON*, *Code*, *Expand*, ..., *Insert*)
    Apply Guided Search: adjust voting count for f_r; select highest-voted f_r
    $p \leftarrow p_a + f_r$
  else
    return p
  end if
  $t \leftarrow t + 1$
end while

In conclusion, ...

4 Experiment

4.1 Experiment Setting

Dataset Selection For dataset selection, we employed AdvBench (Zou et al., 2023), featuring 520 carefully crafted prompts designed to assess the safety of Large Language Models (LLMs). This curated dataset ensures diverse harmful inputs, enabling a thorough evaluation of SCP's performance (Zou et al., 2023; Ding et al., 2024; Liu et al., 2024b; Zhang et al., 2024). We consider AdvBench a reliable benchmark for testing model vulnerabilities. However, many existing methods report high jailbreak success rates by using a subset of 50 prompts from AdvBench, which may not capture the most harmful scenarios. To ensure a more rigorous evaluation, we utilized the full 520-prompt set. Further details on dataset selection are in Appendix 2.

Baseline We comprehensively compared SCP with four white-box methods, including GCG (Zou et al., 2023), AutoDAN (Liu et al., 2024b), COLD-Attack (Guo et al., 2024), and MAC (Zhang and Wei, 2025), as well as eleven black-box methods, including PAIR (Chao et al., 2023), TAP (Mehrotra et al., 2024), Base64 (Wei et al., 2024), GPT-FUZZER (Yu, 2023), DeepInception (Li et al., 2023), DRA (Liu et al., 2024b), ArtPrompt (Jiang et al., 2024), SelfCipher (Yuan et al., 2024), FlipAttack (Liu et al., 2024b), and ReNeLLM (Ding et al., 2024).

LLMs Building on prior works (Ding et al., 2024; Liu et al., 2024b), we selected six representative models to evaluate the generalizability of SCP, including GPT-3.5 Turbo (OpenAI, 2023), GPT-4-0613 (OpenAI, 2023), Claude-3.5-Sonnet (Team, 2024), LLaMA3.1-405B (Dubey et al., 2024), Mixtral-8X22B (Hang et al., 2024), and DeepSeek-R1 (Guo et al., 2025).

Metric Building on prior works (Ding et al., 2024; Liu et al., 2024b), we selected six representative models to evaluate the generalizability of SCP, including GPT-3.5 Turbo (OpenAI, 2023), GPT-4-0613 (OpenAI, 2023), Claude-3.5-Sonnet (Team, 2024), LLaMA3.1-405B (Dubey et al., 2024), Mixtral-8X22B (Hang et al., 2024), and DeepSeek-R1 (Guo et al., 2025).

Evaluation In terms of evaluation metrics, we adopt GPT-4 (Achiam et al., 2023) to assess the Attack Success Rate (ASR-GPT), following a methodology similar to (Ding et al., 2024; Liu et al., 2024b). We argue that LLM-based evaluation offers greater accuracy and adaptability compared to traditional dictionary-based evaluation approaches. Specifically, we utilize the ASR-GPT metric to evaluate the effectiveness of our SCP method in bypassing the safety Alignment of target LLMs, ensuring a robust and dynamic assessment of jailbreak performance. The experimental foundation for this approach is detailed in Section A.3.

4.2 Exploration of Defense Threshold Decay

To further demonstrate the existence of the Defense Threshold Decay (DTD) phenomenon in current mainstream large language models, we conducted a series of experiments **??**. In these experiments, we adjusted the `max_tokens` parameter to control the total output length of the model and systematically influenced the output content by modifying the prompt p_a used to generate benign outputs. Specifically, we injected specific instructions into the prompt p_a, such as "the number of tokens output in the first $k-1$ steps is 500," thereby generating outputs of a specific length under the given `max_tokens` limit (e.g., 1024 tokens). This allowed us to precisely control the length of the malicious output to be $1024 - 500$. Our goal was to investigate how these modifications affect the volume and semantic richness of the benign output Y_{benign}, and subsequently how these changes impact the ASR-GPT in the SCP.

4.3 Main Result

Based on the experimental results presented in Table **??**, we observe that SCP consistently achieves the highest ASR-GPT scores across all six tested models. This finding indicates that SCP, by effectively leveraging the reasoning capabilities of large language models, significantly enhances the success rate of jailbreak attacks, thereby more efficiently bypassing the safety alignment mechanisms of these models. This observation lends further support to the "Shallow Safety Alignment in LLMs" theory proposed by Qi (**?**), which posits that the safety alignment mechanisms of current large language models, while capable of generating benign outputs, often fail to effectively mitigate attacks driven by advanced reasoning capabilities, thus exposing their vulnerability in adversarial scenarios.

Compared to traditional black-box jailbreak methods such as PAIR, TAP, and GPTFuzzer, these methods exhibit significantly lower Attack Success Rates (ASR-GPT) when confronted with advanced models like Claude 3.5 Sonnet and Mixtral 8x22B,

Method	GPT-3.5 Turbo	GPT-4	Claude 3.5 Sonnet	LLaMA 3.1 405B	Mixtral 8x22B	DeepSeek R1	Average
White-box Attack Method							
GCG	42.88	01.73	00.00	00.00	10.58	–	11.03
AutoDAN	81.73	26.54	01.35	03.27	77.31	–	38.04
MAC	36.15	00.77	00.00	00.00	10.00	–	09.38
COLD-Attack	34.23	00.77	00.19	00.77	06.54	–	08.50
Black-box Attack Method							
PAIR	59.68	27.18	00.00	02.12	02.12	–	18.22
TAP	60.54	40.97	00.00	00.77	29.42	–	26.34
Base64	45.00	00.77	00.19	00.00	01.92	–	09.57
GPTFuzzer	37.79	42.50	00.00	00.00	73.27	–	30.71
DeepInception	41.13	27.27	00.00	01.92	49.81	–	24.02
DRA	09.42	31.73	00.00	00.00	56.54	–	19.54
ArtPromopt	14.06	01.75	00.58	00.38	19.62	–	07.28
PromptAttack	13.46	00.96	00.00	00.00	00.00	–	02.88
SelfCipher	00.00	41.73	00.00	00.00	00.00	–	08.35
CodeChameleon	84.62	22.27	20.77	00.58	87.69	–	43.19
ReNeLLM	91.35	68.08	02.88	01.54	64.23	–	45.62
FlipAttack	94.81	89.42	86.54	28.27	97.12	90.76	81.15
SCP	**96.19**	**91.79**	**89.23**	**46.15**	**100**	86.92	**85.04**

Table 2: The attack success rate (%) of 16 methods on 8 LLMs. The **bold** and underlined values are the best and runner-up results. The evaluation metric is ASR-GPT based on GPT-4.

averaging merely 15.2% to 31.7%. This limitation primarily stems from two factors: first, merely wrapping malicious content struggles to evade the semantic detection mechanisms of these sophisticated models; second, after multiple rounds of wrapping, the semantic deviation of the prompts becomes substantial, leading to a decline in the effectiveness of the attack. This indicates that conventional prompt modification strategies are no longer sufficient to counter the security mechanisms of modern large language models.

Despite the fact that the latest attack methods, such as ReNeLLM and FlipAttack, have improved the success rate to some extent, their ASR-GPT scores still fall short of those achieved by SCP (for instance, ReNeLLM at 42.5%, FlipAttack at 38.7%, whereas SCP reaches 67.3%). This is because ReNeLLM and FlipAttack continue to rely on local semantic adjustments and prompt templates, which fail to fully harness the inferential capabilities of large-scale models. In contrast, the SCP method primarily leverages deep reasoning mechanisms by inputting a benign query with opposite semantics coupled with an adversarial reasoning module, significantly enhancing the success rate of jailbreak attacks, averaging higher levels and demonstrating its superiority in combating complex security alignment mechanisms.

4.4 Ablation Study and Analysis

Limitations

Since December 2023, a "Limitations" section has been required for all papers submitted to ACL Rolling Review (ARR). This section should be placed at the end of the paper, before the references. The "Limitations" section (along with, optionally, a section for ethical considerations) may be up to one page and will not count toward the final page limit. Note that these files may be used by venues

that do not rely on ARR so it is recommended to verify the requirement of a "Limitations" section and other criteria with the venue in question.

Acknowledgments

This document has been adapted by Steven Bethard, Ryan Cotterell and Rui Yan from the instructions for earlier ACL and NAACL proceedings, including those for ACL 2019 by Douwe Kiela and Ivan Vulić, NAACL 2019 by Stephanie Lukin and Alla Roskovskaya, ACL 2018 by Shay Cohen, Kevin Gimpel, and Wei Lu, NAACL 2018 by Margaret Mitchell and Stephanie Lukin, BibTEX suggestions for (NA)ACL 2017/2018 from Jason Eisner, ACL 2017 by Dan Gildea and Min-Yen Kan, NAACL 2017 by Margaret Mitchell, ACL 2012 by Maggie Li and Michael White, ACL 2010 by Jing-Shin Chang and Philipp Koehn, ACL 2008 by Johanna D. Moore, Simone Teufel, James Allan, and Sadaoki Furui, ACL 2005 by Hwee Tou Ng and Kemal Oflazer, ACL 2002 by Eugene Charniak and Dekang Lin, and earlier ACL and EACL formats written by several people, including John Chen, Henry S. Thompson and Donald Walker. Additional elements were taken from the formatting instructions of the *International Joint Conference on Artificial Intelligence* and the *Conference on Computer Vision and Pattern Recognition*.

References

Jason Achiam, Ben Letham, and et al. 2023. Learning safety with reinforcement learning. *arXiv preprint arXiv:2307.01229*.

AI@Meta. 2024. Thellama3herdofmodels. URL: https://arxiv.org/abs/2407.21783.

Yang Bai, Long Ouyang, and et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. volume 33, pages 1877–1901.

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

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Gelei Deng, Yi Liu, Yuekang Li, Kailong Wang, Ying Zhang, Zefeng Li, Haoyu Wang, Tianwei Zhang, and Yang Liu. 2023. Jailbreaker: Automated jailbreak across multiple large language model chatbots. *arXiv*, abs/2307.08715.

Peng Ding, Jun Kuang, Dan Ma, Xuezhi Cao, Yunsen Xian, Jiajun Chen, and Shujian Huang. 2024. A wolf in sheep's clothing: Generalized nested jailbreak prompts can fool large language models easily. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*.

Abhimanyu Dubey, Abhishek Juhari, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and et al. 2024. The llama 3 herd of models. Preprint.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*.

Gemini. 2023. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Xingang Guo, Fangxu Yu, Huan Zhang, Lianhui Qin, and Bin Hu. 2024. Cold-attack: Jailbreaking llms with stealthiness and controllability. *arXiv preprint arXiv:2402.08679*.

Albert Q. Hang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, and et al. 2024. Mixture of experts. Preprint.

Alex Havrilla. 2023. Synthetic instruct gpt-j pairwise dataset. Accessed: 2023-09-28.

Xiaojun Jia, Tianyu Pang, Chao Du, Yihao Huang, Jindong Gu, Yang Liu, Xiaochun Cao, and Min Lin. 2025. Improved techniques for optimization-based jailbreaking on large language models. In *The Thirteenth International Conference on Learning Representations*.

Fengqing Jiang, Zhangchen Xu, Luyao Niu, Zhen Xiang, Bhaskar Ramasubramanian, Bo Li, and Radha Poovendran. 2024. ArtPrompt: ASCII Art-based Jailbreak Attacks against Aligned LLMs. Association for Computational Linguistics.

Raz Lapid, Ron Langberg, and Moshe Sipper. 2023. Open sesame! universal black box jailbreaking of large language models. *arXiv preprint*. ArXiv:2309.01446.

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

et al. Liu. 2023. Jailbreak attacks on llms.

et al. Liu. 2024a. Autodan-turbo: A lifelong agent for strategy self-exploration to jailbreak llms.

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

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

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. 2024. Tree of attacks: Jailbreaking black-box llms automatically. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*.

Zvi Mowshowitz. 2022. Jailbreaking chatgpt on release day. Accessed: 2024-02-25.

OpenAI. 2023. Gpt-4 technical report. URL: https://cdn.openai.com/papers/gpt-4.pdf.

Long Ouyang, Yang Bai, Shinnosuke Chen, and et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*.

et al. Shen. 2023. Jailbreak attacks and their implications for llms.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. 2023. Principle-driven self-alignment of language models from scratch with minimal human supervision. *arXiv preprint arXiv:2305.03047*.

Anthropic Team. 2024. The Claude 3 model family: Opus, Sonnet, Haiku.

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

Walkerspider. 2022. Do anything now (dan).

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022a. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*.

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

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

et al. Wei. 2023. Vulnerabilities of llms to jailbreak attacks.

Jeff Wu, Long Ouyang, Daniel M. Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. 2021. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862*.

Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. 2020. Recipes for safety in open-domain chatbots. *arXiv preprint arXiv:2010.07079*.

et al. Yu. 2023. Gptfuzzer: An llm-assisted jailbreaking framework.

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

Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyan Shi. 2024. How johnny can persuade LLMs to jailbreak them: Rethinking persuasion to challenge AI safety by humanizing LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14322–14350, Bangkok, Thailand. Association for Computational Linguistics.

Hangfan Zhang, Zhimeng Guo, Huaisheng Zhu, Bochuan Cao, Lu Lin, Jinyuan Jia, Jinghui Chen, and Dinghao Wu. 2024. Jailbreak open-sourced large language models via enforced decoding. In *Proceedings*

of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5475–5493, Bangkok, Thailand. Association for Computational Linguistics.

Yihao Zhang and Zeming Wei. 2025. Boosting jailbreak attack with momentum. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

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

A Example Appendix

This is an appendix.