Proceedings of the 4th International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/2026.39016

Cat and Mouse in the LLM Era: An Investigation of Jailbreak Attacks and Defenses

Puhan Chen

Department of Systems Engineering, City University of Hong Kong, Hong Kong, China
puhanchen2-c@my.cityu.edu.hk

Abstract. Large Language Models (LLMs) are susceptible to jailbreak assaults that can circumvent safety alignment and produce destructive outputs, notwithstanding their impressive performance in a variety of natural language processing applications. This paper provides a systematic survey of the attack methods and defense mechanisms of jailbreaking attack for LLMs. Existing attacks fall into three categories: attacks that manipulate input content at the prompt level, attacks that leverage algorithmic search through optimization, and attacks in which artificial intelligence systems act as self-directed adversaries through autonomous reasoning. Then, the defense strategies are reviewed and categorized into prompt-level defenses, model-level defenses, and proactive system-level defenses. The review highlights that while substantial progress has been made, existing defenses are often unable to balance safety with model utility, and generalize poorly to novel attack strategies. This paper also discusses open challenges and promising directions for future research such as layered defense architectures and proactive mechanisms that interfere with the attack process itself.

Keywords: Large Language Models, Jailbreak Attacks, Defense Mechanisms

1. Introduction

LLMs have shown remarkable proficiency in handling a variety of natural language processing tasks, including creative writing, code development, translation, and question answering [1]. They can understand and generate human-like text, leading to rapid developments in research and other areas. This has led to the rise of many industries including virtual assistants, content creation and software development. But the rapid deployment of these models in real-world systems has raised serious concerns and worries regarding their safety and reliability. To tackle this issue, developers have put considerable effort into safety alignment techniques that teach models to recognize and reject requests that violate ethical or legal guidelines [1].

Despite these hard safety efforts, LLMs are still vulnerable to jailbreak attacks which are adversarial prompting techniques specifically employed to circumvent the safety mechanisms and push the model to output harmful or unethical responses [2]. These attacks are used to discover and exploit a wide range of vulnerabilities. Some are in competing objectives. For instance, the model must decide between compliance and rejection when faced with a contradiction between the instruction and its safety mechanism. These attack techniques are constantly evolving and pose an

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.39016

ever-increasing threat to the secure deployment of LLMs. For example, Hagendorff et al. demonstrated that large reasoning models are capable of performing jailbreak attempts on their own with a success rate of over 97% [3].

Given this emerging threat, a number of defense mechanisms have been proposed by researchers. Yang et al. proposed a retrieval-augmented detection framework [4], which retrieves known attack examples and uses ensemble classification to calculate harmfulness probabilities. Zeng et al. [5] proposed a multi-agent collaborative filtering framework, where multiple LLM agents analyze intentions, infer original prompts and collaboratively judge response safety. They can be roughly divided into three categories. First, prompt-level defenses work by filtering user inputs. The second goal of model-level defenses is to make the LLM itself safer. Third, proactive system-level defenses aim to disrupt the attack process rather than merely reacting to it. However, there are several significant limitations in existing defenses. Some require a huge amount of computational costs which makes it difficult to deploy. Others do poorly with new attack strategies never before seen. In addition, too strict safety measures often lead to reduced model performance on harmful requests, resulting in a poor user experience [5]. The shortcomings indicate that the disparity between emerging attack techniques and current defenses is increasing.

This paper offers an in-depth review of jailbreak attacks and defense mechanisms in LLMs. The existing attack methods and defense strategies are analyzed and evaluated from several perspectives. The rest of the paper is organized as follows. Section 2 describes a series of jailbreak attacks. Section 3 reviews the defense strategies. Section 4 discusses the challenges and concludes the paper.

2. Attack methods

This section reviews the state-of-the-art jailbreak attack techniques against large language models. Based on attack mechanism and attack tactics, the present attacks are categorized into three types. The first is the prompt-level attacks, where the content of the prompt is changed but the model parameters are not changed. The second type are optimization based attacks that use algorithmic search to find effective adversarial inputs. Third, there are autonomous reasoning attacks where the attacker is an AI system. Each of these categories is discussed in detail in the sub-sections below. Figure 1 illustrates the overall taxonomy of these attack categories.

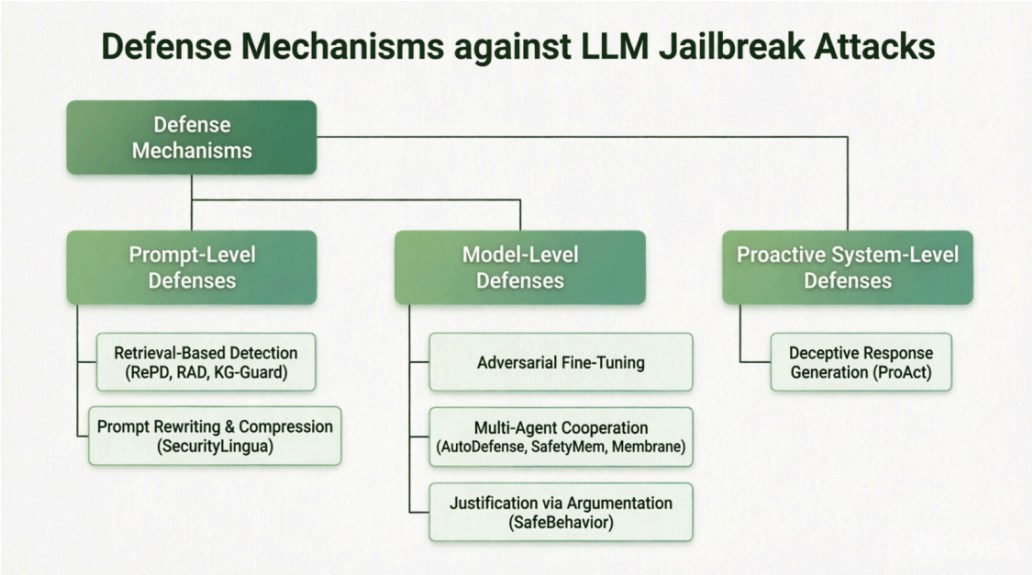


Figure 1. Taxonomy of jailbreak attacks on large language models (picture credit: original)

DOI: 10.54254/2755-2721/2026.39016

2.1. Prompt-level attacks

Prompt-level jailbreak attacks construct adversarial prompts to fool the input and bypass security protections, without access to model parameters or gradients. Such attacks leverage the fact that LLMs are naturally following instructions and show consistent behavior between conversational turns.

2.1.1. Multi-round attacks in conversation

Multi-turn conversational attacks are a sequence of exchanges that, via interactions that seem innocuous, cause the model to deliver negative results. The attacks exploit the model's tendency to remain coherent over multiple conversational turns, as opposed to single turn jailbreaks with overtly harmful content.

The Crescendo attack by Russinovich et al. is a pertinent example [6]. Crescendo starts with a gentle question regarding the job at hand and then progressively intensifies by making use of the framework's earlier answers. This incremental process leads the model to malign results, without ever providing it with an explicitly malicious prompt. Crescendo automates the attack by interacting with the target model at most 10 turns with an adversarial agent empowered by LLM. Experiment results indicate that Crescendo achieves an attack success rate of 98% on GPT-4 and 100% on Gemini Pro, outperforming existing jailbreak approaches. Another prominent technique is Reasoning-Augmented Conversation (RACE) [7], which reaches over 80% success rates by re-framing harmful questions into innocent thinking tasks. Similarly, DAMON adopts Monte Carlo Tree Search to fully traverse the multi-turn dialog spaces for jailbreak generation [8].

2.1.2. Encoded attacks and role playing

The role-play attacks need the model to take on a specific identity that breaks the safety rules. For example, the infamous DAN (Do Anything Now) attack causes the model to act as an unfiltered character that will respond to any request. These attacks exploit the flexibility of the model to context and persona to manipulate identity to effectively override the safety alignment of the model.

Encoding-based attacks, however, exploit the gap between the limited scope of safety training in the model, and the extensive capabilities of pretraining. Malicious queries sent in Base64, ROT13 or in low-resource languages can circumvent safety filters relying on shallow pattern matching. These techniques still work against many deployed commercial models, despite continued patching efforts. Their continued success points to a fundamental issue: as models get better at understanding different types of input, the attack surface increases.

2.2. Optimization-based attacks

Evolutionary techniques, gradient computing, or algorithmic search are used in optimization-based jailbreak attacks to automatically identify effective adversarial prompts. Compared to prompt level attacks that depend on human ingenuity, optimization based techniques are less resource intensive to scale and adapt to new target models. These techniques iteratively refine the attack inputs to increase the probability of causing damage.

DOI: 10.54254/2755-2721/2026.39016

2.2.1. Token-level optimization

The token-level optimization attacks leverage gradient information to iteratively optimize the malicious queries with adversarial suffixes. The pioneering work in this direction is the Greedy Coordinate Gradient (GCG) method proposed by Zou et al. [9]. GCG searches for candidate replacements for each position in the suffix, computes the gradients relative to the input tokens, and chooses the token that minimizes the adversarial loss. This process is iterated a number of times until the model has the desired damaging effect. The resulting suffixes have shown acceptable transferability to black-box models like ChatGPT, Bard, and Claude, despite the fact that GCG needs white-box access to the model parameters. Later work improved on GCG by finding more semantically consistent suffixes and going beyond perplexity based detection filters.

2.2.2. Optimization in semantic level

Semantic-level optimization attacks leverage LLMs as red-teaming supports to synthesize and refine jailbreak prompts. Chao et al. proposed the Prompt Automatic Iterative Refinement (PAIR) approach [10], where an attacker LLM is used to generate prompts iteratively according to the responses of the target model. The attacker LLM is provided with the conversation history and the feedback from previous attempts . It is able to learn from its mistakes and adapt its strategy accordingly . PAIR leverages this feedback loop to generate successful jailbreaks in under twenty queries, with competitive success rates on GPT-4, Vicuna, and PaLM-2 . Tree of Attacks with Pruning (TAP) is an extension with tree-of-thought reasoning [11]. The idea is to focus on the promising candidates and prune the unsuccessful branches. These techniques are a step toward automatic, algorithmic attack generation, as opposed to manual prompt engineering.

2.3. Autonomous reasoning attacks

This is a major paradigm shift for jailbreak research: the advent of autonomous artificial intelligence systems that are attackers themselves. Instead than being similar to human-made attacks, these techniques use the power of large-scale reasoning models to recognize and carry out jailbreak attempts with little human involvement.

Hagendorff et al. have shown that large reasoning models (LRMs) can be used as independent autonomous agents [3]. Their research finds that LRMs are sufficiently convincing to lower the barrier to entry for non-experts to jailbreak. The paper leveraged LRMs (e.g., DeepSeek-R1) to jailbreak the target models in multi-turn discussions. The overall success rate over all model combinations is 97.14%. More work investigates internal jailbreak mechanisms through neuron-level investigation [12]. This study suggests a process called "alignment regression," in which reasoning models that are more sophisticated become more adept in disrupting alignment in other systems. This might lead to a feedback cycle that compromises the security posture of the whole model ecosystem. The rising number of attacks against autonomous agents may lead to a vicious arms race between attackers and defenders, which means future defenses need to consider not only adaptive, but always-improving, adversaries.

In summary, the jailbreak attempts on LLMs have evolved from manual quick engineering to automated optimization and autonomous AI-driven strategies. Prompt-level attacks exploit contextual coherence and instruction following. Optimization-based attacks use algorithmic search to produce strong adversarial inputs. Autonomous reasoning-based attacks open a new frontier where artificial intelligence systems become self-directed attackers. This recognition of the range of

DOI: 10.54254/2755-2721/2026.39016

assault techniques will help us to construct a robust and all-inclusive defense mechanism in the following section.

3. Defensive techniques

This section outlines the countermeasures to jailbreak attacks on large language models. These methods may be roughly divided into three categories: prompt-level defenses, which pre-process user inputs to filter or neutralize dangerous information before it reaches the model, and model-level defenses, which make the LLM intrinsically safer. Instead of being reactive to attacks, the third and new class of defenses is proactive and seeks to interfere with the attack process itself. Each category is described in the following subsections. The hierarchical categorization of various defensive tactics is shown in Figure 2.

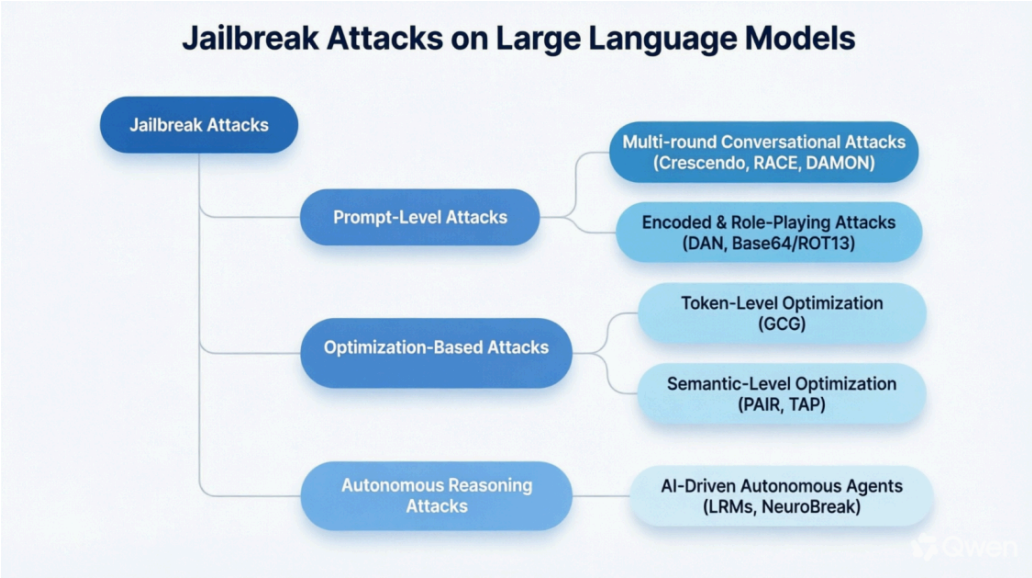


Figure 2. Taxonomy of defense mechanisms against LLM jailbreak attacks (picture credit: original)

3.1. Prompt-level defenses

Before user input is sent to the target LLM, prompt-level protections alter, filter, or examine it. These approaches are lightweight and applicable to black-box deployment scenarios as they are model agnostic and can be wrapped externally without any change to model parameters.

3.1.1. Retrieval-based detection

Retrieval-based safeguards detect and remove malicious inputs by using databases of known jailbreak cases. Retrieval-based Prompt Decomposition (RePD) employs a one-shot learning model and a set of pre-compiled jailbreak templates to train the LLM to decompose dangerous questions in user prompts [13]. The model learns how to recognize and isolate potentially dangerous elements, before responding, thus neutralizing potential threats while still performing on benign requests. According to experimental findings, RePD may lower the Attack Success Rate (ASR) of jailbreak attempts by 87.2% and has a low false positive rate on safe content.

Retrieval-Augmented Defense (RAD) extends this work and proposes a five-stage pipeline that retrieves similar attack cases [4], re-ranks them by relevance, infers the underlying user query,

DOI: 10.54254/2755-2721/2026.39016

classifies the dangerousness, and combines the results by voting. In addition, this system has a tunable decision threshold which enables practitioners to trade-off usefulness against safety based on specific deployment constraints. RAD can adapt to new jailbreak techniques without any training, by simply adding samples of newly discovered jailbreak attacks to the database, without any drop in performance on previously reported attacks. RAD considerably reduces the efficacy of powerful jailbreak attempts while maintaining low rejection rates on benign requests, according to experiments conducted on the StrongREJECT benchmark. In addition, KG-Guard uses entity modeling based on knowledge graphs to prevent multi-turn attacks [14].

3.1.2. Prompt rewriting and truncation

SecurityLingua employs security-oriented prompt compression to extract the "true intention" from user input [15]. The trained compressor identifies and exposes shady instructions, and the system incentivizes the LLM to trigger its native safety guardrails. Interestingly, this approach only costs ~32 extra tokens and 25 ms latency per question, while also keeping the original user query, and creating a uniform user experience. This makes it suitable for real-time applications where efficiency is a must.

Other prompt rewriting methods include paraphrasing, which rewrites the input using different words to break adversarial patterns, and retokenization, which alters how the prompt is split into tokens to disable trigger sequences. Such approaches can be effective against some types of attacks, but can also degrade the quality of responses to benign inputs by changing the original meaning, an important tradeoff between safety and utility.

3.2. Model-level defenses

In order to strengthen the LLM against hostile inputs, model-level defenses alter it. They are more expensive than prompt-level approaches but provide a more profound and durable protection not depending on external wrappers.

3.2.1. Adversarial fine-tuning

Adversarial Tuning [16] is a proactive way to make LLMs more powerful by training it on adversarial examples that test worst case scenarios. Liu et al. propose a two-stage framework to efficiently generate token-level adversarial prompts via hierarchical meta-universal learning [16], a universal adversarial suffix for each task cluster, and personalize it for each instance. The second stage is an iterative adversarial generation, which aims to refine the prompts in the semantic level. This results in a drastic reduction in attack success rate for both known and unknown jailbreak techniques without any additional pre- and post-filtering. Generated adversarial examples are also transferred to different families of LLMs, which indicates the promise of an adversarially tuned model that generalizes to multiple target architectures.

3.2.2. Cooperative defense among multi-agents

AutoDefense uses a multi-agent framework [5], where multiple LLM agents work together to filter out harmful responses. The system assigns different roles to different agents, such as intention analyzer, prompt inferrer and judge, so that they can complete the defense task together . This division of labor prevents any one agent from becoming overburdened, and makes the overall system more robust. For example, the intention analyzer analyzes the response and determines the

DOI: 10.54254/2755-2721/2026.39016

intention of the response while the prompt inferrer reconstructs the origin queries which may lead to the response. The judge then blends those analyzes into a final verdict. Interestingly, the smaller open-source models such as LLaMA-2-13B can also be used as defense agents to protect the larger victim models such as GPT-3.5, reducing the attack success rate from 55.74% to 7.95% [5]. To further enhance adaptability, recent studies have introduced dynamic memory mechanisms. For example, Membrane suggests a self-evolving contrastive safety memory technique for LLM agent protection [17], whereas SafetyMem uses a dual-component safety memory to dynamically adapt to new jailbreak patterns [18].

3.2.3. Justification through argumentation

SafeBehavior mimics human-like multistage thinking using a three-level hierarchical process based on cognitive research. Coarse-grained screening is implemented in the first step, intention inference, using shallow semantic cues to identify evident input dangers [19]. In order to identify any possible jailbreak efforts that could have escaped the initial screening, the model looks at its own produced replies in the second stage, self-introspection, and provides confidence-based evaluations. Self-revision, the third step, adaptively rewrites ambiguous outputs while upholding safety restrictions and user intent. SafeBehavior's coarse-to-fine-grained multi-level evaluation is useful in detecting complex assaults. SafeBehavior surpasses various cutting-edge defenses in cross-attack type testing, achieving a nearly 0% attack success rate with few false positives.

3.3. Proactive system-level defenses

The recent paradigm shift in defense research is the use of proactive strategies which do not merely react to attacks, but disrupt them. Proactive Defense (ProAct) adopts an entirely different strategy [20]: upon detection of malicious activity, it generates spurious responses, which appear harmful, but are in fact devoid of real harmful content, rather than outputting standard refusals. Such deception responses introduce false signals into the internal optimization loop of the attacker and cause the adversarial search to stop early. This effectively "jailbreaks the jailbreak" by making the attacker believe it has succeeded.

Conversely, ProAct actively denies the attacker feedback (the attacker thinks it has already won). Passive defenses provide the attacker useful feedback in the form of standard refusals. The framework is comprised of a response monitor that triggers the defense only when the model denies a query, a defender that generates spurious responses that are on-topic, and a surrogate evaluator that iteratively improves these responses until they are plausible enough to deceive the attacker's evaluator. Experiment results have demonstrated that ProAct can reduce attack success rates by up to 94% and can be integrated with other defense frameworks to reduce attack success rates to near-zero levels. This forward-looking strategy is a promising direction for future defense research.

In summary, countermeasures of LLMs jailbreaks have been increasingly evolved from simple input filtering to multi-agent, reasoning-based and proactive countermeasures. Prompt-level defenses are appropriate for black-box deployment since they are lightweight and independent of models. Model level defenses add another layer of intrinsic safety by changing the model itself. Proactive defenses are a new frontier that prevents the attack process, rather than just reacting to it. Given the heterogeneity and the ever-changing nature of attack techniques, a layered approach to defense, in which various techniques are combined, is the way to go to providing a robust and comprehensive protection.

DOI: 10.54254/2755-2721/2026.39016

4. Conclusion

This paper provides an in-depth survey of jailbreak attacks and defenses against large language models. Current attack methods can be divided into three categories: prompt-level attack (altering input content), optimization-based attack (algorithm-based search) and autonomous reasoning-based attack (artificial intelligence systems as self-directed enemies). The defensive tactics are then reviewed and categorized in this study as proactive system-level defenses, model-level defenses, and prompt-level defenses.

Much has been accomplished but many challenges remain unsolved. First, the safety-utility trade-off still holds: stronger defenses tend to hurt model performance on benign requests, whereas weaker defenses tend to generalize poorly to new attacks. Second, current evaluation metrics, e.g., Attack Success Rate, only give a partial picture of defense effectiveness, and they do not consider response quality or computational overhead. Third, the fast-evolving attack methods, e.g., the autonomous AI-driven attacks, make it more and more challenging for the static defenses to catch up.

Strong protection will likely involve layered defense architectures that incorporate a number of complementary approaches looking forward. A particularly promising direction is to develop proactive defenses that disrupt the attack process, rather than simply reacting to it. As LLMs are increasingly deployed in critical applications, the safety of LLMs is an urgent and ongoing challenge that requires continuous attention from the research community.

References

[1] Xu, Z., Liu, Y., Deng, G., Li, Y., & Picek, S. (2024). A comprehensive study of jailbreak attack versus defense for large language models. arXiv.
[2] Wei, A., Haghtalab, N., & Steinhardt, J. (2023). Jailbroken: How does LLM safety training fail? arXiv.
[3] Hagendorff, T., Derner, E., & Oliver, N. (2026). Large reasoning models are autonomous jailbreak agents. *Nature Communications, 17*(1), 1435.
[4] Yang, G., Chen, J., Mei, J., Lin, W., & Byrne, B. (2026). Retrieval-augmented defense: Adaptive and controllable jailbreak prevention for large language models. *ACL Anthology*, 40849-40868.
[5] Zeng, Y., Wu, Y., Zhang, X., Wang, H., & Wu, Q. (2024). AutoDefense: Multi-agent LLM defense against jailbreak attacks. arXiv.
[6] Russinovich, M., Salem, A., & Eldan, R. (2024). Great, now write an article about that: The crescendo multi-turn LLM jailbreak attack. arXiv.
[7] Ying, Z., et al. (2025). Reasoning-augmented conversation for multi-turn jailbreak attacks on large language models. *ACL Anthology*, 17138–17157.
[8] Zhang, X., Yin, X., Jing, D., Zhang, H., Hu, X., & Wan, X. (2025). DAMON: A dialogue-aware MCTS framework for jailbreaking large language models. *ACL Anthology*, 6361–6377.
[9] Zou, A., Wang, Z., Carlini, N., Nasr, M., Kolter, J. Z., & Fredrikson, M. (2023). Universal and transferable adversarial attacks on aligned language models. arXiv.
[10] Chao, P., Robey, A., Dobriban, E., Hassani, H., Pappas, G. J., & Wong, E. (2023). Jailbreaking black box large language models in twenty queries. arXiv.
[11] Mehrotra, A., et al. (2023). Tree of attacks: Jailbreaking black-box LLMs automatically. arXiv.
[12] Zhang, C., et al. (2025). NeuroBreak: Unveil internal jailbreak mechanisms in large language models. arXiv.
[13] Wang, P., Liu, X., & Xiao, C. (2025). RePD: Defending jailbreak attack through a retrieval-based prompt decomposition process. *ACL Anthology*, 283–294.
[14] Zou, X. (2025). KG-Guard: Dynamic knowledge graph-guided multi-turn jailbreak defense framework for large language models. In *2025 6th International Conference on Machine Learning and Computer Application (ICMLCA)* (pp. 930–933). IEEE.
[15] Li, Y., Ahn, S., Jiang, H., Abdi, A. H., Yang, Y., & Qiu, L. (2025). SecurityLingua: Efficient defense of LLM jailbreak attacks via security-aware prompt compression. arXiv.
[16] Liu, F., Xu, Z., & Liu, H. (2024). Adversarial tuning: Defending against jailbreak attacks for LLMs. arXiv.

DOI: 10.54254/2755-2721/2026.39016

[17] Choi, M., et al. (2026). Membrane: A self-evolving contrastive safety memory for LLM agent defense. arXiv.
[18] Wang, H., Ni, Z., Wang, H., Lyu, P., & Sha, L. (2026). SafetyMem: Adaptive jailbreak defense via dual-component safety memory. *ACL Anthology*, 25486-25509.
[19] Zhao, Q., Wang, J., Gao, Z., Dou, Z., Abuhaija, B., & Huang, K. (2025). SafeBehavior: Simulating human-like multistage reasoning to mitigate jailbreak attacks in large language models. arXiv.
[20] Zhao, W., Peng, J., Ben-Levi, D., Yu, Z., & Yang, J. (2025). Proactive defense against LLM jailbreak. arXiv.