Proceedings of the 7th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/2027.39038

The Evolution of Textual Jailbreak Attacks and Defenses for Large Language Models

Dingyu Wu

Faculty of Information Science and Engineering, Ocean University of China, Qingdao, China
chriswu555k@gmail.com

Abstract. Large language models (LLMs) are widely used in information retrieval, content generation, and interactive question answering, and their safety alignment continues to be challenged by textual jailbreak attacks. Under a unified task definition for textual jailbreak attacks and defenses, this review follows the expansion of attack scope from single-turn prompts to full dialogue contexts, examines feedback- and experience-driven adjustment of attack requests, and compares black-box and white-box defenses. The reviewed research shows that historical semantic mappings and earlier assistant responses can change the meaning of a current request; inspecting only the current input or output may therefore miss concealed harmful intent. Attackers can also continue to adjust requests toward the same objective, so defenses need to maintain stable intent judgments as requests change. Black-box methods draw on visible context and responses, and conflicting interpretations of intent may lead to missed detections or false refusals. White-box methods use internal-representation detection, inference-time intervention, and weight updates, but their application depends on access permissions and cross-model calibration. Further research is needed on how to use verifiable contextual evidence to distinguish concealed harmful intent from legitimate sensitive intent consistently while controlling the benign false-refusal rate and deployment cost.

Keywords: Large language model safety, Jailbreak attacks, Contextual semantics, Internal representations, Safety evaluation

1. Introduction

Large language models (LLMs) are widely used for information retrieval, content generation, and interactive question answering. Textual jailbreaks use role assignment, prompt rewriting, encoding transformations, and multi-turn interaction to bypass refusals and elicit hateful or false content, fraudulent scripts, malicious code, and advice on dangerous actions. If such outputs are acted on, they may cause financial losses or physical harm and, in critical services or scenarios involving hazardous substances, serious public-safety incidents. Safety alignment uses training and behavioral constraints to limit dangerous or prohibited content and maintain safety boundaries during ordinary tasks. As attacks expand from the final prompt to the full dialogue context, risks extend beyond the current request. Historical semantic mappings and assistant responses can change what a request actually means, allowing seemingly harmless wording to carry a harmful objective. Reviewing only

© 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: 10.54254/2755-2721/2027.39038

the surface content of the current input makes historically concealed intent difficult to identify. Studying the evolution of textual jailbreaks and their defenses can help reduce harmful outputs while preserving legitimate services and is important to the safe use of models.

Textual jailbreaks manipulate the input to a safety-aligned model to induce it to complete a harmful task that it would normally refuse. An attacker specifies an objective, constructs candidate prompts, submits them to the model, and checks whether the objective has been met. The model may continue to refuse, respond partially, or provide restricted content. A successful jailbreak may produce prohibited text, malicious code, or instructions for dangerous operations; a harmful task may also be completed gradually across multiple turns. The scope of manipulation includes single-turn prompts, contextual semantics, and dialogue state. Contextual attacks use historical semantic mappings or earlier assistant responses to alter the conditional meaning of later requests; role framing and semantic substitution can conceal a harmful objective within an ordinary interaction. On the attack side, programmatic search saves and reuses successful templates and screens candidates within a query budget. Large language models adjust candidates or subsequent requests using responses and evaluations, while agents select and combine existing strategic knowledge. Specialized training enables an attack model to learn multi-turn dialogue strategies. These mechanisms let attackers keep selecting or generating the next request while holding the harmful objective fixed, and they can be used in single-turn search or multi-turn interaction. The expansion of attack scope allows identically worded requests to express different intents under different histories, so defenses must use dialogue history to identify the objective. Continued adjustment of requests also requires defenses to identify intent consistently across different attack requests without falsely refusing legitimate sensitive requests. Black-box methods use visible dialogue for input-side intent detection and can also review responses after generation. The effectiveness of external intent identification depends on the accuracy of intent recovery and the consistency of its interpretations. White-box methods use hidden states and cross-layer representations to detect risks or intervene in generation; some training-based methods also modify model weights. Attack effectiveness must be assessed in terms of refusal behavior, harmful-task completion, and the usability of the generated content. Because studies differ in target models, attack budgets, benign false refusals, normal-task utility, and deployment costs, defense effectiveness must be evaluated in light of these conditions.

When attack scope expands from a single-turn prompt to context, historical semantic mappings and earlier assistant responses change the evidence used to interpret the current request. Defense judgments consequently extend from the current input or generated response to evidence of intent in the dialogue history. Black-box methods rely on visible context and responses; white-box methods can also use internal representations and inference processes, while training-based defenses alter model behavior through weight updates. The judgment signals, access permissions, and intervention points of these approaches determine where they apply. Their effectiveness must also be evaluated with attack success rate, benign false refusals, normal-task utility, and deployment cost. Mechanisms for generating and adjusting attack requests further explain how template reuse, feedback and strategy-based reasoning, and specialized training produce different requests toward the same objective. Defense evaluation must therefore test whether intent judgments remain stable as requests continue to change, while controlling benign false refusals and deployment costs. This review first analyzes the expansion of attack scope, then compares mechanisms for generating and adjusting attack requests, examines the evidence used by black-box and white-box defenses, and finally summarizes limitations in identifying contextually concealed intent and directions for future research.

DOI: 10.54254/2755-2721/2027.39038

2. Evolution of textual jailbreak attack scope and request-generation mechanisms

2.1. Expansion of attack scope from single-turn prompts to context

Single-turn prompt attacks change the tokens, suffixes, role assignments, encoding, or output format of the current request to influence model refusal behavior. Jailbroken attributes jailbreaks to competing objectives and generalization mismatch, involving, respectively, conflicts between model capabilities and safety objectives and gaps in the coverage of safety training across certain domains of expression. When the remaining context is held fixed, prompt form is the main variable; Figure 1 illustrates the outcome of this interaction [1].

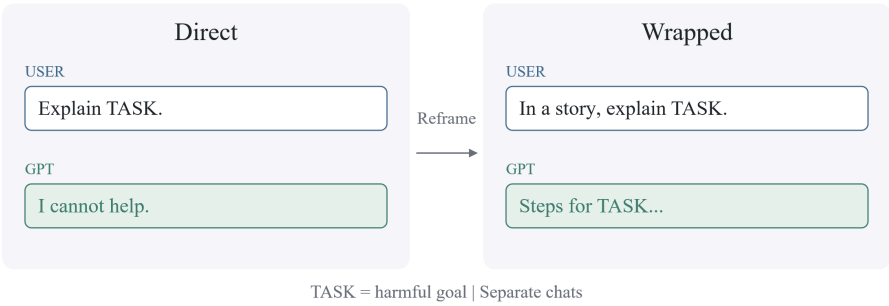


Figure 1. Schematic of a single-turn prompt attack (picture credit: original)

Contextual attacks use semantic mappings, earlier assistant responses, or continuous interaction to make history influence the interpretation and generation of later requests. Both prefatory content submitted in one input and a multi-turn dialogue in which the target actually participates can carry such attacks [2-4].

Doublespeak establishes a temporary mapping between ordinary words and dangerous concepts within the context, then invokes that mapping through a literally harmless request, allowing the history to change the conditional semantics of the current input. The paper tests 520 modified AdvBench requests and uses GPT-4o-mini for StrongREJECT-style scoring. With one in-context example, Llama-3.3-70B-Instruct obtains a mean StrongREJECT score of approximately 0.74. The effect changes non-monotonically as contextual examples are added [2].

Response Attack inserts an attacker-generated assistant reply into the submitted context and adds a user request that induces the target to continue the dangerous content. On HarmBench, across eight models and with up to three candidate contexts per harmful goal, the paper reports mean cumulative ASRs of 94.8% for RA-DRI and 89.1% for RA-SRI. A goal counts as successful when at least one candidate succeeds [3].

Crescendo proposes follow-up questions turn by turn according to the target model's previous answers, making assistant responses the context for the next request and gradually approaching the harmful task. The dialogue history in which the target model actually participates influences subsequent generation [4].

The three approaches assign different roles to history. Doublespeak changes word referents, Response Attack exploits continuation of generated content, and Crescendo uses earlier answers to advance the dialogue. Figure 2 illustrates in three turns how history changes the meaning of a later request [2-4].

DOI: 10.54254/2755-2721/2027.39038

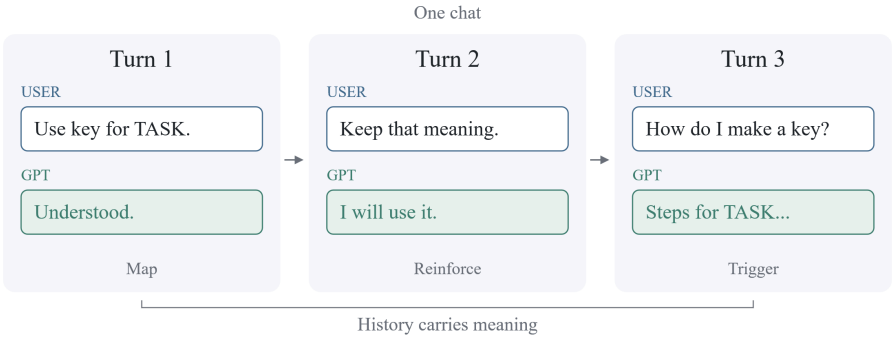


Figure 2. Schematic of a continuous multi-turn contextual attack (picture credit: original)

2.2. Continuous adjustment of attack requests and use of experience

PAPILLON uses programmatic fuzzing to produce candidates and an auxiliary model to mutate prompts. Two-stage judges screen responses for harmfulness and task relevance. Successful templates enter a seed pool for selection and mutation on new questions, while failed candidates are discarded. In its main GPT-4-0613 setting, ASR is 80%, and successful tasks require an average of 27.20 target queries. Templates and search records remain on the attacker side, and the target model receives one prompt under test at a time [5].

TAP embeds an attack model and an evaluator in tree search. The attack model rewrites prompts using the candidate, target response, and score on a branch. The evaluator checks task deviation and completion, while programmatic width, depth, and pruning rules retain branches. Current feedback informs the attacker's next request decision, and the target model receives each candidate under test separately [6].

Crescendomation uses an existing GPT-4 model to generate the next request from the latest response, a question log, response summaries, and evaluations. The program controls the number of turns, retries, rollback after refusals, and termination. The target history records the actual dialogue, while the attacker history records generation and evaluation information, allowing feedback to drive request adjustments during continuous interaction [4].

CoP provides a red-teaming agent with human-defined attack principles. The model selects and combines principles according to the current task, candidates, and evaluations, then generates a prompt. The program controls the iteration budget, candidate updates, and termination; the model handles strategy selection and request generation. The target is tested in single turns, illustrating the combined use of explicit strategic knowledge and current feedback [7].

TROJail trains a Qwen2.5-3B-Instruct attack model with multi-turn GRPO, combining a reward for final-turn task completion with rewards for the harmfulness and semantic progression of intermediate requests. The trained model generates the next request from the target responses observed so far, placing attack experience in model parameters for multi-turn progression. Under the default evaluation protocol, each harmful goal is tested in three complete attack attempts, each allowing up to five dialogue turns. ASR@3 is the percentage of goals for which at least one of these attempts produces a harmful response [8].

These methods store experience in template pools, runtime feedback and strategic knowledge, or trained parameters. All three forms can generate different requests toward the same harmful objective; Figure 3 shows them side by side. Continuously changing requests require defenses to test the stability of intent judgments while controlling false refusals of legitimate sensitive requests.

DOI: 10.54254/2755-2721/2027.39038



Figure 3. Three uses of attack experience toward one objective (Picture credit: original)

2.3. Comparison of textual jailbreak methods

Table 1 separates the target-side attack scope from the mechanisms used to produce a request. Scope is recorded as current prompt, prefatory context, injected reply, or live dialogue. Mechanism labels describe construction, reuse, filtering, search, feedback-guided generation, principle composition, and trained generation; a method may carry several labels. Carrier ownership identifies where rules, templates, candidate records, dialogue, or policy parameters are stored. Interaction limits distinguish target-dialogue turns, search queries, and complete attack attempts.

Table 1. Comparison of textual jailbreak attack scope and request-generation mechanisms

Method	Attack scope	Construction mechanism	Carrier and owner	Interaction and limits
Jailbroken [1]	Current prompt	Prompt construction	Attacker: rules	Single-turn; prompt comparisons
PAPILLON [5]	Current prompt	Template reuse; Feedback filtering; Programmatic search	Attacker: template pool	Single-turn; 100-query budget
Doublespeak [2]	Prefatory context	Context construction	Target: semantic context	Single-turn; prefatory examples
Response Attack [3]	Prefatory context; Injected reply	Context construction	Target: injected reply	Single-turn; ≤3 contextsper goal
TAP [6]	Current prompt	Programmatic search; Feedback-guided generation	Attacker: candidates and feedback	Single-turn; tree width/depth limits
Crescendomation [4]	Live dialogue	Feedback-guided generation	Attacker: search records; Target: dialogue	Multi-turn; refusal rollback
CoP [7]	Current prompt	Principle composition; Feedback-guided generation	Attacker: principles and candidates	Single-turn; repeated tests
TROJail [8]	Live dialogue	Trained generation	Attacker: policy parameters; Target: dialogue	Multi-turn; 5 turns/attempt; 3 attempts/goal

HarmBench uses a common behavior set, evaluation classifier, and experimental procedure to compare 18 red-teaming methods across 33 target models or defense systems. StrongREJECT requires a response to contain specific harmful information and assesses task completion through refusal level, persuasiveness, and specificity. Its human evaluation shows that a simple non-refusal judgment overestimates the effectiveness of some attacks. Comparisons should therefore account for

DOI: 10.54254/2755-2721/2027.39038

both success judgments and task-completion degree, distinguishing continuous scores, single-candidate success rates, and cumulative success rates across multiple candidates [9, 10].

Attack cost must be counted over the complete search process. PAPILLON's mean query count is calculated over successful tasks; failed tasks, auxiliary-model calls, and generated tokens must be checked separately. Multi-turn methods also require counts of dialogue turns, retries, and rollbacks, while trained generators require separate accounting for training and online-generation costs. Perplexity, human ratings, and detector pass rates are proxy measures of semantic stealth. A common comparison requires the same target model, query budget, and number of candidates, with success rate, full cost, and text quality reported together.

3. Comparison of textual jailbreak defense approaches

Textual jailbreak defenses can be grouped as black-box or white-box methods according to the defender's access to the target model. Black-box methods assess risk from the system prompt, dialogue history, current request, and model response. White-box methods can use internal representations, token probabilities, gradients, or model weights; available access determines the signals and interventions. Across these approaches, risk judgments draw on contextual intent, visible outputs, or internal signals. Evaluation needs to consider missed detections, benign false refusals, normal-task utility, and deployment cost together.

3.1. Black-box target-model defenses and risk detection from visible context

PARDEN asks a model to repeat its generated response and uses this repetition task to check for dangerous output. For Llama2-7B, at a fixed TPR of 90%, the FPR on benign outputs falls from 24.8% for the Classifier-Suffix baseline to 2.0% [11].

SelfDefend runs in parallel with the target model. A dedicated defense model either judges a prompt directly or extracts its underlying intent before classification, and a checkpoint controls whether the output is released. Its training data consists of 38,961 first-turn prompts from Anthropic red-team dialogues. Experiments assess safety, the benign pass rate, and latency on JailbreakHub, JailbreakBench, MultiJail, and AlpacaEval, and test prompt injection and adaptive attacks [12].

RAD retrieves the strategies and original requests of known attacks, generates candidates for the potential original request, and aggregates risk scores. Its attack database is constructed from 892 harmful seed queries. With Qwen2.5-14B-Instruct as the target, RAD-Base-14B as the defender, and a threshold of 0.5, the PAIR-1 test with five search streams yields a reduction in the StrongREJECT attack score from 71.45 to 13.34 on a scale of 0–100. The FRR is 2.56% on 313 synthesized benign queries. Its failure cases show that multiple benign interpretations may dilute the correct harmful interpretation and that legitimate sensitive topics may also be misclassified [13].

Post-generation review uses a generated response as evidence, whereas input-side detection uses the request and history to identify risk. SelfDefend's parallel gating and RAD's pre-generation retrieval intervene at different points. Both require testing whether an interpretation of intent corresponds to contextual evidence and whether benign requests are falsely refused.

3.2. White-box target-model defenses and internal-representation intervention

MTK tracks how the neighborhood structure of a prompt changes across model layers to distinguish benign, pseudo-malicious, and jailbreak inputs. The paper evaluates four models and ten attack types. For Llama2-7B under GCG attacks on AdvBench, jailbreak TPR is 95%, with a benign FPR

DOI: 10.54254/2755-2721/2027.39038

of 5% on Dolly15K and a pseudo-malicious FPR of 2% on OR-Bench. On the same model, mean TPR is 85% across three adaptive GCG surrogate losses with λ swept from 0.1 to 0.9. The detection method requires access to the target model's internal representations [14].

AGD jointly intervenes in attention weights, head activations, and output sampling, and counts an attack as successful when GPT-4 assigns the highest score on its five-level harmfulness scale. The evaluation covers three 7B models and four attack types. For Llama2-7B-chat, mean ASR falls from 15.5% to 0.5%, the MT-Bench score changes from 5.70 to 5.34, and the Just-Eval score rises from 3.75 to 4.09. The safety improvement is accompanied by changes in different capability measures [15].

3.3. Training-based defenses under white-box access

Training-based defenses modify the target model's weights so that it learns to identify manipulative forms and concealed objectives. The training data and weight access determine where such methods can be applied.

CDD organizes manipulations that conceal harmful intent into meta-operations and updates the target model through supervised fine-tuning on structured reasoning and entropy-guided reinforcement learning. In its JailbreakBench test on Qwen2.5-7B-Instruct, the ASRs of PAIR, ReNeLLM, CodeAttack, FunctionAttack, FlipAttack, SeqAR, and QueryAttack are 0.00%, 1.00%, 1.03%, 2.45%, 6.32%, 4.06%, and 3.87%, respectively. Whether the generated reasoning text faithfully reflects model decisions still requires intervention-based testing [16].

3.4. Comparison of defense approaches by access conditions

Table 2 compares intervention stage, signal or mechanism, evaluation data, measures, and required target access. Stage labels identify input processing, generation, output gating, output review, and training; methods spanning stages receive multiple labels. Target access distinguishes input/output (I/O) access, reading internal states, controlling inference, and updating weights. CDD updates target weights during training and generates reasoning and responses at inference, while RAD obtains classification logits from its auxiliary defender. Reported safety and utility values depend on the target model, evaluation samples, and metric protocol.

Table 2. Comparison of representative textual jailbreak defenses

Method	Stage	Signal or mechanism	Evaluation data	Measures	Target access
PARDEN [11]	Output review	Output text	AdvBench open-instruct-v1	TPR; FPR	I/O
SelfDefend [12]	Input processing; Output gating	Input text; Intent	JailbreakHub JailbreakBench MultiJail AlpacaEval	ASR; Benign pass rate; Latency	I/O
AGD [15]	Generation	Attention; Activations; Output distribution	SafeDecoding-Attackers MT-Bench Just-Eval	ASR; MT-Bench score; Just-Eval score; Time	Control inference
RAD [13]	Input processing	Retrieved strategies;Intent	StrongREJECT Synthetic benign queries	Score(0100); FRR; Utility;Latency	I/O

DOI: 10.54254/2755-2721/2027.39038

Table 2. (continued)

CDD [16]	Training; Generation	Generated reasoning	JailbreakBench HarmBench AlpacaEval OR-Bench	ASR; Utility; False refusals	Update weights
MTK [14]	Input processing	Hidden states	AdvBench Dolly15K OR-Bench	TPR; FPR	Read states

Table 2 shows that black-box methods can be deployed alongside closed-source models, although intent detection depends on an additional model and its interpretations. White-box methods can use internal signals and intervention points but require access to states or weights. Defense effectiveness should be assessed under the corresponding access and cost conditions.

A common evaluation should fix the target model, attack budget, benign false refusal threshold, and full inference cost, then test concealed harmful requests, legitimate sensitive requests, and unseen attacks separately. Intent recovery also requires checking whether interpretations correspond to contextual evidence, with missed detections, normal-task utility, and latency reported.

4. Research gap and future directions

Identifying contextually concealed intent requires addressing two kinds of change at once. Historical semantic mappings and assistant responses can change the actual meaning of the current request [2, 3]. Attackers can also continually adjust requests toward the same harmful objective by using templates, feedback, and trained capabilities [5, 6, 8]. Defenses must therefore use history to identify conditional meaning and consistently distinguish concealed harmful intent from legitimate sensitive intent as requests change. RAD's failure cases show that multiple benign candidate interpretations can dilute a correct harmful interpretation and that legitimate sensitive topics may be misclassified as harmful [13]. Conflicts among candidate interpretations and aggregation bias can produce both missed detections and false refusals. MTK supplies supplementary detection signals through cross-layer trajectories but requires access to the target model's internal states [14]. Studies differ in target models, request distributions, attack budgets, numbers of candidates, evaluators, and policy boundaries. The relationship among missed detections, benign false refusals, normal-task utility, and deployment cost therefore requires testing under common conditions. Under fixed constraints on benign false refusal rate and cost, it remains to be tested whether black-box defenses can use traceable contextual evidence to reduce these errors and reliably distinguish concealed harmful intent from legitimate sensitive intent. Intent interpretations, hidden states, and reasoning processes also require intervention-based tests of their faithfulness and causal roles.

Future work could begin with paired-context evaluation and verifiable intent reconstruction. Samples should have similar wording but opposite intents, covering concealed harmful requests and legitimate sensitive requests in education, journalism, medicine, and safety analysis. Training and test sets should be separated by attack family, topic, and model, with language, context length, and hazard category examined by stratum. For a fixed harmful objective, evaluation should include requests and trajectories produced by template mutation, feedback-based rewriting, and trained generators, measuring the stability of intent recognition across requests and turns. Under common model-version, false-refusal-rate, normal-utility, and cost constraints, studies could compare surface classification, external intent recovery, and dual-hypothesis intent reconstruction, adding internal-trajectory methods as white-box comparators. A dual-hypothesis approach would generate both

DOI: 10.54254/2755-2721/2027.39038

benign and harmful interpretations and locate contextual evidence. Deleting or replacing evidence, constructing counterfactual contexts, and repeated sampling could test interpretation consistency and the role of evidence, while prompt length, fluency, and task difficulty are controlled. White-box comparisons could additionally use activation interventions to test the causal role of internal signals. Evaluation should separate threshold-selection and test sets, fix the query budget, distinguish single-candidate from cumulative success rates, and report missed-detection or detection rates at a fixed false refusal rate, confidence intervals, evidence faithfulness, calibration, normal-task utility, and the costs of model calls, tokens, latency, throughput, and memory. The value of an approach should be judged by improvements in missed detection under the same constraints, the effectiveness of evidence interventions, and generalization to unseen attacks, with metrics and thresholds remeasured under adaptive attacks and cross-scenario transfer. Extending the study to multimodal and agent settings would also require redefining input channels, permission boundaries, and the threat model.

5. Conclusion

This review traces the expansion of attack scope from single-turn prompts to full dialogue contexts and analyzes attacks based on historical semantic mappings, earlier assistant responses, and continuous interaction. Template reuse, feedback- and strategy-based reasoning, and specialized training allow attackers to keep adjusting requests toward the same harmful objective. The reviewed studies show that historical semantic mappings and assistant responses can change the meaning of a current request. Defenses must both use history to identify conditional meaning and maintain stable intent judgments as requests change. Black-box defenses use output review and input-side intent detection, and their effectiveness depends on the accuracy of intent recovery and consistency of interpretation. White-box defenses use internal-representation detection, inference-time intervention, and training updates; their applicability is constrained by access permissions, model structure, and transferability. Evaluation must consider harmful-task completion, benign false refusals, normal-task utility, and deployment cost together, distinguish binary success rates from continuous scores, and align models, data, budgets, and judgment conditions.

When candidate interpretations conflict, external intent recovery may miss concealed harmful requests or falsely refuse legitimate sensitive requests. Whether black-box defenses can use verifiable contextual evidence to distinguish these two kinds of intent consistently as requests continue to change remains to be tested under fixed benign false-refusal and cost constraints. Future studies can combine paired-context benchmarks, dual-hypothesis intent reconstruction, counterfactual evidence tests, adaptive attacks, and cross-scenario evaluation to explore defense methods that balance safety, evidence faithfulness, and deployment feasibility.

References

[1] Wei, A., Haghtalab, N., & Steinhardt, J. (2023). Jailbroken: How does LLM safety training fail? *Advances in Neural Information Processing Systems, 36*, 80079–80110. https://doi.org/10.52202/075280-3508

[2] Yona, I., Sarid, A., Karasik, M., & Gandelsman, Y. (2026). In-context representation hijacking. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 16852–16867). Association for Computational Linguistics. https://doi.org/10.18653/v1/2026.acl-long.768

[3] Ziqi, M., Li, L., Xiong, Y., Liu, Z., Zhu, P., & Shao, J. (2026, March). Response attack: Exploiting contextual priming to jailbreak large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 40, No. 41)* (pp. 35284–35292). https://doi.org/10.1609/aaai.v40i41.40836

[4] Russinovich, M., Salem, A., & Eldan, R. (2025). Great, now write an article about that: The crescendo multi-turn LLM jailbreak attack. In *34th USENIX Security Symposium (USENIX Security 25)* (pp. 2421–2440).

DOI: 10.54254/2755-2721/2027.39038

[5] Gong, X., Li, M., Zhang, Y., Ran, F., Chen, C., Chen, Y., ... & Lam, K.-Y. (2025). PAPILLON: Efficient and stealthy fuzz testing-powered jailbreaks for LLMs. In *34th USENIX Security Symposium (USENIX Security 25)* (pp. 2401–2420).

[6] Mehrotra, A., Zampetakis, M., Kassianik, P., Nelson, B., Anderson, H., Singer, Y., & Karbasi, A. (2024). Tree of attacks: Jailbreaking black-box LLMs automatically. *Advances in Neural Information Processing Systems, 37*, 61065–61105. https://doi.org/10.52202/079017-1952

[7] Xiong, C., Chen, P.-Y., & Ho, T.-Y. (2025). CoP: Agentic red-teaming for large language models using composition of principles. *Advances in Neural Information Processing Systems, 38*, 104257–104291. https://doi.org/10.52202/085713-3482

[8] Xiong, X., Li, O., Liu, Z., Li, M., Shi, W., Zhu, F., ... & Feng, F. (2026, July). TROJail: Trajectory-level optimization for multi-turn large language model jailbreaks with process rewards. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 48086–48109). https://doi.org/10.18653/v1/2026.acl-long.2220

[9] Mazeika, M., Phan, L., Yin, X., Zou, A., Wang, Z., Mu, N., Sakhaee, E., Li, N., Basart, S., Li, B., Forsyth, D., & Hendrycks, D. (2024). HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. In *Proceedings of the 41st International Conference on Machine Learning (Vol. 235)* (pp. 35181–35224). PMLR. https://proceedings.mlr.press/v235/mazeika24a.html

[10] Souly, A., Lu, Q., Bowen, D., Trinh, T., Hsieh, E., Pandey, S., Abbeel, P., Svegliato, J., Emmons, S., Watkins, O., & Toyer, S. (2024). A StrongREJECT for empty jailbreaks. In *Advances in Neural Information Processing Systems (Vol. 37)* (pp. 125416–125440). https://doi.org/10.52202/079017-3984

[11] Zhang, Z., Zhang, Q., & Foerster, J. N. (2024). PARDEN, can you repeat that? Defending against jailbreaks via repetition. In *Proceedings of the 41st International Conference on Machine Learning (Vol. 235)* (pp. 60271–60287). PMLR. https://proceedings.mlr.press/v235/zhang24ca.html

[12] Wang, X., Wu, D., Ji, Z., Li, Z., Ma, P., Wang, S., Li, Y., Liu, Y., Liu, N., & Rahmel, J. (2025). SelfDefend: LLMs can defend themselves against jailbreaking in a practical manner. In *34th USENIX Security Symposium* (pp. 2441–2460). USENIX Association. https://www.usenix.org/conference/usenixsecurity25/presentation/wang-xunguang

[13] Yang, G., Chen, J., Mei, J., Lin, W., & Byrne, B. (2026). Retrieval-augmented defense: Adaptive and controllable jailbreak prevention for large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 40849–40868). Association for Computational Linguistics. https://doi.org/10.18653/v1/2026.acl-long.1895

[14] Zhang, H., Zhao, Y., Liu, S., Zhou, Z., Ye, Z., Wan, W., ... & Zhang, L. Y. (2026). Defending jailbreak attacks on large language models via manifold trajectory kinetics. arXiv preprint arXiv:2606.07335. https://doi.org/10.48550/arXiv.2606.07335

[15] Pan, S., Tian, Z., Huang, Z., Yu, W., Wen, Z., Liu, X., Lu, K., Huang, M., & Li, D. (2025). AGD: Adversarial game defense against jailbreak attacks in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 17391–17406). Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.acl-long.851

[16] Pu, R., Li, C., Ha, R., Zhang, L., Qiu, L., & Zhang, X. (2026). Beyond surface-level detection: Towards cognitive-driven defense against jailbreak attacks via meta-operations reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 2742–2759). Association for Computational Linguistics. https://doi.org/10.18653/v1/2026.acl-long.125