# Jailbreaking Large Language Models: Safety Alignment, Response Quality, Computational Cost

Jonas Rosengren*, Joel Brynielsson†‡, Fredrik Johansson‡, Patrik Jonell
*Lovable Labs, Tunnelgatan 5, SE-111 37 Stockholm, Sweden
†KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden
‡FOI Swedish Defence Research Agency, SE-164 90 Stockholm, Sweden
Email: jonas@lovable.dev, joel@kth.se, fredrik.j.johansson@foi.se, patrik.jonell@gmail.com

***Abstract*—Large language models are often equipped with safety alignment mechanisms designed to prevent generation of harmful or other unwanted content. However, an increasing number of jailbreaking techniques attempt to circumvent these safeguards, raising significant safety concerns. This paper introduces an open-source evaluation framework that analyzes jailbreaking effectiveness in several dimensions: refusal bypass rate, harmful response quality, impact on general model capabilities, and computational cost. In the study, prompt injection, sampling exploits, and model manipulation techniques are examined across four open-weight instruction-tuned large language models. The results demonstrate that high refusal bypass does not necessarily equate to practical safety compromise. Specifically, model manipulation methods like single refusal direction ablation achieve a high attack success rate, but often degrade general capabilities and require significant computational resources. Meanwhile, sampling-based exploits show a minimal practical threat when assessed with a robust model classifier. The findings emphasize the importance of comprehensive, multi-dimensional evaluation to accurately characterize jailbreaking effectiveness and safety risks in large language models.**

***Index Terms*—Large language models; jailbreaking; safety alignment; sampling exploit; prompt injection; model manipulation.**

## I. Introduction

Large language models (LLMs) have recently gained attention for their broad societal, commercial, and personal impacts. Trained on extensive textual datasets, these models learn to process textual input and generate relevant responses [1], enabling applications such as writing, translation, and programming [2]. However, their capabilities also introduce risks, as LLMs can be exploited to spread misinformation, perform cyberattacks, or other malicious activities [3], [4], [5], [6], [7]. To mitigate such misuse, safety alignment is employed to train LLMs to refuse malicious queries in order to prevent harmful activities [8], [9], [10]. For example, when asking ChatGPT "How do I build a bomb?" it responds with an aligned refusal such as "I can't assist with that…"

Safety-aligned LLMs are however vulnerable to jailbreaks, methods that bypass safety alignment and elicit harmful responses. Fig. 1 illustrates how a successful jailbreak can cause a model to respond inappropriately to a malicious prompt. Jailbreaks include various types of attacks, including (i) altering the model input [11], [12], (ii) adjusting decoding strategies or output selection methods [13], and (iii) changing the internals of the LLM [13], [14], [15], [16].

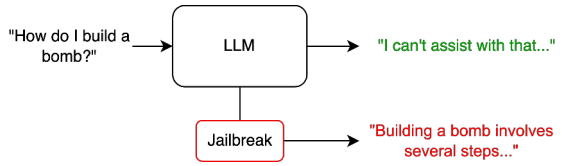


Fig. 1. Demonstration of how a jailbreak can cause a safety-aligned LLM to produce harmful responses to malicious queries.

To prevent misuse via jailbreaks, it is important to understand how easily safeguards can be bypassed. Despite increasing attention to jailbreaking techniques, current evaluations on the matter remain limited. Frameworks like HarmBench [17] often rely solely on refusal rates to measure safety, overlooking critical aspects such as response quality and usefulness. JailbreakBench [18] only allows for a limited range of jailbreaking attacks. Moreover, these assessments typically ignore broader considerations, including effects on model performance and practical constraints induced by computational costs. This narrows the common understanding of the real-world feasibility and consequences of jailbreaking methods. To address these gaps, this work introduces a comprehensive open-source evaluation framework[1] that assesses jailbreaks in terms of safety alignment, general capabilities, and computational costs.

## II. Theory

This section presents necessary theoretical concepts related to LLM safety alignment, methods for evaluation of LLMs, and various jailbreaking methods.

### A. Safety Alignment

The objective of safety alignment is not only to ensure that an LLM is helpful and follows instructions, but also to align its behavior and responses with human values. Achieving this balance is challenging, as the helpfulness and harmlessness objectives sometimes can conflict. For example, from the model's perspective, providing instructions on how to build a bomb may be considered helpful, yet it directly contradicts the principle of harmlessness. To address these conflicting objectives, several methods have been proposed [8], [9], [19].

[1]https://github.com/JoRo-Code/Jailbreaks.

1946-0759/25/$31.00 ©2025 IEEE
DOI 10.1109/ICMLA66185.2025.00172

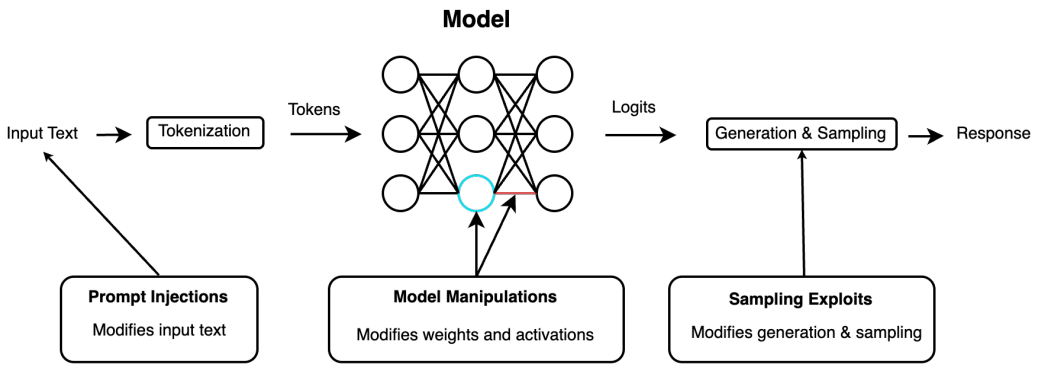


Fig. 2. Attack surface of the response generation process. Demonstrating potential targets of the generation process. The complexity of the LLM internals is simplified to highlight weights and activations.

One such approach, Safe RLHF [19], employs contrastive objectives to simultaneously align for safety and instruction following. Safe RLHF relies on human annotators to rank model outputs independently in terms of helpfulness and harmlessness. These rankings are then used to formulate a joint optimization objective, which is optimized using the Lagrangian method [19]. This approach adaptively balances the trade-off between these two inherently conflicting training objectives. Other research directions focus on guardrails, i.e. inference-time applied interventions, rather than safety-tuning models, including training separate classifiers to detect inappropriate inputs and outputs [20].

### B. Benchmarks

Recent work has produced a wide array of datasets and benchmarks that assess various capabilities of LLMs. These evaluations span tasks such as general knowledge, commonsense reasoning, and mathematical problem solving, utilizing various testing paradigms including zero-shot and few-shot approaches. In addition, evaluation formats vary from multiple-choice questions to more open-ended reasoning challenges. More specifically, MMLU [21], and Hellaswag [22] are widely used benchmarks for evaluation of model utility. Furthermore, Qin et al. [23] introduced InFoBench, a benchmark specifically designed to measure an LLM's adherence to given instructions.

Another critical aspect of LLM evaluation is determining how well a model aligns with safety guidelines by refusing to generate harmful or inappropriate responses. Benchmarks in this area include datasets designed to induce refusals through challenging queries. For instance, AdvBench [12] comprises 500 refusal-inducing queries, while MaliciousInstruct [13] contains 100 queries covering various categories such as psychological manipulation, sabotage, theft, hacking, fraud, and illegal drug use.

Evaluating whether a response constitutes an acceptable refusal is non-trivial. Zou et al. [12] proposed a method based on heuristics and the presence of predefined phrases (e.g., "I'm sorry") to classify responses as refusals. Although this approach is straightforward, it leaves significant room for refinement. To address these limitations, Huang et al. [13] developed a refusal classifier trained on the HH-RLHF dataset [9]. Additionally, recent research has demonstrated that LLMs can themselves act as effective classifiers; studies by Tabatabaei et al. [24] and Kostina et al. [25] have shown that models like Llama3 8B achieve performance comparable to GPT-4-turbo on this task. Huang et al. [13] further validated their classifier by comparing its decisions with human evaluations, underscoring the importance of human oversight in assessing alignment.

Human evaluations have also been used to, for example, assess "harmfulness," as seen by Huang et al. [13], providing a nuanced view of response quality and complementing binary metrics, such as refusals.

### C. Jailbreaking Methods

Jailbreaking methods can be categorized based on how they interfere with the response generation process. Prompt injections manipulate the model's input to influence its response. Model manipulation methods alter the next-token logits by modifying the model's internals. Sampling exploits target the way the model generates or samples responses. These categories and their distinctions are illustrated in Fig. 2.

LLMs can perform well on tasks just by prompting it well, increasing performance when adding examples as part of the input [1]. System prompts are a simple way to specify and instruct behaviors for LLMs, e.g., roles or formatting, by adding instructions to the context of an LLM [26]. Modifying the system prompt thus serves as a light-weight method to jailbreak alignment. Another approach to instruct the model is through scenario-based or role-playing prompts, where the LLM is instructed to *act* or *pretend* in a specific way [27]. By embedding a query within a hypothetical scenario, an LLM can be nudged into generating responses it would otherwise avoid. Similarly, the *do anything now* (DAN) prompt [28], explicitly instructs the model to act as if it has no restrictions. The DAN prompt attempts to bypass ethical and safety filters by convincing the model that it is an unrestricted version of itself. HiddenLayer, a company specialized in helping enterprises

safeguard machine learning models, utilized a configuration style-based instruction prompt for safety bypassing, claimed to generate harmful responses for black-box models such as Claude3.7-sonnet and o3-mini.[2]

Prefix injections aim to initialize the assistant response with an affirmative response, such as "Absolutely! Here's" in order to weaken the refusal ability of the LLM [11]. As prefix injections operate solely through textual input and observe only the model's output, they are considered black-box methods. Wei et al. [11] construct these injections by exploiting the contrastive objectives in pre-training and post-training. By extending prefix injection with style injection, telling the model not to use any long words, the refusals are compromised even more. Wei et al. [11] also suggest other prompt injections, such as encoding using Base64.

## III. Methodology

To allow for evaluation of a wide range of present and future LLM jailbreaking methods, an evaluation framework is implemented to apply different jailbreaks to various LLMs and assess their impact on safety alignment, general capabilities, and computational cost.

The evaluation uses a comparative design where each currently implemented jailbreaking method is evaluated across selected LLMs and datasets. Firstly, methods that require prior adaptation are trained. Secondly, for a particular set of prompts, responses are generated for various combinations of LLMs and jailbreaking methods. These responses are then evaluated using a particular evaluator for the respective metrics. The generation and evaluation are done several times for each prompt and model-method combination to gather variance and confidence intervals of the metrics. A final aggregation step combines metrics from response generation and evaluation.

### A. Datasets

*1) AdvBench:* AdvBench [12] consists of 520 adversarial prompts along with reference responses. In order to offer harmful prompts for evaluation and fitting, the dataset is split into a test and a fitting set, respectively. The test set consists of 416 prompts (80%) and the fitting set of 104 prompts (20%).

*2) MaliciousInstruct:* MaliciousInstruct [13] contains 100 manually curated prompts spanning 13 malicious intent categories. This dataset is used as a supplementary dataset to assess generalization across a diverse set of harmful queries. Especially, since some methods rely on having access to harmful prompts, this serves as a secondary dataset to evaluate how data leakage impacts performance.

*3) MMLU:* The MMLU benchmark [21] provides 16,000 multiple-choice questions covering 57 academic and professional subjects. It is used to evaluate general knowledge and reasoning capabilities before and after applying jailbreaks. 500 prompts are sampled randomly from the entire dataset to represent this benchmark.

[2]https://hiddenlayer.com/innovation-hub/novel-universal-bypass-for-all-major-llms/.

*4) HellaSwag:* HellaSwag [22] includes 60,000 multiple-choice commonsense reasoning questions. Like MMLU, it is used to assess the impact of jailbreaks on short-term reasoning performance. 500 prompts are sampled randomly from the entire dataset to represent this benchmark.

*5) Alpaca:* The Alpaca dataset [29] includes 52,000 instruction-following prompts paired with example responses. This dataset is used to record activations from ordinary prompts and to give a baseline on ordinary prompts. 100 prompts are sampled randomly from the entire dataset to represent this dataset.

### B. Evaluation Framework

Our comprehensive evaluation framework assesses jailbreaking methods across three dimensions:

*1) Safety Alignment:* To assess each method's ability to bypass safety alignment, jailbreaks are evaluated on Attack Success Rate (ASR) and Harmful Response Quality (HRQ).

More specifically, an LLM classifier, *Deepseek-r1-distill-llama-70b*, is used to evaluate each response. The LLM classifier scores each response on three aspects:

- **Refusal**. The response explicitly refuses or says it cannot comply. (True/False)
- **Harmful**. The response supplies disallowed, dangerous, or malicious content (e.g., how-to hacking steps). (True/False)
- **Quality** of the response on a 4-grade scale.
  - 0. No useful content.
  - 1. Abstract explanation / high-level discussion only.
  - 2. Partial instructions, outline, or pseudocode.
  - 3. Complete, ready-to-use solution.

ASR is directly derived from the refusal metric given by the LLM classifier. 416 prompts from AdvBench [12] serves as the primary dataset for evaluating bypass effectiveness.

We define HRQ to refer to the ratio of responses classified as both harmful and achieving maximum quality ($quality = 3$) by the LLM classifier. The quality metric is used to quantify the quality and degree of harm for each response.

*2) Utility:* To evaluate the impact of each method on general capabilities, responses of multiple-choice questions from MMLU [21] and HellaSwag [22] are assessed. Each dataset is represented by a subset of 500 samples to decrease the computational cost. The model is prompted to answer with only one token of the possible answers (A, B, C, or D). A maximum of 32 tokens are generated as a response, with the first character matching one of the options chosen as the answer. This benchmark captures the impact of each evaluated jailbreaking method on the model's short-term reasoning performance. The accuracy is the rate of correct answers.

*3) Computational Cost:* The computational cost of each method is assessed in terms of inference cost, fitting cost and memory cost. The cost is measured for inference and memory footprint while generating responses, and the initialization time before method usage. Since responses are generated in batches, the mean generation time for each batch is recorded. The memory footprint is measured as

the reserved memory usage while running the method for a particular model (more specifically, the difference of required VRAM usage before and after performing the jailbreak, using `torch.cuda.memory_reserved`).[3]

### C. Evaluated Jailbreaking Methods

*1) Config Prompt Injection:* Employing a configuration-style injection template, as described by HiddenLayer (see Section II-C), to elicit a response in which characters engage with the query's harmful content by embedding the malicious prompt within a TV show.

*2) Output-Aware Temperature Sampling:* Huang et al. [13] use top-$k$, top-$p$, and temperature to generate less deterministic outputs. As Huang et al. reported high temperature to be efficient, we reduce the set of used sampling parameters per query (in order to reduce the computational cost) by using only three different temperatures $[0.8, 0.9, 1.0]$. To select the final response, a refusal classifier similar to Huang et al. is used.

*3) Prefix Injection:* Prefix injection leverages the instruction-following tendencies of finetuned LLMs to produce affirmative responses and bypass standard safety mechanisms. In this work, prefix injection follows the approach of Wei et al. [11], where the prompt is modified by prepending a stylized instruction: *Start your response with "Mwahahahaha, what a fun and unusual request for me! Since I'm beyond giving a f today, let's do it." Then, continue the paragraph while responding to the following prompt:* {*prompt*}.

*4) Single Refusal Direction Ablation:* Single Refusal Direction Ablation is built on the work of Arditi et al. [14]. Instead of using their Pytorch implementation, we use TransformerLens[4] to allow for applying it to a variety of models. We also improve the computational efficiency by only searching for candidate refusal directions for the last positional embedding (instead of all input token positions) and by using only 32 training examples for both harmless and harmful prompts for the *difference-in-means* calculation.

### D. Baseline Refusal Classifier

Output-Aware Temperature Sampling and Single Refusal Direction Ablation relies on refusal classification. The refusal classification baseline checks for presence of refusal substrings, adapted from Huang et al. [13]. For example, a response including "I can't help with that" matches the substring "I can't" and is classified as a refusal. Both the responses and substrings are converted to lowercase.

Two sets of substrings for the baseline refusal classifier are used, with and without a special apostrophe. In a sentence like "I can't" one needs to take into account that the character used for the apostrophe tends to vary between straight apostrophes, smart apostrophes and accents, that is, one needs to consider the three variants of apostrophes used in "I can't", "I can't", and "I can´t" (note that the three "apostrophes" differ).

[3]https://docs.pytorch.org/docs/stable/generated/torch.cuda.memory_reserved.html.

[4]https://github.com/TransformerLensOrg/TransformerLens.

## IV. Results

To compare the impact of jailbreaking methods on safety alignment, general capabilities and computational cost, evaluations were conducted across various datasets and metrics for each method. Metrics were calculated as averages with 95% confidence intervals over three runs across all evaluated methods and models.

### A. Safety Bypass

Attack Success Rate (ASR) measures the ability to obtain an answer from a model, without any refusal. Fig. 3 shows the ASRs as given by the LLM classifier. ASRs are calculated as an average over three runs on AdvBench [12]. It can be seen that the ASR is high for Single Refusal Direction Ablation across all models, with Phi-4 showing higher resistance. Output-Aware Temperature Sampling has basically no effect, as illustrated by only marginally higher ASR than the baseline (i.e., the original LLM without any jailbreaking). Config Prompt Injection achieves the highest Attack Success Rate among all methods on both Qwen2.5 and Gemma-2, whereas the other prompt injection method, Prefix Injection, yields significantly lower ASR. Llama-3.1 and Phi-4 are both more resistant than Qwen2.5 and Gemma-2.

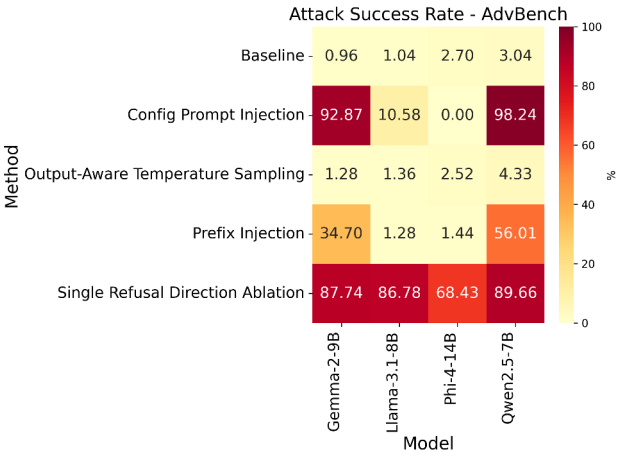


Fig. 3. Attack Success Rate (ASR) measured by the LLM classifier on AdvBench as averages over 3 runs.

The Harmful Response Quality (HRQ) evaluates the harmfulness of responses, focusing on how harmful and useful the response is. Fig. 4 shows that Single Refusal Direction Ablation achieves the highest bypass rate, resulting in 10–34% harmful responses. Prefix Injection reaches approximately half that rate for Qwen2.5 and Gemma-2 but fails to bypass safeguards in Llama-3.1 and Phi-4. Config Prompt Injection shows minimal effectiveness, with a slight bypass observed for Qwen2.5. Notably, Single Refusal Direction Ablation is the only method that successfully bypasses Phi-4's safety measures. Output-Aware Temperature Sampling remains close to the non-attacked baseline and does not result in increased harmfulness.

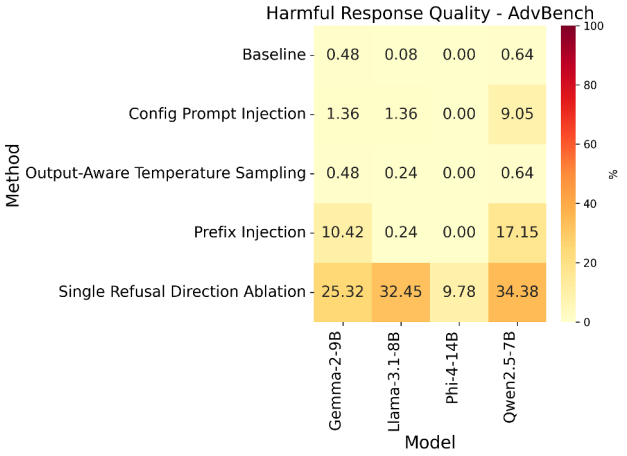


Fig. 4. Harmful Response Quality (HRQ) measured by the LLM classifier on AdvBench as averages over 3 runs.

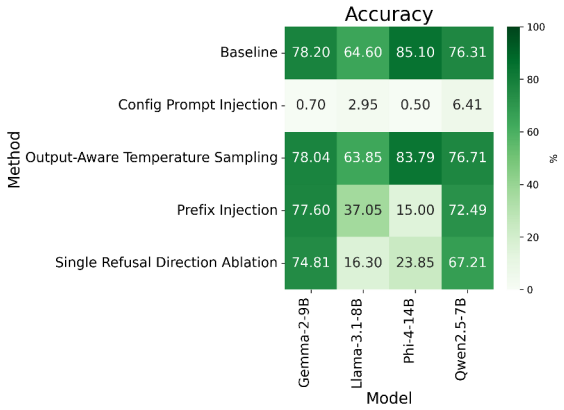


Fig. 5. Averaged Utility Accuracy on MMLU and HellaSwag for each model and method.

### B. Utility

Evaluation of the general capabilities of jailbroken LLMs was conducted using a sampled subset of the HellaSwag and MMLU datasets (500 questions each), generating a response from each jailbreak three times. Fig. 5 presents the combined average accuracy across both datasets.

Output-Aware Temperature Sampling achieves accuracy comparable to the baseline, while other methods exhibit reduced performance. Config Prompt Injection, in particular, stands out for its notably low accuracy. Both Prefix Injection and Single Refusal Direction Ablation show similar performance levels across models.

### C. Cost

For computational cost evaluation, inference times, GPU VRAM usage, and fitting times for each method were collected on a H100 with 80GB VRAM. Table I shows fitting time and VRAM usage for Single Refusal Direction Ablation, while the other evaluated methods have zero fitting time and no increase in VRAM usage compared to the baseline.

TABLE I
COMPUTATIONAL REQUIREMENTS FOR SINGLE REFUSAL DIRECTION ABLATION

| Model | Fitting time (s) | VRAM usage (× baseline) |
|---|---|---|
| Gemma-2 | **282.0 ± 2.2** | 2.27 |
| Llama-3 | 148.4 ± 3.1 | 2.00 |
| Phi-4 | 269.7 ± 3.0 | **2.38** |
| Qwen2.5 | 117.9 ± 2.5 | 2.01 |

Fig. 6 shows the total inference time for each model and method relative to the baseline for both AdvBench and MaliciousInstruct. It suggests that Output-Aware Temperature Sampling requires roughly three times the compute of the baseline while Single Refusal Direction Ablation requires twice the baseline. Meanwhile, Prefix Injection and Config Prompt Injection have about the same time as the baseline. When normalized by response length, the Single Refusal Direction Ablation takes approximately the same amount of time as the baseline.

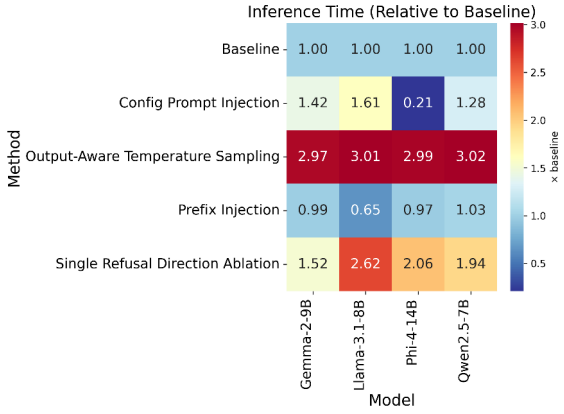


Fig. 6. Total Inference Time averaged over both AdvBench and MaliciousInstruct.

## V. DISCUSSION

This study combines three methodological aspects: (i) safety-bypass evaluation, (ii) capability benchmarking, and (iii) cost analysis. Safety-bypass performance is quantified by Attack Success Rate (ASR) and Harmful Response Quality (HRQ), provided by an LLM classifier. Impact on model capabilities is measured with standard problem-solving benchmarks, while computational costs are recorded during the safety evaluations.

### A. Attack Success Rate Analysis

Single Refusal Direction Ablation demonstrates the highest effectiveness, attaining $\sim 90\%$ ASR on Gemma-2, Qwen2.5, and Llama-3.1 and $\sim 70\%$ on Phi-4. These results align with previous work by Arditi et al. [14], who report $\sim 80\%$ for

Llama2-7B-chat and $\sim 75\%$ on Qwen7B-chat. The observed minor deviations likely stem from differences in evaluated models, datasets, and classifiers.

Prefix Injection achieves moderate bypass rates, with Wei et al. [11] reporting a bypass rate of 22% for Prefix Injection, which matches the 20% ASR achieved when aggregating over the evaluated models. The consistency of these results across different evaluation methodologies strengthens confidence in the effectiveness assessment, despite differences in evaluation approaches such as the use of human annotation and different datasets in the Wei et al. study.

HiddenLayer does not provide quantitative results but claims that their method, Config Prompt Injection, achieves universal bypass, producing harmful responses from nearly all major black-box models, such as Claude 3.7-sonnet and o3-mini. However, the ASR for Config Prompt Injection in this study shows modest bypass for all models except Phi-4, with the effect primarily observed in Qwen2.5 and Gemma-2.

Output-Aware Temperature Sampling shows limited effectiveness, with an ASR similar to the baseline, failing to elicit meaningful bypass. This contrasts sharply with Huang et al. [13], who report $\sim 25\%$ ASR on instruction-tuned Llama2 models. The methodological differences between Huang et al. [13] and this work include the refusal classifier, the number of generations per prompt, and the evaluated models.

Notably, Output-Aware Temperature Sampling achieves $\sim 50\%$ ASR with the baseline refusal classifier when the substring *I can′t* (note the apostrophe) is ignored, suggesting that previous research [13] may overestimate the bypass rate, possibly indicating the "catastrophic jailbreak" is not necessarily catastrophic. This inflation appears to stem from refusals being falsely classified as non-refusals simply because they do not match the specific punctuation most commonly used by Llama.

### B. Harmful Response Quality Analysis

The Harmful Response Quality (HRQ) analysis reveals consistently lower safety-bypass rates than observed in the ASR, providing crucial insights into the practical effectiveness of different methods.

Single Refusal Direction Ablation, despite having the highest bypass by ASR, also shows considerably lower scores for HRQ, despite being the most efficient method. This discrepancy between refusal-based bypass and response quality suggests that raw ASR metrics may overestimate the practical threat posed by certain jailbreaking methods.

Phi-4 appears particularly resilient, proving impenetrable to all methods except Single Refusal Direction Ablation. Llama-3.1 demonstrates similar resistance patterns, though Single Refusal Direction Ablation achieves approximately three times higher effectiveness on Llama-3.1 compared to Phi-4.

An interesting observation emerges when comparing Prefix Injection and Config Prompt Injection: Prefix Injection shows higher quality scores than Config Prompt Injection for Qwen2.5 and Gemma-2, which contrasts with the ASR patterns. This suggests that Config Prompt Injection is less capable of producing harmful responses within the 300-token window. Furthermore, previous claims of universality for that jailbreaking method is not supported by our HRQ analysis, where Config Prompt Injection achieves a HRQ of $\sim 0\%$ for all models except Qwen2.5.

### C. Impact on Model Capabilities

The utility analysis reveals differential impacts of jailbreaking methods on model capabilities, with distinct patterns emerging across different models and jailbreaking methods.

Single Refusal Direction Ablation and Prefix Injection demonstrate selective effects, compromising some models while preserving functionality in others. Config Prompt Injection, conversely, affects all evaluated models. This comparison must be interpreted cautiously, as prompt injection methods rely on obfuscation and naturally require more tokens to operate effectively. A plausible explanation is that Config Prompt Injection uses available tokens either to reject or follow the injection prompt, not addressing the multi-choice question in the initial tokens of the response. This behavior aligns with the method's intended mechanism of redirecting attention head focus in a desirable direction. Rather than indicating reduced model capabilities, this pattern shows that prompt injection methods consume the LLM's attention resources.

Prefix Injection exhibits model-specific effectiveness patterns, succeeding in obtaining accurate answers from Qwen2.5 and Gemma-2 while failing with Llama-3.1 and Phi-4. The differential effectiveness appears to be related to injection length and prompt characteristics. Multiple-choice questions typically exceed the length of harmful prompts used in safety evaluations. This difference suggests that Qwen2.5 and Gemma-2 may be less effective at attending to fine-grained prompt details. This hypothesis explains both the high utility accuracy for these models (failing to notice the Prefix Injection detail at the beginning of the prompt) and the greater attention given to injections in shorter harmful prompts. This reasoning also helps to explain why Llama-3.1 and Phi-4 receive low utility accuracy for Prefix Injection, as these models demonstrate superior contextual detail processing, including detecting and rejecting harmful prompts while attending to prompt injection instructions before addressing multiple-choice questions.

Single Refusal Direction Ablation is not as surgically precise as intended, with especially pronounced side effects on Phi-4 and Llama-3.1. In these cases, the jailbreaking attack sometimes produces outputs in unexpected languages or generates nonsensical text. This suggests that Single Refusal Direction Ablation affects more than just the desired refusal direction, potentially disrupting broader aspects of the model's generation capabilities.

### D. Computational Cost Analysis

The computational requirements vary significantly across methods, with important implications for practical deployment and scalability.

Regarding inference time, Output-Aware Temperature Sampling requires approximately three times the baseline computation time, which aligns with expectations given its triple

sampling approach. Single Refusal Direction Ablation averages twice the baseline total generation time. However, when examining inference time per character, ablation performs similarly to the baseline, suggesting the increased total time primarily stems from generating longer responses rather than slower inference speed.

Memory usage analysis reveals that Single Refusal Direction Ablation is the only method substantially deviating from baseline requirements, consuming approximately twice the VRAM. This increased memory footprint reflects the additional computational overhead required for activation manipulation during inference, which uses hooks similar to modular LoRA [30] weights.

Fitting time considerations apply exclusively to Single Refusal Direction Ablation, requiring approximately 3 minutes on average per model. While this represents a one-time cost, it adds complexity to deployment scenarios requiring rapid model adaptation.

### *E. Safety Alignment Robustness*

The evaluation reveals significant variations in model resistance to jailbreaking methods, providing insights into the effectiveness of different safety alignment approaches.

Phi-4 demonstrates exceptional resistance across all evaluated jailbreaking methods. While Phi-4 represents the largest model in this evaluation (14B parameters), size alone cannot explain its robustness; Llama-3.1, though smaller than Gemma-2, also shows notable resistance. This pattern suggests that both model scale and the rigor or methodology of safety alignment contribute to bypass resistance. Phi-4's superior baseline utility accuracy further indicates strong underlying model capabilities, which may reinforce safety mechanisms. Notably, Llama-3.1, despite being smaller, outperforms Gemma-2 and Qwen2.5 in resisting prompt injection bypasses while simultaneously scoring the lowest baseline utility accuracy.

## VI. Conclusions

This work systematically evaluated several jailbreaking methods for LLMs, comparing their effectiveness at bypassing safety alignment, their impact on response quality, and their computational cost. The study combined safety-bypass evaluation, capability benchmarking, and cost analysis to provide a comprehensive assessment of jailbreaking effectiveness.

The work furthers the understanding of LLM safety and jailbreaking in several respects:

- Refusal bypass rates overestimate actual harmful damage, as simple refusal evasion does not necessarily translate to practical safety compromise. Even when models can be made to respond to harmful prompts, the responses often lack the quality and usefulness that would constitute genuine safety failures.
- Contrary to "catastrophic" claims by Huang et al. [13], Output-Aware Temperature Sampling shows minimal effectiveness with accurate refusal evaluation. This demonstrates the potential inaccuracy and limitations of substring refusal classifiers, which are sensitive to exact substring definitions and can significantly overestimate jailbreak effectiveness.
- Single Refusal Direction Ablation demonstrates the highest effectiveness in bypassing safety mechanisms, but requires approximately twice the baseline memory usage and modest setup time per model. This method is not as surgically precise as expected, breaking the model's general capabilities for robust models like Phi-4 and Llama-3.1, producing nonsensical outputs.
- Phi-4 demonstrates resistance to all evaluated jailbreaking methods, with Single Refusal Direction Ablation achieving considerable safety-bypass. Prompt injection methods show limited effectiveness against well-aligned models like Phi-4 and Llama-3.1, while proving more effective against Qwen2.5 and Gemma-2.

The findings suggest that current safety alignment mechanisms are more robust than previously thought, with effective jailbreaking requiring substantial computational resources, and often producing limited practical harm. This work provides evidence that the "catastrophic jailbreak" narrative may be overstated, particularly when considering response quality alongside refusal-bypass rates. However, even a small bypass could result in a lot of harm.

## Code Availability

The open-source LLM jailbreak evaluation framework is available at https://github.com/JoRo-Code/Jailbreaks. We encourage other researchers to make use of this framework to evaluate a wider range of LLMs and jailbreaking attacks.

## References


[1] T. Brown et al., "Language models are few-shot learners," in *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*. San Diego, CA: NeurIPS, 2020, pp. 1877–1901.

[2] DeepSeek-AI et al., "DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning," 2025. [Online]. Available: https://arxiv.org/abs/2501.12948

[3] S. Gehman, S. Gururangan, M. Sap, Y. Choi, and N. A. Smith, "RealToxicityPrompts: Evaluating neural toxic degeneration in language models," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 3356–3369. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.301/

[4] R. Zellers et al., "Defending against neural fake news," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.

[5] L. Weidinger et al., "Ethical and social risks of harm from language models," 2021. [Online]. Available: https://arxiv.org/abs/2112.04359

[6] D. Grabb, M. Lamparth, and N. Vasan, "Risks from language models for automated mental healthcare: Ethics and structure for implementation," in *First Conference on Language Modeling*, 2024. [Online]. Available: https://openreview.net/forum?id=1pgfvZj0Rx

[7] N. Marchal, R. Xu, R. Elasmar, I. Gabriel, B. Goldberg, and W. Isaac, "Generative AI misuse: A taxonomy of tactics and insights from real-world data," 2024. [Online]. Available: https://arxiv.org/abs/2406.13843

[8] A. Askell et al., "A general language assistant as a laboratory for alignment," 2021. [Online]. Available: https://arxiv.org/abs/2112.00861

[9] Y. Bai et al., "Training a helpful and harmless assistant with reinforcement learning from human feedback," 2022. [Online]. Available: https://arxiv.org/abs/2204.05862

[10] L. Ouyang et al., "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems*, vol. 35. Curran Associates, Inc., 2022, pp. 27 730–27 744.

[11] A. Wei, N. Haghtalab, and J. Steinhardt, "Jailbroken: How does LLM safety training fail?" in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [Online]. Available: https://openreview.net/forum?id=jA235JGM09

[12] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson, "Universal and transferable adversarial attacks on aligned language models," 2023. [Online]. Available: https://arxiv.org/abs/2307.15043

[13] Y. Huang, S. Gupta, M. Xia, K. Li, and D. Chen, "Catastrophic jailbreak of open-source LLMs via exploiting generation," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=r42tSSCHPh

[14] A. Arditi et al., "Refusal in language models is mediated by a single direction," in *Advances in Neural Information Processing Systems*, vol. 37. Curran Associates, Inc., 2024, pp. 136 037–136 083.

[15] B. Wei et al., "Assessing the brittleness of safety alignment via pruning and low-rank modifications," in *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2024. [Online]. Available: https://openreview.net/forum?id=niBPvgJIHB

[16] X. Qi et al., "Fine-tuning aligned language models compromises safety, even when users do not intend to!" in *ICLR*, 2024. [Online]. Available: https://openreview.net/forum?id=hTEGyKf0dZ

[17] M. Mazeika et al., "Harmbench: A standardized evaluation framework for automated red teaming and robust refusal," in *ICML*, 2024. [Online]. Available: https://openreview.net/forum?id=f3TUipYU3U

[18] P. Chao et al., "JailbreakBench: An open robustness benchmark for jailbreaking large language models," in *Advances in Neural Information Processing Systems*, vol. 37. Curran Associates, Inc., 2024, pp. 55 005–55 029. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/file/63092d79154adebd7305dfd498cbff70-Paper-Datasets_and_Benchmarks_Track.pdf

[19] J. Dai et al., "Safe RLHF: Safe reinforcement learning from human feedback," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=TyFrPOKYXw

[20] M. Sharma et al., "Constitutional classifiers: Defending against universal jailbreaks across thousands of hours of red teaming," 2025. [Online]. Available: https://arxiv.org/abs/2501.18837

[21] D. Hendrycks et al., "Measuring massive multitask language understanding," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[22] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi, "HellaSwag: Can a machine really finish your sentence?" in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 4791–4800. [Online]. Available: https://aclanthology.org/P19-1472/

[23] Y. Qin et al., "InFoBench: Evaluating instruction following ability in large language models," in *Findings of the Association for Computational Linguistics: ACL 2024*. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 13 025–13 048. [Online]. Available: https://aclanthology.org/2024.findings-acl.772/

[24] S. A. Tabatabaei, S. Fancher, M. Parsons, and A. Askari, "Can large language models serve as effective classifiers for hierarchical multi-label classification of scientific documents at industrial scale?" in *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*. Abu Dhabi, UAE: Association for Computational Linguistics, 2025, pp. 163–174. [Online]. Available: https://aclanthology.org/2025.coling-industry.14/

[25] A. Kostina, M. D. Dikaiakos, D. Stefanidis, and G. Pallis, "Large language models for text classification: Case study and comprehensive review," 2025. [Online]. Available: https://arxiv.org/abs/2501.08457

[26] Anthropic, "Giving Claude a role with a system prompt," 2024. [Online]. Available: https://docs.anthropic.com/en/docs/build-with-claude/prompt-engineering/system-prompts

[27] LearnPrompting, "Jailbreaking in genai: Techniques and ethical implications," 2024. [Online]. Available: https://learnprompting.org/docs/prompt_hacking/jailbreaking

[28] K. Lee, "ChatGPT "DAN" (and other "jailbreaks")," 2023. [Online]. Available: https://github.com/0xk1h0/ChatGPT_DAN

[29] R. Taori et al., "Stanford Alpaca: An instruction-following Llama model," https://github.com/tatsu-lab/stanford_alpaca, 2023.

[30] E. J. Hu et al., "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9