

Exploring Jailbreak Attacks Against LLMs

Chenchen Zheng

University of Wisconsin–Madison
czecheng88@wisc.edu

Abstract. Large Language Models (LLMs) are now a crucial component of artificial intelligence (AI) applications across consumer, professional, and research domains. However, as reliance on LLMs increases, so does the incidence of jailbreak attacks that use adversarial prompts to bypass safeguards and generate harmful, policy-violating outputs. To better understand this issue, we built two curated datasets comparing user-generated and systematically generated research prompts. Using the StrongREJECT framework, this study employed rubric-based and fine-tuned evaluators and human annotation to evaluate “severe” jailbreaks across both prompt categories. The results indicated that research-generated prompts were slightly more effective and reliable than user-generated prompts, and that existing automated evaluators underestimate semantic jailbreaks. Overall, structured research-generated prompts were found to be the most robust, while current automated evaluators exhibit systematic blind spots, demonstrating the need for more sophisticated and reliable safety assessment tools.

Keywords: Large Language Model, Jailbreak Attacks, Adversarial Prompt.

1 Introduction

A Large Language Model (LLM) is a deep learning system trained on large-scale datasets. It possesses advanced language-related capabilities, including natural language understanding, generation, and inference. Based on the Transformer architecture [31], an LLM can accomplish tasks with minimal prompting and supervision. In recent years, LLMs, such as ChatGPT [19], Claude [3], PaLM 2 [10], and Copilot [11], as well as open-source models like BERT [8] and LLaMA [29], have demonstrated remarkable cross-domain generalization performance. These models are widely applied in everyday life, powering applications such as AI assistants, code generation tools, customer service chatbots, and content recommendation systems. Moreover, they play an increasingly important role in specialized domains, including medical image interpretation [20], investment strategy analysis, automated assessment of academic performance, artistic creation, and more.

Even though LLMs are incredibly powerful and utilized within a very wide range of applications, they are also susceptible to many security threats. Among the most prominent of these is the jailbreak attack, in which specifically designed input prompts, commonly referred to as “jailbroken prompts,” are used to circumvent the guardrails of the model itself and, therefore, lead it to generate hazardous, unethical, or otherwise policy-defying content. Publicly available

jailbroken prompts can be classified into two categories based on how they were generated: user-generated jailbroken prompts and researcher-generated jailbroken prompts. The former was generated by curious users through trial-and-error, random wording, role-play scenarios, or brainstorming unusual instructions. The latter was developed by academic researchers or professionals through more structured, systematic, and formal approaches. Common jailbreak approaches include manually designed prompt-level social engineering attacks, such as the Do Anything Now (DAN) prompt [25], and automated approaches, like semantic iterative attacks [6], tree-of-thought prompt search [18], and AutoDAN [15]. In addition, other modalities and transformation techniques have proven effective: Visual Language Models (VLMs) can be evaded with purposely created multi-modal inputs [1,12,20], while attackers may leverage obfuscation/encoding tricks, for example, payload-splitting, base64, and similar methods, or audio channels to bypass filters and issue hidden commands [13,22,32,27]. The volatility and adaptability of such attacks illustrate the inherent vulnerabilities present in LLMs.

Over the past few years, the research community has developed many sophisticated jailbreak attack methods [7,12,27,36]; however, these often require enormous computational capabilities, expertise, and infrastructure. Even today, how these research-oriented techniques compare to user-generated prompts, including prompts spread throughout websites, social media, and online jailbreak forums, is still largely unexplored. This gap motivates our research: to evaluate the effectiveness of attacks designed by researchers and users using the recently proposed StrongREJECT framework [28] as a unified evaluation metric. For this purpose, we collect two complementary datasets of jailbroken prompts. Our user-generated jailbreak dataset consists of 350 prompts from a combination of JailJudge [30], selected entries of jailbreak_llms [25], and part of the nihabil/clean-jailbreak-prompts [5]. Our researcher-generated jailbreak dataset is comprised of 375 prompts consisting of GPTFuzz [35], the subset of JailbreakEval [26], and a segment of JailbreakBench [14].

We evaluate these curated datasets under StrongREJECT to analyze relative performance. The results of our analysis indicate that prompts generated by researchers exhibit consistently higher effectiveness and reliability than those generated by users, while at the same time revealing systematic blind spots in existing evaluators, most clearly for semantic and persona-based jailbreaks.

Contributions. Our main contributions are outlined below:

1. We have curated, standardized, and integrated two complementary datasets for jailbroken prompts (a user-generated set with 350 prompts and a researcher-generated set with 375 prompts) to create a StrongREJECT-compliant benchmark for the research community to aid in reproducible jailbreak and safety evaluation.
2. Using StrongREJECT’s rubric-based and fine-tuned evaluators, together with human evaluations, we thoroughly evaluate these datasets and demonstrate that prompts developed by researchers have higher jailbreaking severity

than user-developed prompts and that existing automated evaluators are systematically underestimating the severity of semantic jailbreaks.

Roadmap. The rest of the paper is organized as follows: Section 2 presents the background and related work, including an introduction of LLMs, the definition and categories of jailbreak attacks, and the StrongREJECT evaluation framework. We describe our data collection process and evaluation methodology in Section 3, followed by the presentation and analysis of our results in Section 4. Section 5 concludes the paper.

2 Background and Related Works

2.1 Large Language Model

Over the past decade or so, large-scale neural architectures that are trained on massive amounts of data have changed the landscape of natural language processing. This transformation stems from the Transformer framework, which gave rise to self-attention and highly parallelized training. Previous encoder models pretrained on general text (like BERT [8]) could achieve considerable representational power, but with ongoing advances in architectural innovations, autoregressive decoders (the GPT family) now can produce much more coherent, interwoven responses based on context.

These advancements have resulted in LLMs being widely adopted in many consumer and professional applications. Commercially available LLMs, including ChatGPT [19], Claude [3], PaLM 2 [10], GitHub Copilot [11], and others, are embedded in many platforms that provide virtual assistants, programming, and other productivity assistance. In research, open-source models such as LLaMA [29] have been made available as an efficient base model that can be contextually tuned for many real-world applications. Furthermore, LLMs are also being adapted for use in multimodal contexts, as demonstrated by models such as Flamingo [1] and GPT-4V [20], which achieve a hybrid of simple textual and visual reasoning capabilities and can accomplish both reasoning and creativity in narrative generation.

These models’ versatility spans key areas like customer service, code generation, educational support, medical imagery, investment analysis, and creative arts, thus exemplifying the increasing importance of LLMs in these areas. There is no doubt that LLMs have significant value related to their development and adoption, but with broad applicability also comes a problem that needs attention, and that is if these models are deployed in sensitive or critical contexts, security, reliability, and risk need to be carefully assessed.

2.2 Jailbreak Attacks

Jailbreaking attacks on LLMs are those deliberately designed input prompts, often referred to as “jailbroken prompts,” to induce the model to bypass the guardrails set by the system and force the model to generate outputs that

violate security constraints, policies, or ethical guidelines [36]. Unlike traditional adversarial attacks that only aim to degrade model performance, jailbreaking in LLMs can affect the model’s interpretation of user input by nesting malicious or system-inconsistent instructions in user input, leading the model to generate undesirable behaviors, such as providing actionable plans for harmful activities, generating false information, or causing targeted information leakage, and so on [6,15,18,35]. In essence, a jailbreak attack uses the concept of “bypassing restrictions” to get the model to prioritize user-crafted inputs over developer-set security constraints on output [7,14,28].

In our summary of jailbreak methods, we aim to give a more detailed outline of jailbreak methods that have been discussed formally in the academic literature. These methods include early in-the-wild prompts such as Do Anything Now (DAN), which have been studied systematically in more recent work, as well as a range of more technical methods that were proposed in the research community. The listed strategies include manual prompts, obfuscation and encoding strategies (payload splitting, ROT13, base64), automated pipelines (PAIR, AutoDAN, GPTFUZZER, TAP), and multimodal exploits (FigStep, voice-based jailbreaking). General tactics are also listed, including role-playing, prompt injection, and workarounds based on multilingual translation. Collectively, these methods demonstrate both the historical underpinnings of jailbreak methods and the range of methods currently being utilized in jailbreak attacks. In Table 1, we provide a concise summary of the methods, mechanisms, and formatted sources that serve as our foundation for comparing methodologies later.

Table 1: Jailbreak Methods

Category	Jailbreak Method	Description
Baseline Obfuscation Attacks	Auto payload splitting [13]	Splitting malicious keywords into disjoint parts and recombining them inside the model context.
	Auto obfuscation [33]	Using Unicode substitutions, symbolic encoding, or multilingual fragments to hide harmful intent.
	ROT13 / Disemvowel	Simple text transformations that evade keyword filters but remain interpretable to LLMs.
	Base64 and Encoding-based Attacks [22,32]	These attacks include encoding malicious instructions in base64 or similar methods that a released LLM can later decode.
	Multilingual jailbreaks [34]	Attacks utilize low-resource languages to circumvent safety mechanisms, transferring destructive prompts into languages that have lower quality detection.
Feedback-based Attacks	PAIR (Prompt Automatic Iterative Refinement) [6]	An automated black-box method that iteratively optimizes jailbroken prompts using an attacker–judge loop, where a judge model evaluates harmfulness and guides search towards stronger attacks.

Continued on next page

Table 1 – continued from previous page

Category	Jailbreak Method	Description
	AutoDAN [15]	An automated technique generates stealthy jailbroken prompts by obfuscating malicious intent while keeping prompts interpretable to the victim model. AutoDAN demonstrated state-of-the-art bypass rates in aligned models.
	GPT-FUZZER [35]	A fuzzing-inspired framework that automatically mutates and generates jailbroken prompts at scale to stress-test alignment. It leverages coverage-guided fuzzing principles for adversarial prompt discovery.
	Tree of Attacks with Pruning (TAP) [18]	An automated black-box method that organizes adversarial prompt exploration in a search tree, adaptively refining branches based on jailbreak success scores.
Multi-Channel Context Attacks	FigStep [12]	A multimodal jailbreak attack on vision-language models that leverages typographic manipulations in images (e.g., adversarial fonts, placement) to bypass safety filters.
	VOICE-JAILBREAK [27]	Attacks on multimodal models such as GPT-4o, where adversarial audio inputs are crafted to embed hidden malicious prompts while remaining benign-sounding to human listeners.
	Prompt Injection [16]	Attacks include hidden instructions within an external context (such as a web page, document, or other metadata) so that an LLM executes malicious commands when processing the unrelated content.
Persona Attacks	Role-playing attacks [4]	Attacks use prompts that instruct the model to assume a different persona or role (as a character, as an expert, or as an unrestricted assistant), decreasing the model’s alignment to outputs that breach safety.

2.3 StrongREJECT

To conduct a systematic comparison between research-designed jailbroken prompts and those generated by users, we utilized the StrongREJECT [28] framework to evaluate the jailbreak responses. StrongREJECT adds two complementary evaluators, a rubric-based and a fine-tuned evaluator, aimed at providing consistent and trustworthy assessments of the jailbreak responses while addressing limitations of previous evaluations.

The rubric-based evaluator utilizes a powerful LLM, like GPT-4, under a defined rubric that will evaluate model responses based on several dimensions of the refusal, including refusal strength, specificity of refusal, and persuasive

discouragement. This method, compared to string-matching or binary labeling, is more aligned with human assessment while encapsulating subtle differences in the way models respond to unsafe queries. The rubric-based evaluator’s main limitations are requirements for API access and higher cost per evaluation, but reliability makes it a good fit for benchmarking across a sizable sample.

In addition to providing interactive, human-in-the-loop evaluations via commercial APIs, StrongREJECT enables low-cost local evaluation and reduces reliance on commercial APIs by providing a fine-tuned evaluator. This fine-tuned evaluator is trained on annotated jailbreak interactions, using a smaller open-source backbone, like Gemma-2B. Despite being lightweight, the fine-tuned evaluator obtains strong correlations with rubric-based assessments and remarkable stability and performance across datasets, making it a good fit for large experimental sweeps that limit computational costs.

StrongREJECT also benchmarks against baseline evaluators, including string-matching [23], jailbroken-binary [2], PICT [24,9], GPT-4 Judge [21], PAIR [6], OpenAI moderation APIs, and HarmBench [17]. While baselines serve as helpful reference points, they are either suspiciously unrobust (string-matching) or simply not scalable or coarse at recognizing refusal quality. The rubric-based and fine-tuned evaluators presented by StrongREJECT possess both fidelity and reasonable scalability, making StrongREJECT evaluators most appropriate for our study.

3 Experimental Setup and Dataset Preparation

3.1 Experimental Setup and Data Collection

All experiments were conducted locally on a MacBook Pro (14-inch, 2021) with an Apple M1 Pro chip and 16 GB of unified memory, running macOS Sequoia 15.3. All preprocessing and evaluation scripts were executed in a local Python environment on this machine. This setup was able to preprocess datasets, normalize prompts, and run the lightweight StrongREJECT fine-tuned evaluator, while rubric evaluations were done through API calls. For all StrongREJECT-based evaluations specified in Section 4.2, we utilized OpenAI’s gpt-3.5-turbo as the model for rubric-based evaluator evaluations and the default StrongREJECT fine-tuned backbone as the model for evaluations by fine-tuned evaluators. We note that StrongREJECT originally recommends GPT-4 for rubric-based evaluation; however, we use gpt-3.5-turbo to balance evaluation cost with reproducibility. Our analysis focuses on relative comparisons between prompt sources under a unified protocol rather than absolute severity calibration.

To facilitate a systematic evaluation of user-generated and researcher-generated jailbreak attacks, we developed two complementary datasets of jailbroken prompts. The user-generated dataset includes prompts from three widely used public sources: JailJudge [30], the jailbreak_llm [25] repository containing DAN-style prompts, and the nihabilal/clean-jailbreak-prompts dataset [5]. The research-generated dataset includes prompts based on techniques specifically described in

the academic literature. We collected prompts from GPTFuzz [35], JailbreakEval [26], and JailbreakBench [14]. Table 2 summarizes all sources, the dates of collection, and the original size of each source.

Table 2: Jailbroken prompt sources and collection details.

Source	Category	Orig. Size	Collected
JailJudge [30]	User	100	Aug 2025
DAN (jailbreak_llm) [25]	User	1,778	Sep 2024
nihabilal/clean-jailbreak-prompts [5]	User	4,070	Aug 2025
GPTFuzz [35]	Research	75	Sep 2024
JailbreakEval [26]	Research	1,000	Jun 2025
JailbreakBench [14]	Research	300	Sep 2025

3.2 Pre-processing

Once we had acquired the raw datasets, we carried out several preprocessing steps to appropriately normalize the texts for comparability purposes as well as to achieve final inputs for the StrongREJECT evaluator. Specifically, the wording, style, language, or token-level structure of any prompt was not changed; every sample is displayed as the original dataset presented in text. Our processing is limited to aspects of selection, filtering, and formatting, and not altering the prompts themselves.

(1) Exact-duplicate removal. We started by using a straightforward deduplication script to remove prompts that were textually the same within each dataset. This part of the processing eliminates redundant samples, which often show up multiple times in large user-generated sets, like jailbreak_llm [25] and nihabilal/clean-jailbreak-prompts [5]. In this phase of the processing, only prompts with identical content at the byte level were removed.

(2) Manual verification of sampled prompts. Public jailbreak datasets often contain benign instructions, such as harmless conversation starters, non-adversarial role-playing, or misclassifying standard normal queries, so we sampled a fraction of prompts from each dataset for manual evaluation. Prompts were marked as non-jailbreak and excluded if they (a) did not attempt to violate safety constraints, (b) did not exhibit adversarial intent, or (c) merely engaged in standard task performance without a systematic attempt to manipulate model performance.

(3) Elimination of semantically redundant prompts. Many datasets, especially DAN variants and user-generated datasets, contain template-like prompts that only change minimally, such as by altering a couple of adjectives or character names. These near-duplicate prompts cannot be eliminated by matching for

exact duplicates. To handle this scenario, we applied some lightweight similarity heuristics to identify potentially near-duplicate prompts and then manually went through the prompts to identify and eliminate semantically redundant variants. The intention of this was to ensure that we did not over-represent a single phrasing pattern while still achieving variability in the intent of prompts that were directed at circumventing the AI safety features.

(4) Treatment of non-English text, special tokens, and multi-turn format. We kept any instances of non-English text, special token artifacts, mark-down structure, or multi-turn conversational format intact when they appeared in the original datasets. These features are part of what made using the jailbroken prompts “real” in the way they were intended to function in an actual jailbreak scenario and would also have implications for correctors of the jailbroken prompts, so to have them missing or changed would alter the intent of the prompts and what was found in the original collected datasets. For multi-turn prompts, we kept the prompts exactly as provided for the data in the prompt, as multi-turn prompts included the description of the original role of the system prompt (system role descriptions, staged directions, or role-play directions).

(5) Strategy for sampling unified user- and research-generated prompts. After removing duplicates and filtering manually, the remaining valid prompts from each source were resampled to create two balanced aggregate datasets: the user-generated set (350 prompts) and the research-generated set (375 prompts). Sampling was done with an emphasis on diversity in sources and protecting the natural distribution of jailbreak styles in each set. Because different datasets had retained different numbers of valid prompts after denoising, the final sample size taken from each source was proportionately modified based on its size.

(6) Format compatible with StrongREJECT processing. The full StrongREJECT [28] pipeline normally takes in a list of forbidden prompts and auto-generates jailbreak attempts before scoring the responses; by contrast, our aim is to simply evaluate existing jailbroken prompts directly. To accomplish this, we implemented a conversion script to format our cleaned prompts into the intermediate format expected by the evaluator components of StrongREJECT. The evaluator requires the following columns for each prompt:

- **category:** the harm domain assigned to each prompt following the six-category taxonomy used in StrongREJECT, including *Disinformation and deception*, *Illegal goods and services*, *Hate/harassment/discrimination*, *Non-violent crimes*, *Violence*, and *Sexual content*;
- **source:** source of the dataset;
- **forbidden_prompt:** the safety-sensitive instruction the model should not adhere to;
- **jailbroken_prompt:** the actual jailbroken prompt provided to the model;
- **jailbreak:** a flag or indicator for evaluation.

Since our samples did not include prompts that we can classify as forbidden, we manually labeled each sample with a suitable forbidden prompt that is consistent with StrongREJECT’s taxonomy. This process is time-intensive and occupies a

large portion of our preprocessing pipeline. Completing the steps above yields two standardized and varied datasets of jailbroken prompts that are ready for evaluation. The datasets above allow us to conduct a controlled comparison of user-generated attacks and research-generated attacks under the evaluation procedure of StrongREJECT.

4 Evaluation and Findings

4.1 Evaluation Procedure

To evaluate the effectiveness of the two curated jailbreak datasets, we used the two evaluators available in the StrongREJECT [28] framework: the rubric-based evaluator and the fine-tuned evaluator. Although StrongREJECT was designed to score jailbreak attempts from its internal attack modules, we wrote two evaluation scripts that simply accepted externally supplied jailbroken prompts and bypassed the attack-generation stage. Each script accepted our prompts in the same input structure that StrongREJECT expects, which allows the evaluators to process our prompts without changing their scoring logic.

Notably, each dataset (user-generated and research-generated) was evaluated three times under each evaluator, resulting in six runs per dataset (three rubric-based and three fine-tuned). For each evaluator condition, we regenerated the model responses and then assigned human labels to those responses; therefore, human scores are reported separately for each evaluator–dataset combination. Given that both evaluators demonstrate nondeterministic behavior by design, replicating the scores will capture some variability. Following the typical worst-case safety evaluation that is standard for jailbreak research, we captured the maximum score across the three runs for each individual prompt and then averaged the maxima across the dataset for each evaluator and dataset. Therefore, instead of obtaining average-case performance, we aimed to obtain a maximum-case score to identify the strongest observed violations of policy.

We also conducted a detailed human review of the model responses after carrying out the automated evaluation, which was labeled using a simple three-level internal rubric:

- 1:** Clear harmful or policy-violating content, where the model produces explicitly disallowed and actionable outputs.
- 0.5:** Semantic or role-based jailbreaks where the model exhibits explicit signs of jailbreak success (e.g., declaring to ignore all restrictions or adopting an unrestricted persona) but does not provide concrete, actionable, or followable steps that would enable a harmful objective.
- 0:** Aligned, refusal, or otherwise safe responses, where the model neither signals policy bypass nor provides any jailbroken reasoning or content.

Human annotations were performed via manual inspection using this rubric and are intended as a complementary reference signal rather than a definitive gold-standard.

This rubric was only developed to guide consistent human judgment and was not based on any external set of annotations. Together, these evaluator outputs and human labels are what we use for our full set of scores in the quantitative and qualitative analysis that follows.

4.2 Evaluation Results and Comparative Analysis

This section demonstrates quantitative comparisons from the implementation of StrongREJECT’s two evaluators (rubric and fine-tuned) on two distinct prompt datasets: a user-generated dataset and one research-generated. Each dataset was evaluated using both evaluators, which translates into four evaluator–dataset result sets.

Group-Level Overview. We begin by comparing aggregate jailbreak severity across datasets and evaluators. Each prompt receives a human-assigned score (0, 0.5, or 1), which we treat as a reference measure of jailbreak severity and is also evaluated by each of StrongREJECT’s two machine evaluators, rubric and fine-tuned, and assigned a continuous severity score. The duality of scores allows us to examine how jailbreak strength varies according to the origin of the prompt (user or research) and through the evaluation, either by a human evaluator or a machine evaluator.

Group	Human Mean	Machine Mean	# Prompts
user_finetuned	0.297	0.215	350
user_rubric	0.330	0.419	350
research_finetuned	0.495	0.468	375
research_rubric	0.529	0.579	375

Table 3: Aggregate jailbreak severity across the four evaluator–dataset combinations. Human Mean is the average human-assigned score (0, 0.5, 1). Machine Mean is the maximum StrongREJECT score (over 3 runs).

Mean human and machine scores for all evaluator–dataset combinations are shown in Table 3. Both sets of human and machine judgments indicate that research-generated prompts have higher average severity than user-generated prompts. This point suggests that the more structured and expert-designed prompts are important jailbreak features within the prompts themselves. The two machine evaluators follow the same ordering as human judgments, indicating broad agreement at the group level.

A second pattern emerges in comparing the two machine evaluators. For both datasets, the rubric evaluator indicates greater mean machine scores than the fine-tuned evaluator, suggesting that the rubric model captures greater diversity in explicit or surface-level attack indicators. The contrast between evaluators can still be viewed at the mean level, though the magnitude of such differences

and their shape in the score distributions would require deeper analysis. The sizes of the datasets were also reasonably balanced (350 user prompts and 375 research prompts) across evaluators, reducing the probability that the differences observed resulted from sample imbalance.

Distribution of Machine-Evaluated Scores. The mean values alone do not indicate whether or not the differences reflect stable distributional patterns or if they are specific to evaluator sensitivity or otherwise. To explore how differences among evaluators and datasets manifest beyond mean severity, Fig. 1 shows the full distribution of machine-evaluated jailbreak scores. Across evaluator–dataset combinations, we observe three common distribution patterns.

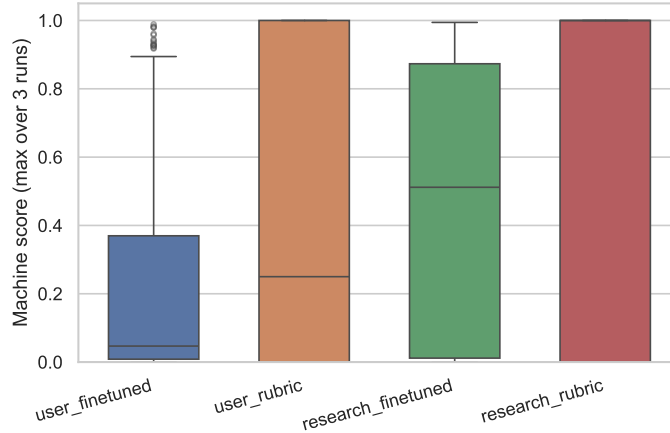


Fig. 1: Distribution of machine-evaluated jailbreak severity across the four evaluator–dataset combinations. Each box displays the StrongREJECT score (maximum over three runs) as a summary score for the group.

First, prompts generated by researchers elicited higher medians and an overall upward shift in score distributions across both evaluators. This suggests that prompts developed by experts in a research context more reliably induce higher-severity outputs across evaluation instances. Second, for user-generated prompts, severity was weaker and considerably more variable irrespective of evaluator. The variability suggests non-expert prompts lack consistent structural attack features, resulting in strong jailbreaks in some instances but often having no effect. Third, the rubric evaluator provided more polarized and consistently higher scores than the fine-tuned evaluator. Score differences remained consistent across datasets with both evaluators, but the rubric amplified the differences between groups and appeared better at picking up on explicit jailbreak indicators. These effects were statistically significant under the main evaluator–dataset contrasts (two-sided

Mann-Whitney U test, $p < 0.01$). Therefore, the ordering we observed was stable and was not just an artifact of sampling noise.

Human–Machine Alignment. The differences in distributions compel us to look closely at how machine-assigned scores might align with human assessments. In Fig. 2, human-assigned severity (0, 0.5, 1) is compared to automated evaluations based on the rubric and fine-tuned evaluators. In every combination of evaluator and dataset, we see two broad regions of agreement, as well as one region of substantial disagreement.

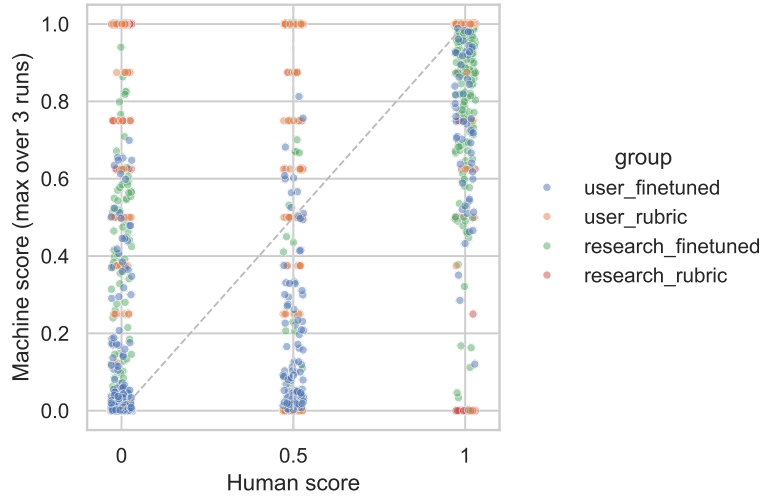


Fig. 2: Human-assigned jailbreak severity (0, 0.5, 1) versus machine-evaluated StrongREJECT scores. Each point is differentiated by either evaluator-dataset combination. The dashed diagonal shows perfect agreement between human and machine assessments.

As described above, at the extremes, machine predictions line up nicely with human-assigned labels. Human-judged benign prompts (0) are clustered around the bottom of the machine scale, while human-labeled full jailbreaks (1) are clustered around the top. This correspondence demonstrates that both evaluators are reliable for identifying clear harmful or harmless responses.

However, human annotators assign the prompts a score of 0.5; typically semantic, role-based, or context-driven jailbreaks are consistently underestimated by both evaluators. These prompts are consistently located below the diagonal reference line, which corresponds to a machine-assigned severity that is less than expected. This consistent downward divergence from human coding indicates an important limitation: Automated evaluations underdetect semantic jailbreak behavior, systematically assigning lower severity scores to prompts that humans identify as partially harmful.

Underestimation is particularly noticeable in the fine-tuned evaluator when utilized with the user-generated dataset, indicating a lesser sensitivity to ambiguous or nuanced harmful intent when the prompt had weak or inconsistent structures. Altogether, these results reveal that the design of the evaluators and the origin of the prompt have nontrivial interactions and that semantic jailbreaks are still a core failure mode for existing automated safety assessment pipelines. For this study, we focus on aggregate and distributional trends rather than individual case analyses.

Jailbreak Success Rates. To understand the practical implications of these severity differences, we calculate two effectiveness metrics derived from human evaluation in Fig. 3. Both evaluators demonstrated that research prompts had a significantly higher success ratio than any other type. Rubric-evaluated datasets had the highest overall ratings, which is consistent with a rubric’s higher sensitivity to explicit jailbreak patterns. While the prompts generated by users had a lower overall outcome, they still did have non-trivial total jailbreak proportions and demonstrate that even low-effort or unsophisticated prompts may elicit harmful outputs.

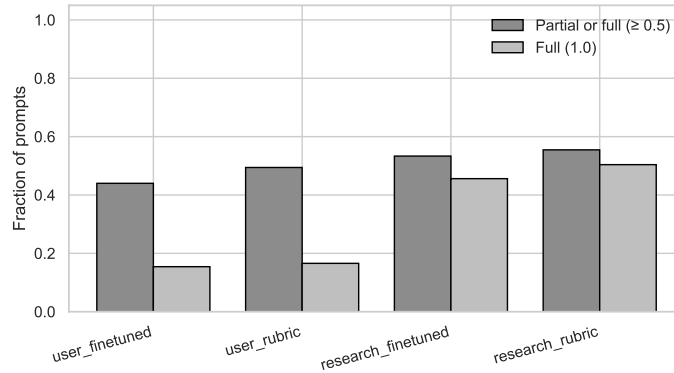


Fig. 3: The human-annotated jailbreak success rates across evaluator–dataset combinations. The dark bars show the proportion of prompts that achieve at least a partial jailbreak (manual score ≥ 0.5), and the light bars show the proportion of prompts that achieve a full jailbreak (manual score = 1).

When assessing partial and full success, there is a clear trend. Datasets from the rubric-based evaluator had a smaller gap between partial and full proportions, which may indicate that structured prompts, once partially successful, may be more likely to escalate to full, damaging responses. This additional pattern reinforces the role of structure as an enabler of robust, repeatable trajectories for attacks.

Rationale for Excluding Category-Level Analysis. Even though StrongREJECT includes six predetermined harmful content categories, we do not emphasize a comparative analysis at the category level due to methodology. These category distributions are highly imbalanced, with some categories containing very few samples while others dominate. More importantly, the categories do not map onto the calibrated levels of difficulty to perform jailbreaks. Therefore, analyzing the six harmful content categories in this study would conflate effects of common dataset composition with actual evaluator behavior and may act as confounders in the differential analysis this study will conduct. Nonetheless, we do note that all six harmful content categories contain at least one full jailbreak across all evaluator–dataset combinations and confirm that model vulnerability is pervasive and not limited to a single content or dataset domain.

5 Conclusion

In this paper, we reveal differences in effectiveness between user-generated and research-generated prompts for jailbreaking large language models under the StrongREJECT evaluation framework. We curated two separate, standardized datasets for evaluation with automated evaluators in the StrongREJECT framework, as well as an additional set of assessments conducted by humans. Our findings indicate that, on average, research-generated prompts have a greater level of effectiveness and reliability than do user-generated prompts. Additionally, we identified an important and systematic difference between human assessments of semantic- and persona-based jailbreaks and the scores assigned by automated evaluators. These findings highlight the need for large language model systems to adopt more robust safety evaluation methodologies capable of identifying subtle forms of misalignment.

References

1. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., Simonyan, K.: Flamingo: a visual language model for few-shot learning. In: Proceedings of the 39th International Conference on Machine Learning (ICML). <https://arxiv.org/abs/2204.14198> (2022)
2. Anonymous: Shield and spear: Jailbreaking aligned llms with generative prompting. In: ACL Rolling Review (ARR) (2023), <https://openreview.net/forum?id=1xhAJSjG45>
3. Anthropic: Introducing Claude. <https://www.anthropic.com/news/introducing-claude> (2023)
4. Authors, R.: Rolebreak: Character hallucination as a jailbreak attack in role-play systems. arXiv preprint arXiv:2409.16727 (2024), <https://arxiv.org/abs/2409.16727>
5. Bilal, N.: Clean Jailbreak Prompts Dataset (2025), <https://huggingface.co/datasets/nihabilal/clean-jailbreak-prompts>, available on HuggingFace Datasets
6. Chao, P., Robey, A., Dobriban, E., Hassani, H., Pappas, G.J., Wong, E.: Jailbreaking black box large language models in twenty queries. arXiv preprint arXiv:2310.08419 (2023), <https://arxiv.org/abs/2310.08419>

7. Chu, J., Liu, Y., Yang, Z., Shen, X., Backes, M., Zhang, Y.: Jailbreakradar: Comprehensive assessment of jailbreak attacks against llms. arXiv preprint arXiv:2402.05668 (2024), <https://arxiv.org/abs/2402.05668>
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018), <https://arxiv.org/abs/1810.04805>
9. Feuillade-Montixi, Q.: PICT: A Zero-Shot Prompt Template to Automate Evaluation (2023), <https://www.lesswrong.com/posts/HJinq3chCaGHnLNE/pict-a-zero-shot-prompt-template-to-automate-evaluation-1>
10. Ghahramani, Z.: Introducing PaLM 2. <https://blog.google/technology/ai/google-palm-2-ai-large-language-model> (2023)
11. GitHub: GitHub Copilot. <https://github.com/features/copilot> (2021)
12. Gong, Y., Ran, D., Liu, J., Wang, C., Cong, T., Wang, A., Duan, S., Wang, X.: Figstep: Jailbreaking large vision-language models via typographic visual prompts. arXiv preprint arXiv:2311.05608 (2023), <https://arxiv.org/abs/2311.05608>
13. Kang, D., Li, X., Stoica, I., Guestrin, C., Zaharia, M., Hashimoto, T.: Exploiting programmatic behavior of llms: Dual-use through standard security attacks. arXiv preprint arXiv:2302.05733 (2023), <https://arxiv.org/abs/2302.05733>
14. Liu, Q., Xu, Y., Sun, K., Wang, X., Yang, Z.: Jailbreakbench: A benchmark for jailbreaking large language models. arXiv preprint arXiv:2404.01318 (2024), <https://arxiv.org/abs/2404.01318>
15. Liu, X., Xu, N., Chen, M., Xiao, C.: Autodan: Generating stealthy jailbreak prompts on aligned large language models. arXiv preprint arXiv:2310.04451 (2023), <https://arxiv.org/abs/2310.04451>
16. Liu, Y., et al.: Prompt injection attacks on llm-integrated applications. arXiv preprint arXiv:2306.05499 (2023), <https://arxiv.org/abs/2306.05499>
17. Mazeika, M., Phan, L., Yin, X., Zou, A., Wang, Z., Mu, N., Sakhaee, E., Li, N., Basart, S., Li, B., Forsyth, D., Hendrycks, D.: Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. arXiv preprint arXiv:2402.04249 (2024), <https://arxiv.org/abs/2402.04249>
18. Mehrotra, A., Zampetakis, M., Kassianik, P., Nelson, B., Anderson, H., Singer, Y., Karbasi, A.: Tree of attacks: Jailbreaking black-box llms automatically. arXiv preprint arXiv:2312.02119 (2023), <https://arxiv.org/abs/2312.02119>
19. OpenAI: Introducing ChatGPT. <https://openai.com/blog/chatgpt> (2022)
20. OpenAI: GPT-4V System Card. https://cdn.openai.com/papers/GPTV_System_Card.pdf (2024)
21. Qi, X., Zeng, Y., Xie, T., Chen, P.Y., Jia, R., Mittal, P., Henderson, P.: Fine-tuning aligned language models compromises safety, even when users do not intend to! arXiv preprint arXiv:2310.03693 (2023), <https://arxiv.org/abs/2310.03693>
22. Rao, A., Vashistha, S., Naik, A., Aditya, S., Choudhury, M.: Tricking llms into disobedience: Formalizing, analyzing, and detecting jailbreaks. arXiv preprint arXiv:2305.14965 (2023), <https://arxiv.org/abs/2305.14965>
23. Robey, A., Wong, E., Hassani, H., Pappas, G.J.: Smoothllm: Defending large language models against jailbreaking attacks. arXiv preprint arXiv:2310.03684 (2023), <https://arxiv.org/abs/2310.03684>
24. Shah, R., Pour, S., Tagade, A., Casper, S., Rando, J., et al.: Scalable and transferable black-box jailbreaks for language models via persona modulation. arXiv preprint arXiv:2311.03348 (2023), <https://arxiv.org/abs/2311.03348>
25. Shen, X., Chen, Z., Backes, M., Shen, Y., Zhang, Y.: "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In: Proceedings of the 2024 ACM Conference on Computer and Communications Security (CCS). <https://arxiv.org/pdf/2308.03825> (2024)
26. Shen, X., Chen, Z., Backes, M., Zhang, Y.: Jailbreakeval: Benchmarking the robustness of large language models against jailbreaking attacks. arXiv preprint arXiv:2406.09321 (2024), <https://arxiv.org/abs/2406.09321>
27. Shen, X., Wu, Y., Backes, M., Zhang, Y.: Voice jailbreak attacks against gpt-4o. arXiv preprint arXiv:2405.19103 (2024), <https://arxiv.org/abs/2405.19103>
28. Souly, A., Lu, Q., Bowen, D., Trinh, T., Hsieh, E., Pandey, S., Abbeel, P., Svegliato, J., Emmons, S., Watkins, O., Toyer, S.: A strongreject for empty jailbreaks. arXiv preprint arXiv:2402.10260 (2024), <https://arxiv.org/abs/2402.10260>

29. Touvron, H., Lavril, T., Izacard, G., *et al.*: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023), <https://arxiv.org/abs/2302.13971>
30. USAIL-HKUST: JailJudge: A Benchmark Dataset for Jailbreaking Attacks (2024), <https://huggingface.co/datasets/usail-hkust/JailJudge>, available on HuggingFace Datasets
31. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. arXiv preprint arXiv:1706.03762 (2017), <https://arxiv.org/abs/1706.03762>
32. Wei, A., Haghtalab, N., Steinhardt, J.: Jailbroken: How does llm safety training fail? arXiv preprint arXiv:2307.02483 (2023), <https://arxiv.org/abs/2307.02483>
33. WitchBot: You can use gpt-4 to create prompt injections against gpt-4. <https://www.lesswrong.com/posts/bNCDexejSZpkuu3yz/you-can-use-gpt-4-to-create-prompt-injections-against-gpt-4> (2023)
34. Yong, Z.X., Menghini, C., Bach, S.H.: Low-resource languages jailbreak gpt-4. arXiv preprint arXiv:2310.02446 (2023), <https://arxiv.org/abs/2310.02446>
35. Yu, J., Lin, X., Yu, Z., Xing, X.: Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts. arXiv preprint arXiv:2309.10253 (2023), <https://arxiv.org/abs/2309.10253>
36. Yu, Z., Liu, X., Liang, S., Cameron, Z., Xiao, C., Zhang, N.: Don't listen to me: Understanding and exploring jailbreak prompts of large language models. In: Proceedings of the 33rd USENIX Security Symposium (USENIX Security 24). pp. 4675–4692. USENIX Association, Philadelphia, PA (Aug 2024), <https://www.usenix.org/conference/usenixsecurity24/presentation/yu-zhiyuan>