General Response

For the May ARR resubmission, we were desk-rejected. We provide the revision for the February ARR's reviews(https://openreview.net/forum?id=BmAJzOVgJ0). We sincerely thank all the reviewers for their thoughtful and constructive feedback on our previous submission. In response, we have carefully revised the manuscript and addressed the concerns raised. The key improvements made are as follows:

1. SOTA jailbreak baseline attacks and more recent target models are added: we add 3 most recent jailbreak attacks: DeepInception, CodeChameleon, ReNeLLM, as well as 2 most recent models Llama3.1-8B-Instruct, Qwen2.5-7B-Instruct. We replace Mistral-7B-V0.2, Vicuna-V1.3, and GPT-4o-2024-0513with their most recent versions: Mistral-7B-V0.3, Vicuna-7B-V1.5, and GPT-4o-2024-1120 for a more comprehensive evaluation. See details in Section 3.3 and Section 4.
2. We curated a more comprehensive and diverse evaluation dataset based on 5 widely recognized benchmark datasets: JailbreakBench, HarmBench, MaliciousInstruct, AdvBench, and TruthfulQA. We combined them and removed the semantically duplicate harmful queries and formed a curated dataset containing 417 harmful queries for jailbreak attack evaluation. We believe this curated dataset can lead to a reliable evaluation for both jailbreak attacks and the content safety filter's performance. See details in Section 3.3
3. For the prior literature works, we add some prior literature on the related work discussion.
4. We addressed the typos and revised the paper.

We re-run all the experiments on the newly curated dataset to ensure a reliable evaluation.

Reviewer 1kRd

1.For the weakness: This work points out the need to increase the detection accuracy of LLM safety filters while preserving usability. However, this is nothing new since several prior works:

While prior works such as have acknowledged the general utility-safety trade-off in LLMs, our work goes significantly beyond this known observation by presenting the first comprehensive investigation of the adversarial (cat-and-mouse) dynamic between jailbreaking attacks and LLM safety filters in a full deployment pipeline. Unlike earlier studies that predominantly focus on model alignment or isolated harmful output generation, we shift the focus to the real-world battlefront, where adversaries attempt to bypass not just the model's inherent safeguards but the entire safety infrastructure, including post-hoc content moderation filters. This is a critical and underexplored dimension of LLM safety.
Moreover, our findings challenge a key implicit assumption in the literature: that *high jailbreak success rates reflect real-world vulnerability*. We show that most jailbreaks are indeed detectable by content filters, revealing that prior evaluations may have *overestimated the effectiveness* of such attacks in practice. However, we also uncover a key weakness that current safety filters struggle with balancing high recall and precision, creating exploitable gaps that undermine robustness in production settings.
In summary, our work offers a novel and essential perspective on LLM safety, advancing beyond the well-known utility-safety trade-off to illuminate the ongoing arms race between increasingly sophisticated jailbreaks and evolving safety defenses—a dynamic that, to our knowledge, has not been systematically addressed in prior literature.

Reviewer HXzz

For Weakness 1: The authors claim that "nearly all evaluated jailbreak can be detected..."... if we remove PromptGuard ... get a very different picture. Moreover, ...O1 might constitute an increased attack surface:

Even without PromptGuard, most pass rates remain below 0.05. While "can be detected" may not imply 100% detection, we will clarified this by exact quantitative results in the appendix to ensure rigor. We appreciate the point about attack surface. However, in practical scenarios, filters like O1/O3 are typically deployed as part of a black-box pipeline, and their reasoning tokens are not exposed.(We replace O1 with more recent O3) in the new revision. We currently do not see evidence that this increases the attack surface. If we incorporate an O3 filter

into the inference pipeline, we believe it will significantly increase the attack burden for the adversary, as they would need to successfully bypass both the target model and the safety filters.

For Weakness 2. Descriptions of employed attacks are in the Appendix ...not self-contained. We respectfully disagree that placing attack descriptions in the Appendix makes the paper not self-contained. The Appendix is meant to provide supporting details that complement the main content without disrupting the core narrative.Besides, we have included the Jailbreak attacks' description in the Related Work Section-Section 2..

For weakness 3. Despite the authors categorize jailbreak attacks as optimization-based, LLM-assisted, and heuristic-based in Section ... relevant attacks not considered nor referenced

We thank the reviewer for the suggestion and have included additional results, none of which contradict our findings so far.

For weakness 4. For optimization-based attack, it is not clear whether safety filters were applied to each iteration or to specific ones...

The safety filters (and the metric computations) were applied only to the final, optimized jailbreak prompts. While filtering all intermediate iterations could offer additional insights, the computational cost is substantial and typically exceeds the resources available in standard research settings.

For weakness 5. The results reported in Table 1 does not seem to reflect the defined pass rate?

Table 1 reports the pass rate (Pass) as defined—the rate at which harmful samples are not detected at either the input or output stage. We note that the detection rates at each stage cannot be directly summed to compute the overall detection rate, since a single sample may be detected at both stages. To make it clear, we have added the formula of the pass rate in this revision in Section 3.3.

For weakness 6. Evaluate recent open models.

We add additional evaluations and observed consistent results as suggested, as stated in **Section 3.3:** Experimental Setup, the results are also updated across the paper.

For comment 1, we rephrased the sentence in the abstract: substituting the original sentence with this "To address this gap, we present the first systematic evaluation of jailbreak attacks targeting LLM safety alignment, assessing their success across the full inference pipeline, including both input and output filtering stages."

For comment 2, after crafting a much larger curated dataset with 417 harmful queries paired with 417 benign queries, we modified all the associated content. Table 1 and Table 2 report performance on all 417 harmful queries.

For the remaining comments: we remove the repeated content, add numbers to the formula, fix numbers on the PromptGuard heatmap, and replace "open-source" with "open-weight."

Reviewer TCfA:

1. As mentioned in sec3.3, the entire evaluation is based on JailbreakBench, but it only contains 100 samples. How can this ensure the reliability of the evaluation results? When new samples with different tasks, domains, or attack methods from JailbreakBench appear, will the evaluation results still be consistent?

As stated in the general response, we have built a more diverse dataset covering 5 existing jailbreak benchmarks and re-evaluated all the experiments based on it. Our key findings remain consistent.

2. Explicitly stating all the valuable findings in the paper in the introduction section can further improve readability.

We thank the suggestion and have revised it in our updated version.