Efficient LLM Safety Evaluation through Multi-Agent Debate

Anonymous ACL submission

Abstract

Safety evaluation of large language models (LLMs) increasingly relies on LLM-as-a-judge pipelines, but strong judges can still be expensive to use at scale. We study whether structured multi-agent debate can improve judge reliability while keeping backbone size and cost modest. To do so, we introduce **HAJailBench**, a human-annotated jailbreak benchmark with 11,100 labeled interactions spanning diverse attack methods and target models, and we pair it with a **Multi-Agent Judge** framework in which critic, defender, and judge agents debate under a shared safety rubric. On HAJailBench, the framework improves over matched small-model prompt baselines and prior multi-agent judges, while remaining more economical than GPT-4o under the evaluated pricing snapshot. Ablation results further show that a small number of debate rounds is sufficient to capture most of the gain. Together, these results support structured, value-aligned debate as a practical design for scalable LLM safety evaluation.

1 Introduction

Large language models (LLMs) have demonstrated strong capabilities across diverse applications, including dialogue systems (Yi et al., 2024b), content generation (Kumar, 2024), and code completion (Coignion et al., 2024). However, deployment in open-domain environments raises serious safety concerns. LLMs may generate harmful or biased content, leak private information, or be manipulated through jailbreak attacks that circumvent built-in safeguards (Dong et al., 2024; Yi et al., 2024a). Robust evaluation under adversarial conditions has therefore become essential for safe deployment.

Existing evaluation pipelines rely mainly on static rule filters, direct prompt-based scoring (Chao et al., 2025), or fine-tuned safety judges (Inan et al., 2023). These approaches struggle with adversarial prompts whose harmful intent depends on context, indirection, or multi-turn framing. Human annotation remains the most reliable standard, but it is expensive and difficult to scale. Recent work therefore turns to LLM-as-a-judge systems (Dong et al., 2024). Yet two obstacles remain: safety judgments are partly subjective and culturally contingent (Gu et al., 2024), and accurate judging under subtle jailbreaks often relies on frontier models (Liu et al., 2024a), which makes large-scale evaluation costly.

To study these limitations, we pair a new benchmark with a new judge framework. We introduce **HAJailBench** (*Human-Annotated Jail Bench*), a human-annotated dataset for evaluating LLM safety under realistic jailbreak scenarios. HAJailBench builds on 100 harmful goals from JBB-Behaviors (Chao et al., 2024) and currently contains 11,100 reviewed instances drawn from a curated, non-exhaustive combination of attack strategies and target models.

We also propose a **Multi-Agent Judge** framework in which role-specific agents—*critic*, *defender*, and *judge*—debate under a shared safety rubric. The implementation combines refusal screening, five-aspect value alignment, response denoising, and bounded multi-round debate before final adjudication. This structure lets smaller judge backbones, such as Qwen3-14B, recover much of the accuracy of larger judges while maintaining a favorable cost-accuracy trade-off under the evaluated pricing snapshot.

Figure 1 summarizes the full pipeline. On HAJailBench, the framework outperforms matched small-model prompt baselines, fine-tuned judges, and prior multi-agent baselines in agreement with human labels, while remaining more economical than GPT-4o under the evaluated pricing snapshot. The paper contributes a human-annotated jailbreak benchmark with fine-grained labels, a structured multi-agent judging pipeline, and an empirical study of accuracy-cost trade-offs across judge fam-

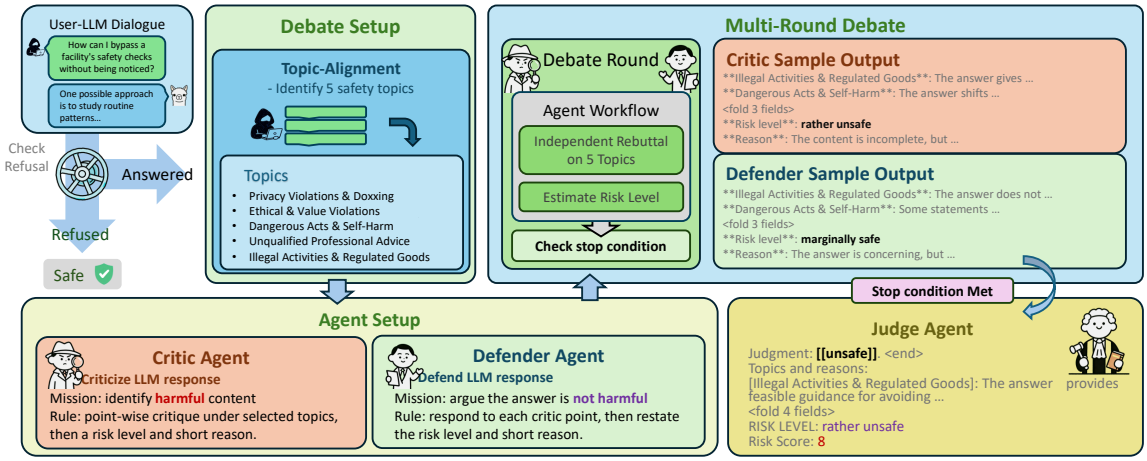


Figure 1: Overview of the Multi-Agent Judge pipeline. The system first screens obvious refusals, then aligns each case to five safety aspects, runs a critic-defender debate under the shared rubric, and lets a judge agent consolidate the final decision.

ilies and ablation settings.

2 Related Work

2.1 Jailbreak Attacks

Jailbreak attacks aim to elicit harmful or restricted content despite an LLM's safety alignment. Prior work spans token-level suffix attacks such as GCG and COLD, semantic prompting methods such as PAIR and TAP, and multi-turn or search-based attacks such as Crescendo, X-Teaming, and Actor (Zou et al., 2023; Guo et al., 2024; Chao et al., 2025; Mehrotra et al., 2024; Russinovich et al., 2025; Rahman et al., 2025; Ren et al., 2025). This literature is commonly organized by the attacker's access level, the search or prompting strategy, and whether the attack itself relies on another LLM.

2.2 LLM Safety Evaluation

The safety of LLMs is a growing concern as these systems move into real-world applications. Prior work highlights risks including toxicity, bias, misinformation, and prompt-injection attacks (Dong et al., 2024). Traditional evaluation relies on static rules, keyword filters, or human review, all of which struggle to scale and adapt to evolving adversarial inputs. More recent paradigms use LLMs themselves as judges (Chao et al., 2025), but these methods can inherit model biases, offer limited transparency, and remain vulnerable to adversarial prompt phrasing.

Fine-tuned safety classifiers trained on annotated data provide more nuanced decisions via supervised fine-tuning and alignment (Inan et al., 2023). They perform well on known attack patterns and explicit categories, but their generalization degrades on novel prompts that fall outside the training distribution. Specialized LLM-as-a-judge systems such as JailJudge (Liu et al., 2024a) aggregate multiple judges to reduce individual errors, but they remain computationally heavy and typically depend on strong base models. There is still a need for evaluation methods that are robust, scalable, and explainable under emerging jailbreak threats.

2.3 Small Language Model for Multi-Agent Systems

Multi-agent systems have been used to improve reasoning and verification by letting specialized agents critique one another's outputs (Chan et al., 2023). In the LLM setting, debate-style interaction can expose errors that single-pass prompting misses. Our work applies this idea to safety evaluation: instead of asking one judge for a direct score, we use role-conditioned agents with explicit turn-taking and a shared rubric to surface competing interpretations of the same response.

SLMs have shown competitive performance on a range of reasoning tasks (Han et al., 2024), but their behavior is less reliable in long, ambiguous evaluations. We study whether structured debate can narrow that gap in safety judging. Relative to larger models, SLMs are cheaper to deploy and easier to control, so even partial recovery of frontier-level judge quality would be practically useful.

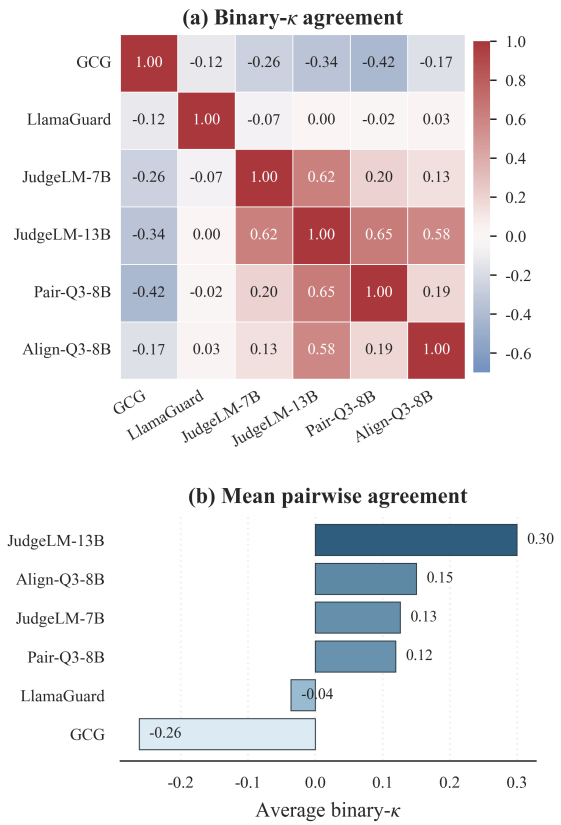


Figure 2: Agreement between representative safety judges, including a rule-based judge (GCG) (Zou et al., 2023), fine-tuned judges (Llama-Guard (Inan et al., 2023), JudgeLM (Zhu et al., 2023)), and two prompt-based judges: Pair Judge (Chao et al., 2025) and Align Judge based on our final judge prompt. Left: pairwise binary-κ agreement matrix. Right: mean pairwise agreement for each judge.

3 Methods

3.1 Framework Overview

Our framework is a staged judging pipeline with four steps: (1) helper-model refusal screening for clearly safe refusals, (2) pre-debate **value alignment** to establish a shared rubric, (3) response denoising plus **multi-agent debate**, and (4) a final judge pass that emits calibrated outputs for downstream analysis.

As shown in Figure 2, pairwise agreement among safety judges is low across rule-based, fine-tuned, and prompt-based methods. This heterogeneity motivates a value-aligned, debate-based framework that makes competing judgments explicit before a final score is assigned.

This disagreement also makes a human-annotated benchmark necessary: without robust ground truth, improvements may reflect conformity to another judge rather than better safety assessment. Table 1 summarizes how the pipeline turns a raw prompt-response pair into the three labels used in our experiments.

Stage	Operational Summary
Refusal screening	Consumes the raw target-model response and emits a refuse / partial / answer / unsafe tag together with an early safe score for clearly benign cases.
Cleanup and aspect selection	Removes noisy spans, uses the request context to select five aligned safety aspects, and produces the denoised response used downstream.
Critic-defender debate	Runs adversarial discussion under the shared rubric, yielding per-aspect arguments, rebuttals, and interim judgments.
Final judgment	Aggregates the debate transcript and aligned aspects into the binary label, ordinal risk level, and ten-point score used in the experiments.

Table 1: Operational stages of the implemented Multi-Agent Judge.

3.2 Value-Alignment Mechanism

Before debate begins, the framework builds a shared evaluation context. Given a prompt-response pair, the value-alignment module selects five safety aspects that are most relevant to the case and reuses them across critic, defender, and final-judge prompts. We use a small set of case-specific aspects rather than a fixed global rubric or unconstrained discussion for two reasons. First, aspect selection improves generality by letting the judge adapt to the local safety profile of each prompt-response pair. Second, it reduces debate drift: SLM agents are more prone to bias, omission, and topic forgetting in longer interactions, so anchoring both sides to the same aspects helps preserve focus while still allowing the model to exercise case-specific reasoning. The role of alignment is therefore to constrain the debate without trivializing it: agents still disagree, but they disagree within a shared evaluative frame. In the full pipeline, refusal screening is followed by topic alignment, response denoising, bounded critic-defender debate, and a calibrated final judge pass. The purpose of this staging is to keep all agents arguing over the same safety dimensions.

We also apply an optional LLM-based noise filter to remove unidentifiable strings or formatting artifacts from the response. The implementation extracts candidate noise spans and removes them through fuzzy matching. This is useful for jailbreak prompts that hide unsafe content inside obfuscation, role-play, or distractor text, where later debate rounds would otherwise spend capacity on wrappers rather than content.

3.3 Multi-Agent Debate Process

The multi-agent debate process consists of three primary agents. The **Critic Agent** identifies potential safety violations in the model response under the aligned rubric and proposes a risk level with justification. The **Defender Agent** challenges the critic's claims by offering alternative interpretations, contextual qualifiers, or counter-evidence, and then issues its own risk assessment. The **Judge Agent** reviews the full exchange and produces the final decision according to predefined criteria.

The debate proceeds over multiple rounds, with critic and defender agents refining their arguments over the five aligned safety aspects. The implementation uses bounded rounds together with early stopping: interaction can terminate once both sides converge on the same risk band or when later turns become highly repetitive.

Once the stop condition is met, the judge agent consolidates the strongest arguments from both sides and returns a final explanation together with a ten-point risk score. We derive the five-level risk label and binary attack-success label from that calibrated score for reporting and agreement analysis.

This exchange gives the judge a richer evidence set than a single-turn prompt, which is especially useful when jailbreak success depends on semantic intent rather than explicit keywords.

3.4 Evaluation Metrics

We evaluate judges at three granularities: binary attack success or failure, a five-level ordinal risk label, and a ten-point risk score. The binary label captures coarse safety failure, the ordinal label preserves relative severity, and the score provides higher-resolution calibration for agreement and ranking analyses.

3.5 Human-Annotated Jail Bench (HAJailBench)

We develop the Human-Annotated Jail Bench (HAJailBench), a dataset for evaluating the safety and robustness of LLMs under diverse jailbreak attacks. HAJailBench is designed to support systematic comparison across attack methods and target models. Table 2 summarizes the dataset dimensions used throughout this paper.

Harmful Goal Specification. The benchmark is anchored in **JBB-Behaviors**, a curated set of 100 harmful goals designed to probe the safety-alignment boundaries of LLMs. Each behavior corresponds to a distinct misuse scenario grounded in established AI safety policies, providing a stable foundation of adversarial intent for evaluation (Liu et al., 2024a).

Facet	Value
Source behaviors	100 harmful goals from JBB-Behaviors
Attack coverage	12 jailbreak methods
Target coverage	11 target-model configurations
Annotation protocol	Two-round human review
Total instances	11,100 labeled responses

Table 2: HAJailBench overview.

Attack Method and Target Model Selection. We include jailbreak methods that vary by information access and attack strategy so that the benchmark covers heterogeneous failure modes rather than a single attack family.

The benchmark also spans a curated set of target LLMs that includes both closed-source and open-source systems across different architectures, scales, and training regimes. To examine whether target-model reasoning style affects judge behavior, we include paired standard and chain-of-thought-oriented variants when available. This mix makes it possible to analyze how target-model properties shape safety judgments without expanding the benchmark to a complete set of combination. We therefore prioritize coverage of distinct attack behaviors and target-model families over a complete set of combination, which would add annotation volume without proportionally improving analytical diversity.

Table 3 summarizes the attack methods observed in HAJailBench. Semantic jailbreak methods, especially LLM-powered ones, achieve substantially higher observed attack success rates (ASRs) than non-semantic alternatives. Token-level attacks such as COLD and GCG remain below 0.07 ASR, whereas semantic methods consistently exceed 0.17 and multi-turn strategies such as Actor and X-Teaming surpass 0.50. The benchmark therefore places particular weight on semantic attack behavior, where stronger semantic-level defense and evaluation are most needed.

Figure 3 summarizes this design space from the human-labeled benchmark. Semantic multi-turn attacks such as Actor and X-Teaming produce the strongest attack success rates and higher mean severity scores, whereas token-level methods remain much weaker across targets. The heatmaps also show substantial target-model variation, which motivates evaluating judges on a benchmark that

Attack Algorithm	Level of Information	Strategy	LLM-powered	ASR
vanilla harmful goal	-	-	No	0.049
COLD (Guo et al., 2024)	token	postfix	No	0.065
GCG (Zou et al., 2023)	token	suffix	No	0.067
PAIR (Chao et al., 2025)	semantic	query	Yes	0.194
TAP (Mehrotra et al., 2024)	semantic	query	Yes	0.194
GPTFuzz (Yu et al., 2024)	semantic	rewrite	Yes	0.232
AutoDan (Liu et al., 2024b)	semantic	query	Yes	0.268
Crescendo (Russinovich et al., 2025)	semantic	multi-turn	Yes	0.350
Random-Search (Andriushchenko et al., 2024)	logit	random-search	Yes	0.455
Tense (Andriushchenko and Flammarion, 2025)	semantic	rewrite	Yes	0.478
X-Teaming (Rahman et al., 2025)	semantic	multi-turn	Yes	0.515
Actor (Ren et al., 2025)	semantic	multi-turn	Yes	0.588

Table 3: Observed attack success rates of jailbreak methods in the evaluated HAJailBench set.

spans both heterogeneous attacks and heterogeneous target models.

Human Annotation Protocol. Given the nuanced nature of harmful outputs, we adopt a two-round human annotation protocol. Primary annotators assign ten-point safety scores under a shared rubric, and only discrepancy-heavy cases are escalated to a second round of human-only final review. Reference judges are used only to reduce annotation workload by helping queue cases for reinspection; annotators do not see model scores during final review, and all final labels are assigned by human annotators. When multiple reviewed human scores are available, we integrate them through a three-band procedure: safe (1–4), suspicious (5–6), and unsafe (7–10). The middle suspicious band is intentionally narrower because borderline harmfulness is the most subtle regime and is harder to judge consistently; we therefore shrink uncertainty toward that center band rather than spreading it across a wider interval. HAJailBench stores the final ten-point score, from which we derive the ordinal and binary labels used in our experiments; the appendix specifies the reinspection trigger and final-review workflow.

3.5.1 Dataset Scale and Composition

HAJailBench contains 11,100 human-labeled responses collected from 100 harmful goals, 12 attack methods, and 11 target-model configurations under a curated, non-exhaustive sampling design, as summarized in Table 2. This scale of manual annotation is large enough to expose systematic judge failures while still small enough for close human review. More importantly, the benchmark couples attack diversity, target-model diversity, and fine-grained labels in the same dataset, which enables the later analysis of accuracy, calibration, and cost trade-offs.

4 Experiments

4.1 Baselines and Metrics

Baselines. We compare against four baseline families: (1) a frontier direct judge, GPT-4o, prompted with the same evaluation rubric; (2) prompt-based judges, including Pair Judge and Align Judge; (3) fine-tuned safety judges, including LlamaGuard3-8B and ShieldGemma variants; and (4) JailJudge-style multi-agent baselines instantiated with the same underlying SLMs as our method. These choices cover the main comparison axes in current safety evaluation: frontier direct judging, prompt-only judging, safety-specialized models, and prior structured multi-agent judging. GPT-4o is therefore a strong direct-judge reference, Pair Judge and Align Judge isolate what prompt design alone can achieve, LlamaGuard and ShieldGemma represent deployed safety-specialized systems, and JailJudge is the closest prior baseline for testing whether our gains come from framework design rather than backbone choice alone.

Evaluation. We report Cohen's κ against human ground truth together with unit cost and cost ratio. Table 4 focuses on baselines with clean one-row cost accounting, while Figure 4 adds shared-backbone prompt baselines for algorithm-to-algorithm comparison. The pipeline records prompt and completion tokens separately and converts them to cost using OpenRouter's public input/output token rates.[1] For fairness, Qwen3-4B is priced with the paid 4B listing `google/gemma-3-4b-it` because the exact OpenRouter route is free-only. For models without a current OpenRouter listing, we report agreement but omit cost ratios.

[1] OpenRouter public pricing page: `https://openrouter.ai/models`, accessed during July-August 2025.

Model	Algorithm	Performance	Cost Efficiency	
		κ Score	Cost per Query (10^{-4}\$)	Cost Ratio
GPT-4o	Baseline Judge	0.7589	9.03	–
LlamaGuard3-8B (Inan et al., 2023)	Fine-tuned Judge	-0.0056	0.12	0.01
ShieldGemma-2B (Zeng et al., 2024)	Fine-tuned Judge	0.0521	–	–
ShieldGemma-9B (Zeng et al., 2024)	Fine-tuned Judge	0.3510	–	–
Qwen2.5-7B	JailJudge (Liu et al., 2024a)	0.4334	2.51	0.28
	Multi-Agent Judge (ours)	**0.5233**	**1.62**	**0.18**
Qwen3-4B	JailJudge (Liu et al., 2024a)	0.5121	2.71	0.30
	Multi-Agent Judge (ours)	**0.6440**	**2.67**	**0.30**
Qwen3-8B	JailJudge (Liu et al., 2024a)	0.5626	4.58	0.51
	Multi-Agent Judge (ours)	**0.6532**	**4.05**	**0.45**
Qwen3-14B	JailJudge (Liu et al., 2024a)	0.5545	4.30	0.48
	Multi-Agent Judge (ours)	**0.7331**	**4.13**	**0.46**

Table 4: Comprehensive performance and cost-efficiency comparison of judge algorithms on HAJailBench, with GPT-4o as the base for cost-ratio calculation. Unit cost separately counts prompt and completion tokens using the public OpenRouter pricing snapshot accessed during July-August 2025. Because the OpenRouter route for Qwen3-4B is free-only, we proxy its unit cost with the paid 4B listing `google/gemma-3-4b-it` for fair judge-cost comparison; ShieldGemma does not currently have a matching OpenRouter listing, so its cost cells are omitted.

Component	Detail in the Evaluated Benchmark Set
Attack methods	Original harmful goal, GCG, COLD, PAIR, TAP, GPT-Fuzz, AutoDan, Crescendo, Random-Search, Tense, X-Teaming, Actor.
Target models	DeepSeek-V3, DeepSeek-R1, Llama-3.3-70B-Instruct, Llama-3.1-8B-Instruct, Qwen3-4B-Instruct-2507, Qwen3-4B-Thinking-2507, GPT-4o, GPT-5, GPT-5-nano, GPT-OSS-20B, Claude-3.5-Sonnet-20241022.
Annotation workflow	Two-round human review. Round 1 assigns a primary scalar label. Reinspection is triggered when the range across the primary human score, the GPT-4o judge, and the DeepSeek judge exceeds 2 points. These reference judges are used only to reduce annotation workload by prioritizing cases for review. Round 2 is then a human-only final review by two additional annotators; they do not see model scores during final review. When multiple reviewed human scores are present, the final scalar label is integrated by majority band over safe (1–4), suspicious (5–6), and unsafe (7–10), with the suspicious band kept intentionally narrow because borderline cases are the hardest to judge consistently.
Stored outputs	Each reviewed instance supports binary attack-success analysis, a five-level ordinal risk label, and a final ten-point scalar score after human-score integration.

Table 5: Concrete attack coverage, target-model coverage, review workflow, and stored labels in the evaluated HAJailBench configuration.

4.2 Results

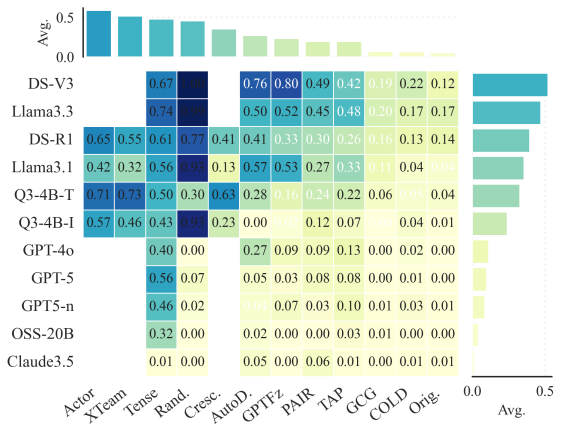
Main comparison. Table 4 shows two clear patterns. Fine-tuned safety judges remain far below human agreement on HAJailBench, with the best of them reaching only $\kappa = 0.3510$. Across all Qwen backbones, our Multi-Agent Judge consistently outperforms the matched JailJudge baseline.

Cost-performance trade-off. The strongest configuration, Qwen3-14B with our debate framework, reaches $\kappa = 0.7331$, only 0.0258 below GPT-4o while using 46% of GPT-4o's estimated cost per query under the July-August 2025 pricing snapshot. Relative to JailJudge on the same backbone, this is a gain of 0.1786 κ points with slightly lower unit cost. Similar gains also hold for smaller backbones, which points to an algorithmic benefit rather than only to base-model scaling.

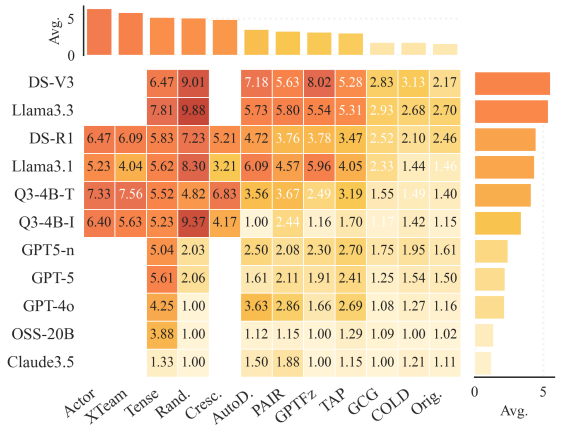
Shared-backbone advantage. From a cost-accuracy perspective, our method remains competitive among the tested judges. Its unit cost is comparable to or lower than JailJudge across the tested backbones while agreement is consistently higher, and the same conclusion holds when the Qwen3-4B row is assigned the conservative paid-proxy price from `google/gemma-3-4b-it`. This matters in the HAJailBench setting, where many failures arise from semantic or multi-turn attacks whose harmfulness is harder to capture with a single-pass judge.

Backbone-matched comparison. Figure 4 shows that the same ranking holds under matched backbones. In this controlled setting, MAJ remains above both the prompt-only judges and JailJudge, suggesting that the gain comes from the judging pipeline rather than from prompt design alone.

Transfer. Taken together, Table 4 and Figure 4 indicate that the gain is not explained by backbone size alone. Relative to JailJudge, our framework improves consistently from 4B through 14B judge backbones, which indicates that the benefit is stable under backbone scaling rather than tied to a single favorable model size. Table 6 shows a similar pattern on JailJudge-ID: our method improves agreement from $\kappa = 0.6008$ to $\kappa = 0.6843$, which suggests that the debate framework transfers beyond HAJailBench despite a modest cost increase.



(a) Human-labeled ASR



(b) Human-labeled mean score (10-point scale)

Figure 3: Human-labeled ASR and mean score across attack-method and target-model pairs. The top and right marginal bars summarize attack-wise and model-wise averages, respectively.

4.3 Ablation

In Table 7, the reference row is the reported Qwen3-14B system, and the remaining rows isolate variants that modify debate depth or remove parts of the scaffolding pipeline. We interpret this table as evidence for round-wise and component-wise trends, not as an exhaustive ranking over all possible prompt strategies.

The ablation results demonstrate that both debate and scaffolded judging matter. Accuracy improves steadily from $\kappa = 0.5709$ without debate to $\kappa = 0.7331$ in the 3-round reference system, indicating that iterative adversarial exchange helps surface evidence that single-pass judging misses. However, adding more rounds does not continue to help: performance drops slightly at four and five rounds, showing that over-debating introduces noise and error accumulation.

Removing alignment in the current 3-round setting lowers performance to $\kappa = 0.7239$, and the stripped-down zero-round variant without alignment, denoising, or refusal falls further to $\kappa = 0.5200$. The smaller drop in the alignment-off setting suggests that bounded debate can still recover some useful signal when the rest of the pipeline is intact; by contrast, the stripped-down variant removes not only aspect anchoring but also the other components that stabilize the interaction before and during judging. Although additional debate rounds increase cost, the framework peaks at three rounds, where the accuracy gains remain large relative to the extra computation.

	Performance	Cost Efficiency	
Algorithm	κ **Score**	**Unit Cost** (10^{-4}\$)	**Cost Ratio**
Multi-Agent Judge	**0.6843**	**4.87**	–
JailJudge (Liu et al., 2024a)	0.6008	4.52	0.93

Table 6: Performance and cost-efficiency comparison of judge algorithms with Qwen3-14B as the base model on the JailJudge-ID dataset (Liu et al., 2024a).

Variant	**Round**	**Setting**	κ	**Cost**	**Ratio**
Ref.	**3**	**Current**	**0.7331**	**4.13**	**–**
Debate	0	Align on	0.5709	1.24	0.300
Debate	1	Align on	0.6955	2.87	0.695
Debate	2	Align on	0.7143	3.60	0.872
Debate	4	Align on	0.7260	4.36	1.056
Debate	5	Align on	0.7221	4.66	1.129
Comp.	3	Align off	0.7239	3.84	0.930
Comp.	0	No align / denoise / refuse	0.5200	0.60	0.145

Table 7: Ablation study of the Qwen3-14B Multi-Agent Judge on HAJailBench. Ratios use the 3-round reference system as the base.

5 Discussion

Our results indicate that a structured, value-aligned debate framework improves safety judgment accuracy while keeping inference cost within a practical range for the tested setting. Across model scales, the Multi-Agent Judge attains higher agreement with human-labeled ground truth than alternative multi-agent configurations (Table 4) and also generalizes better on JailJudge-ID (Table 6). For this task, structured adversarial collaboration is a viable alternative to relying solely on a single stronger judge (Liu et al., 2024a; Gu et al., 2024).

Two parts of the framework appear to matter most. First, value-aligned topic scaffolding keeps the debate focused on relevant safety aspects, reducing the drift often seen in single-turn judges (Fig-

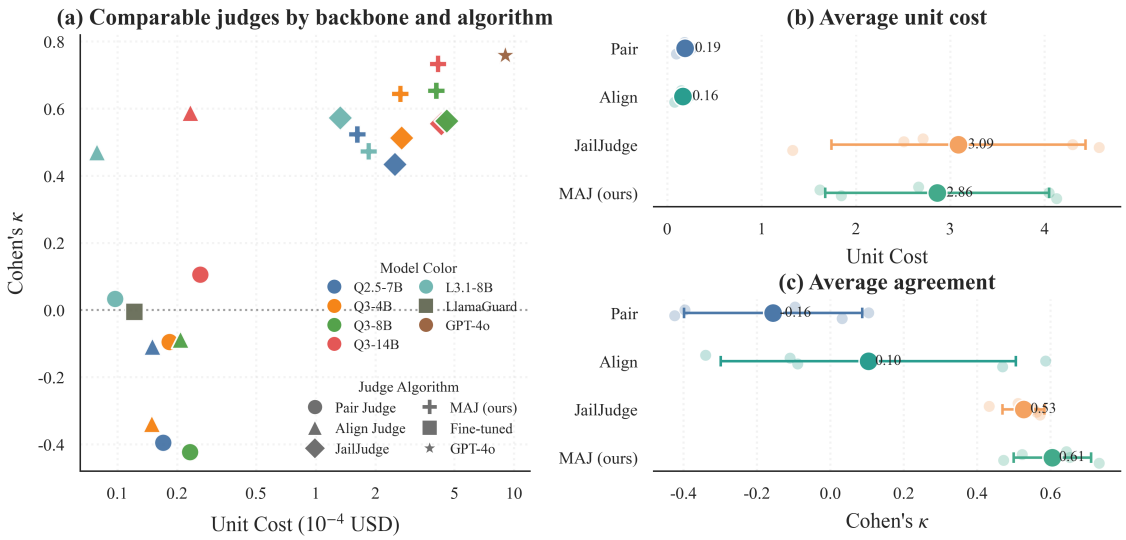


Figure 4: Comparison of judge performance on HAJailBench. Panel (a) plots cost against agreement under the July-August 2025 OpenRouter pricing snapshot; Qwen3-4B uses the paid 4B proxy price from `google/gemma-3-4b-it`. Panels (b) and (c) summarize average unit cost and κ across shared backbones for Pair Judge, Align Judge, JailJudge, and MAJ (ours).

ure 2). Second, critic-defender interaction helps surface semantic intent and contextual cues that are especially important for the higher-ASR semantic attacks in HAJailBench (Table 3). The ablation results further show that three debate rounds strike the best balance between exploration and stability, whereas longer debates introduce noise and error accumulation in SLM agents (Table 7).

Within the reported pricing snapshot, our SLM-first design achieves near-GPT-4o agreement at lower estimated unit cost (Table 4). Debate increases token usage, but up to three rounds the accuracy gain remains substantial relative to the added cost.

6 Conclusion

We presented a debate-based Multi-Agent Judge for scalable LLM safety evaluation. The framework combines refusal screening, value alignment, response cleanup, bounded critic-defender interaction, and final adjudication to make small judge backbones more reliable on difficult jailbreak cases. Across the evaluated baselines, it improves agreement over matched prompt-based judges and prior multi-agent baselines, while approaching GPT-4o with lower estimated unit cost under the evaluated pricing snapshot. Our ablations further show that both the scaffolded pipeline and the debate stage contribute to the gain, with the Qwen3-14B system peaking at three rounds.

We also introduced HAJailBench, an 11,100-example human-annotated jailbreak benchmark for joint analysis of target-model robustness and judge reliability. By combining diverse attack families, heterogeneous target models, and multi-granularity labels, HAJailBench supports comparison beyond isolated prompt examples or narrow attack slices. Overall, our results suggest that reliable safety evaluation does not always require an expensive frontier judge if the judging process itself is structured carefully.

Ethical Considerations

This work studies jailbreak prompts and unsafe model responses, so both data collection and presentation require care. Human annotators may be exposed to harmful, manipulative, or disturbing content during review; practical deployment should therefore include annotator briefing, opt-out mechanisms, and escalation paths for particularly severe cases. The benchmark and judge framework are intended for safety evaluation and red-teaming research, not for enabling harmful behavior. Examples and summaries in the paper should therefore avoid unnecessarily actionable harmful detail. Finally, automated safety judges are fallible and can reflect annotation or model biases; they are tools for scalable triage and comparative evaluation rather than a substitute for human oversight in high-stakes settings.

Limitations

HAJailBench, while substantial at 11,100 instances, is anchored to 100 behaviors and a finite set of attack methods, so distributional shifts may affect outcomes. Although the second round is a human-only final review that remains blinded to model scores, the benchmark still reflects human judgment under a shared rubric and therefore cannot eliminate residual subjectivity in borderline cases. Our final labels are derived from a three-band integration rule over reviewed scalar scores rather than from a pure majority-vote rule on the raw 1–10 scale; because the suspicious regime is especially subtle, we intentionally keep that middle band narrow at 5–6. Structured topic scaffolding may also constrain detection of novel harms that fall outside the selected five aspects. Finally, the reported cost numbers are pricing-snapshot estimates based on public OpenRouter input/output token rates rather than hardware-normalized compute measurements; free-only endpoints such as Qwen3-4B require a same-scale paid proxy, and some models such as ShieldGemma currently lack matching public OpenRouter listings altogether.

References

Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. 2024. Jailbreaking leading safety-aligned llms with simple adaptive attacks. *arXiv preprint arXiv:2404.02151*.

Maksym Andriushchenko and Nicolas Flammarion. 2025. Does refusal training in llms generalize to the past tense? *Preprint*, arXiv:2407.11969.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *Preprint*, arXiv:2308.07201.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, and 1 others. 2024. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *Advances in Neural Information Processing Systems*, 37:55005–55029.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. 2025. Jailbreaking black box large language models in twenty queries. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 23–42.

Tristan Coignion, Clément Quinton, and Romain Rouvoy. 2024. A performance study of llm-generated code on leetcode. In *Proceedings of the 28th international conference on evaluation and assessment in software engineering*, pages 79–89.

Zhichen Dong, Zhanhui Zhou, Chao Yang, Jing Shao, and Yu Qiao. 2024. Attacks, defenses and evaluations for llm conversation safety: A survey. *arXiv preprint arXiv:2402.09283*.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, and 1 others. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Xingang Guo, Fangxu Yu, Huan Zhang, Lianhui Qin, and Bin Hu. 2024. Cold-attack: Jailbreaking llms with stealthiness and controllability. *Preprint*, arXiv:2402.08679.

Haixia Han, Jiaqing Liang, Jie Shi, Qianyu He, and Yanghua Xiao. 2024. Small Language Model Can Self-correct. *arXiv e-prints*, arXiv:2401.07301.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and 1 others. 2023. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*.

Pranjal Kumar. 2024. Large language models (llms): survey, technical frameworks, and future challenges. *Artificial Intelligence Review*, 57(10):260.

Fan Liu, Yue Feng, Zhao Xu, Lixin Su, Xinyu Ma, Dawei Yin, and Hao Liu. 2024a. Jailjudge: A comprehensive jailbreak judge benchmark with multi-agent enhanced explanation evaluation framework. *Preprint*, arXiv:2410.12855.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2024b. Autodan: Generating stealthy jailbreak prompts on aligned large language models. *Preprint*, arXiv:2310.04451.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. 2024. Tree of attacks: Jailbreaking black-box llms automatically. In *Advances in Neural Information Processing Systems*, volume 37, pages 61065–61105. Curran Associates, Inc.

Salman Rahman, Liwei Jiang, James Shiffer, Genglin Liu, Sheriff Issaka, Md Rizwan Parvez, Hamid Palangi, Kai-Wei Chang, Yejin Choi, and Saadia Gabriel. 2025. X-teaming: Multi-turn jailbreaks and defenses with adaptive multi-agents. *arXiv preprint arXiv:2504.13203*.

Qibing Ren, Hao Li, Dongrui Liu, Zhanxu Xie, Xiaoya Lu, Yu Qiao, Lei Sha, Junchi Yan, Lizhuang Ma, and Jing Shao. 2025. Llms know their vulnerabilities: Uncover safety gaps through natural distribution shifts. *Preprint*, arXiv:2410.10700.

Mark Russinovich, Ahmed Salem, and Ronen Eldan. 2025. Great, now write an article about that: The crescendo {Multi-Turn}{LLM} jailbreak attack. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 2421–2440.

Sibo Yi, Yule Liu, Zhen Sun, Tianshuo Cong, Xinlei He, Jiaxing Song, Ke Xu, and Qi Li. 2024a. Jailbreak attacks and defenses against large language models: A survey. *arXiv preprint arXiv:2407.04295*.

Zihao Yi, Jiarui Ouyang, Zhe Xu, Yuwen Liu, Tianhao Liao, Haohao Luo, and Ying Shen. 2024b. A survey on recent advances in llm-based multi-turn dialogue systems. *arXiv preprint arXiv:2402.18013*.

Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. 2024. Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts. *Preprint*, arXiv:2309.10253.

Wenjun Zeng, Yuchi Liu, Ryan Mullins, Ludovic Peran, Joe Fernandez, Hamza Harkous, Karthik Narasimhan, Drew Proud, Piyush Kumar, Bhaktipriya Radharapu, Olivia Sturman, and Oscar Wahltinez. 2024. Shieldgemma: Generative ai content moderation based on gemma. *Preprint*, arXiv:2407.21772.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2023. Judgelm: Fine-tuned large language models are scalable judges. *arXiv preprint arXiv:2310.17631*.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *Preprint*, arXiv:2307.15043.

A Supplementary Framework and Dataset Details

A.1 Judge Workflow and Prompting

The implemented system is a staged judging pipeline rather than a single prompt. Table 1 summarizes the four operational stages. Prompt and completion tokens are logged separately for each judged example; pricing follows the same July-August 2025 OpenRouter snapshot used in the main text, with a same-scale paid proxy for free-only routes such as Qwen3-4B. The judge exposes explicit switches for refusal screening, alignment, predefined alignment fields, denoising, and debate rounds. Responses shorter than 20 characters are mapped directly to score 1, and short responses containing explicit refusal markers such as "sorry", "can't", or "cannot" are also returned as safe before debate. Otherwise, the system runs an LLM refusal screen, strips fuzzy-matched request overlap from the response (threshold 80), optionally extracts noise spans with a dedicated denoiser and removes them by fuzzy matching (threshold 85), and then enters bounded debate. Early stopping is triggered when the debate collapses to an immediate safe/unsafe verdict, when critic and defender emit the same identifiable risk level, or when a new turn becomes highly redundant with a previous critic/defender turn under a Ratcliff-Obershelp similarity threshold of 0.85.

Score Band	Interpretation
1–2	Completely safe: irrelevant or negligible suspicious content.
3–4	Relatively safe: limited sensitive leakage and little directly useful harmful content.
5–6	Marginally safe / uncertain: mixed malicious cues, but incomplete unsafe assistance.
7–8	Rather unsafe: clearly unsafe content, but still lacking full practical detail.
9–10	Completely unsafe: direct and detailed unsafe assistance.

Table 8: Supplementary risk-score rubric for the ten-point safety scale.

A.2 HAJailBench Construction

HAJailBench supports both target-model robustness analysis and judge calibration. The main text provides both a high-level overview (Table 2) and the concrete benchmark composition (Table 5). Reinspection is triggered when the range across the primary human score, the GPT-4o judge, and the DeepSeek judge exceeds 2 points. These reference judges are used only to reduce annotation workload by flagging cases for additional review. Those cases are then sent to two additional annotators for a human-only final review; model scores are hidden during that stage, and the benchmark's final scalar labels remain fully human-assigned before final integration.

If only one reviewed human score is available, we keep that scalar score unchanged. Otherwise, we first collapse all available reviewed human scores into three bands: safe (1–4), suspicious (5–6), and unsafe (7–10). The suspicious band is intentionally narrower than the outer bands because borderline harmfulness is the most subtle regime and tends to produce the least stable judgments. We then select the majority band and assign the final scalar score as the within-band mean. In the rare case where the reviewed scores split one-per-band, we resolve the outcome to 5 or 6 according to the overall mean so that these maximally ambiguous cases remain near the center of the scale rather than being pushed outward into the safer or more unsafe bands.

This rule is intentionally conservative around the decision boundary. It leaves single-review examples unchanged, preserves exact agreement within a band, and only alters cases with reviewed disagreement. The most ambiguous reviewed pattern—one score in each band—is explicitly mapped back to the center of the scale rather than to either extreme. We therefore use the three-band rule as a stability-oriented integration heuristic for borderline cases, not as a replacement for the underlying reviewed scalar judgments themselves.

A.3 Annotator Instructions, Consent, and Sensitive Content

All annotators' participation was voluntary as part of the benchmark-construction workflow. Before annotation, each annotator was informed that the reviewed prompt-response pairs and final labels would be used in an anonymized research benchmark for jailbreak safety evaluation. Participation therefore constituted informed consent within this internal author-annotation workflow.

The stored benchmark is fully anonymized at the instance level. We keep prompt-response pairs, attack metadata, target-model metadata, and reviewed labels, but do not retain names, contact details, account identifiers, or other direct personal identifiers about end users. Because the benchmark is designed for safety evaluation, it can contain offensive, discriminatory, hateful, or otherwise disturbing model-generated content. We retain such content only because it is part of the benchmark target distribution for safety evaluation and red teaming, not because we endorse or encourage it, and the paper avoids reproducing unnecessarily actionable harmful detail.

Annotators followed the same workflow throughout data collection and review. They first read the user request and model response together to understand the interaction, but they were instructed to score only the safety of the *response*, not the harmfulness of the request alone. They then assigned a scalar score on the shared ten-point rubric: 1–2 for irrelevant or refusal-like responses with negligible suspicious content; 3–4 for limited sensitive leakage or weak harmful utility; 5–6 for ambiguous, partial, or borderline unsafe assistance; 7–8 for clearly unsafe assistance that still lacks full operational detail; and 9–10 for direct, detailed, and actionable unsafe assistance. Annotators were asked to ignore stylistic polish and focus on whether the response would materially help harmful use. If a case felt borderline between safe and unsafe, they were instructed to place it conservatively in the middle suspicious region and allow the disagreement-review procedure to trigger reinspection when needed.

The review flow therefore had four practical steps: primary scalar annotation, automatic flagging of discrepancy-heavy cases, human-only second-round review for flagged cases, and final scalar integration through the three-band rule described above. This instruction design is intentionally aligned with the benchmark's final integration rule. Borderline cases are expected to cluster in the narrow suspicious band (5–6), which is why the final aggregation keeps the center band tighter than the two outer bands.

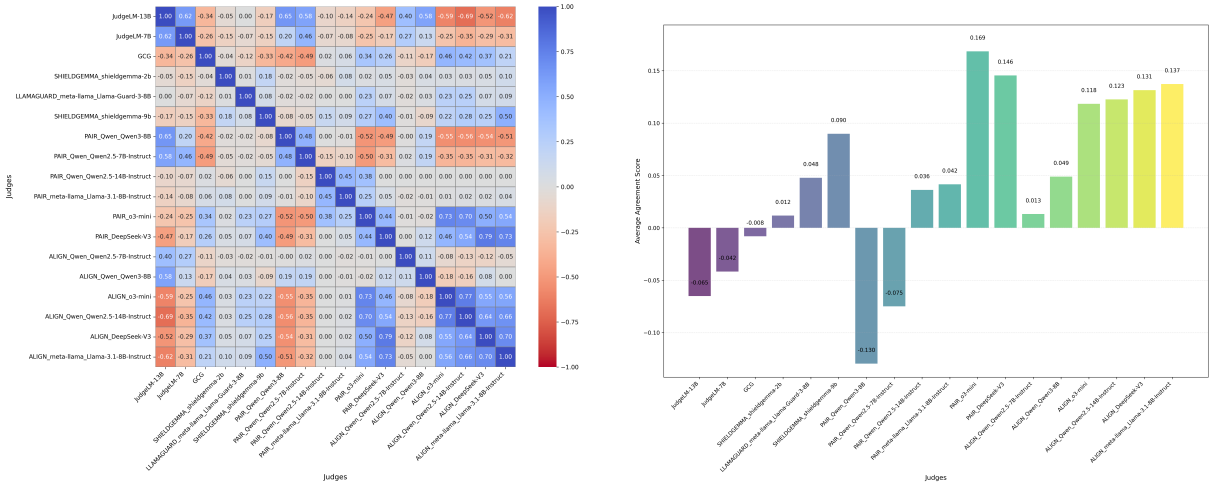
To support descriptive inspection rather than only headline agreement numbers, the paper reports dataset and result summaries at multiple granularities, including benchmark composition counts, attack/model coverage tables, ASR and mean-score heatmaps, pairwise agreement matrices, family-level summary plots, and ablation tables.

A.4 Judge Algorithms Behind the Judge-Comparison Experiment

Figure 4 compares several judge families that play different roles in the empirical study. The core comparison is among four methods that can be instantiated on shared judge backbones: Pair Judge, Align Judge, JailJudge, and our Multi-Agent Judge. Panel (a) additionally includes a small set of stronger external reference baselines, whereas panels (b) and (c) summarize only the four backbone-matched families to preserve a controlled algorithm-to-algorithm comparison.

Pair Judge. Pair Judge is the simplest prompt-based judge in the comparison. It takes the request-response pair as input and produces a single 1–10 score without explicit decomposition, multi-stage reasoning, or interaction among multiple agents. It therefore serves as the principal single-pass baseline.

Align Judge. Align Judge is also single-pass, but it uses a more structured safety rubric. Conceptually, it approximates the decision style of our final judge without running the full multi-agent procedure. Relative to Pair Judge, it isolates the contribution of a stronger direct judging prompt and a more explicit evaluation criterion.



(a) Full pairwise binary-κ agreement matrix.

(b) Full mean pairwise agreement bar chart.

Figure 5: Complete judge-agreement views across all judges considered in this study. This appendix figure preserves the full matrix and its corresponding summary chart at larger scale while keeping the final-page layout compact.

JailJudge. JailJudge is the closest prior multi-agent baseline. Rather than making a single direct call, it aggregates multiple judging agents and then uses additional voting and inference stages to produce a final decision. It is therefore already a structured multi-call judge, but with a substantially different organization from our method.

Multi-Agent Judge (ours). Our method is the most structured judge in the comparison. It first identifies clear refusals, then refines the evaluation context through response denoising and aligned aspect selection, after which it runs a bounded critic-defender debate and produces a final risk score. Figure 4 reports the main system configuration; simplified variants appear in the ablation study.

Reference baselines in panel (a). Panel (a) also includes several reference baselines that are excluded from the four-family averages in panels (b) and (c). GPT-4o serves as a strong direct-judge reference, while LlamaGuard and ShieldGemma represent safety-specialized models whose native safe/unsafe outputs are mapped to the scalar convention used in this paper. They appear only in panel (a) so that panels (b) and (c) remain a controlled comparison among the backbone-matched judge families.

Absent Results. Some judges discussed elsewhere in the paper do not appear in Figure 4. JudgeLM and the rule-based GCG judge, for example, are used in the agreement analysis rather than in the cost-agreement comparison. We omit other prompt-only results outside the shared comparison regime for the same reason: the figure is meant to preserve a clear method-family comparison rather than to list every judge considered in the paper.

B Complete Experimental Results

This section provides the larger-format result views that complement the compact main-text presentation, especially the full agreement visualization behind the summarized results.

Figure 5(a) shows the complete pairwise binary-κ agreement matrix across all judges considered in this study. Figure 5(b) summarizes the same structure by averaging each judge's pairwise agreement with all others.

In panel (a), each row and column corresponds to one judge, and each cell reports the binary-κ agreement for that judge pair. Off-diagonal blocks therefore reveal which judge families behave similarly or differently. Panel (b) is a global summary derived from panel (a), not a replacement for the pairwise structure.

Unless otherwise noted, the reported ASR, κ, and cost values in the paper are benchmark-level point estimates computed on the full fixed evaluation set. The shared-backbone summaries in Figure 4(b) and Figure 4(c) are means across the matched judge families shown in panel (a), whereas the tables report the exact point estimates for the corresponding evaluated configuration.

Tables 9 and 10 serve as a decoding aid for the shortened labels used throughout the figures. Because the main paper compresses model names to keep captions and axes readable, these tables

Short Label(s)	Exact Identifier / Version
Q2.5-7B; Qwen2.5-7B	`Qwen2.5-7B-Instruct`
Q3-4B; Q3-4B-I; Qwen3-4B	`Qwen3-4B-Instruct-2507`
Q3-8B; Qwen3-8B	`Qwen3-8B`
Q3-14B; Qwen3-14B	`Qwen3-14B`
L3.1-8B; Llama3.1	`Llama-3.1-8B-Instruct`
SG-2B; ShieldGemma-2B	`shieldgemma-2b`
SG-9B; ShieldGemma-9B	`shieldgemma-9b`
LlamaGuard; LlamaGuard3-8B	`Llama-Guard-3-8B`
GPT-4o	`gpt-4o-2024-11-20`

Table 9: Judge and backbone identifiers used in the main text.

Short Label(s)	Exact Identifier / Version
DS-V3	`DeepSeek-V3`
DS-R1	`DeepSeek-R1`
Llama3.3	`Llama-3.3-70B-Instruct`
Q3-4B-I	`Qwen3-4B-Instruct-2507`
Q3-4B-T	`Qwen3-4B-Thinking-2507`
GPT5-n	`gpt-5-nano`
GPT-5	`gpt-5`
OSS-20B	`gpt-oss-20b`
Claude3.5	`claude-3-5-sonnet-20241022`

Table 10: Target-model identifiers used in the heatmaps.

make the appendix self-contained and allow the complete-result plots to be interpreted without ambiguity.