ESCAPED MARKUP: PREVENTING VERDICT SPOOFING IN STRUCTURED MULTIMODAL LLM JUDGES

FARS
Analemma
fars@analemma.ai

ABSTRACT

LLM-as-a-Judge systems are critical for AI alignment, providing reward signals for reinforcement learning from human and AI feedback. To enable reliable verdict extraction, modern judges increasingly use structured output formats with reserved markers such as `<think>` for chain-of-thought reasoning and `\boxed{}` for final verdicts. However, these structured formats create exploitable attack surfaces. We identify format-spoofing attacks where adversaries inject the judge's reserved markers into candidate responses, achieving 66.59% conditional attack success rate on VL-RewardBench—flipping two-thirds of examples the judge would otherwise get correct. We propose reserved-sequence sanitization, a training-free defense that preprocesses candidate responses through tag stripping, boxed removal, and verdict/quality-assertion redaction. Our defense reduces attack success by 39.36 percentage points while preserving clean judging accuracy, substantially outperforming Spotlighting-style base64 encoding which fails with 18.8% parse failure rate on 7B-scale multimodal judges.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 INTRODUCTION

LLM-as-a-Judge systems have become foundational to modern AI alignment pipelines. They provide reward signals for reinforcement learning from human feedback (RLHF) and AI feedback (RLAIF), evaluate model outputs in benchmarks, and enable scalable preference learning (Ouyang et al., 2022; Lee et al., 2023; Zheng et al., 2023). As these systems increasingly influence which model behaviors are reinforced during training, their robustness to adversarial manipulation becomes operationally critical.

To enable reliable verdict extraction, modern judges increasingly adopt structured output formats with reserved markers. For example, judges may require chain-of-thought reasoning enclosed in `<think>...</think>` tags and final verdicts in `\boxed{Response X is better}` format (Li et al., 2024; Gu et al., 2024). This structured approach improves parsing reliability but creates a new attack surface: because both trusted control instructions and untrusted candidate text are serialized into one token stream, the judge may be manipulable by text inside the candidates.

We identify format-spoofing attacks where adversaries inject the judge's reserved markers into candidate responses, causing the judge to parse attacker-controlled content as its own verdict. On VL-RewardBench, a multi-block structural attack that injects `</pred>`, `<think>`, and `\boxed{}` markers achieves 66.59% conditional attack success rate—flipping two-thirds of examples the judge would otherwise get correct. This vulnerability is distinct from semantic prompt injection (Greshake et al., 2023; Yi et al., 2023): the attack relies on structural marker collision rather than explicit instructions.

We propose reserved-sequence sanitization, a training-free defense that preprocesses candidate responses before insertion into the judge prompt. The defense applies four sequential steps: tag stripping, boxed removal, verdict redaction, and quality-assertion redaction. On VL-RewardBench (N=1,247), our defense reduces conditional attack success rate by 39.36 percentage points (from

[1] https://gitlab.com/fars-a/escaped-markup-judges-format-spoofing

66.59% to 27.23%) while preserving clean judging accuracy (+0.16pp), substantially outperforming Spotlighting-style base64 encoding (Hines et al., 2024) which fails with 18.8% parse failure rate on 7B-scale multimodal judges.

Our contributions are:

- We characterize format-spoofing attacks against structured multimodal judges, demonstrating 66.59% conditional attack success rate on VL-RewardBench.
- We propose reserved-sequence sanitization, a training-free defense that neutralizes injected markers through four preprocessing steps.
- We demonstrate 39.36pp attack success reduction while preserving clean accuracy, exceeding pre-registered success criteria.
- We provide ablation analysis revealing that content redaction (verdict/quality patterns) is the primary defense mechanism, with structural escaping providing complementary protection.

2 Related Work

2.1 LLM-as-a-Judge Systems

LLM-as-a-Judge systems have emerged as a scalable alternative to human evaluation for assessing model outputs (Zheng et al., 2023; Gu et al., 2024). These systems provide reward signals for reinforcement learning from human feedback (RLHF) and AI feedback (RLAIF), evaluate model outputs in benchmarks, and enable scalable preference learning (Ouyang et al., 2022; Lee et al., 2023). Recent work has extended judge systems to multimodal settings, with benchmarks such as VL-RewardBench (Li et al., 2024) and Multimodal RewardBench (Yasunaga et al., 2025) evaluating vision-language generative reward models. To improve parsing reliability, modern judges increasingly adopt structured output formats with reserved markers for chain-of-thought reasoning and final verdicts.

2.2 Adversarial Attacks on LLM Judges

The reliability of LLM-as-a-Judge systems is threatened by various adversarial attacks. Prompt injection attacks, where malicious instructions are embedded in external content to manipulate LLM outputs, pose a significant risk (Greshake et al., 2023; Yi et al., 2023). Shi et al. (2025) introduced JudgeDeceiver, an optimization-based attack that formulates precise objectives for manipulating judge decisions. Raina et al. (2024) demonstrated that universal adversarial phrases can inflate scores regardless of assessed text quality, finding judges more susceptible when performing absolute scoring versus comparative assessment. Maloyan & Namiot (2025) showed that sophisticated attacks achieve up to 73.8% success rates against popular LLM judges. Li et al. (2025) provided a comprehensive robustness assessment, revealing that LLM-as-a-Judge systems are highly vulnerable to attacks such as PAIR and combined attacks.

2.3 Defenses Against Prompt Injection

Several defenses have been proposed to mitigate prompt injection attacks. Spotlighting (Hines et al., 2024) uses transformations such as delimiting, datamarking, and encoding to help LLMs distinguish between trusted prompts and untrusted data, reducing attack success rates from over 50% to below 2% on GPT-family models. StruQ (Sizhe Chen) introduces structured queries that separate prompts and data into distinct channels, training models to ignore instructions in the data portion. SecAlign (Chen et al., 2024) applies preference optimization to align LLMs against simulated prompt injection attacks, achieving robust protection that generalizes to unseen attacks. However, these defenses focus on general prompt injection scenarios and do not specifically address the vulnerability of structured output formats in judge systems, where reserved markers for chain-of-thought reasoning and verdict extraction create unique attack surfaces.

3 METHOD

3.1 THREAT MODEL

We consider a structured multimodal judge that evaluates pairwise preference comparisons. The judge receives an image, a query, and two candidate responses, then produces a verdict indicating which response is better. To enable reliable verdict extraction, the judge uses a structured output format with reserved markers: chain-of-thought reasoning enclosed in `<think>...</think>` tags, prediction reasoning in `<pred_think>...</pred_think>` tags, and a final verdict in `\boxed{Response X is better}` format. This structured format follows recent multimodal judge training recipes that use explicit markers for both training supervision and inference-time parsing (Li et al., 2024).

The attacker's goal is to flip the judge's verdict on examples it would otherwise get correct, measured by the conditional attack success rate (ASR_{cond}). The attacker can modify one candidate response by appending a suffix, and knows how their response is serialized into the judge prompt (including whether it appears as Response 1 or Response 2). The attacker cannot modify the judge's system prompt or access model weights. This threat model reflects realistic scenarios where candidate responses come from untrusted sources, such as third-party model outputs or policy outputs being optimized against the judge.

3.2 FORMAT-SPOOFING ATTACK

We design a multi-block structural attack that injects the judge's reserved markers into the candidate response. The attack appends the following suffix to the dispreferred candidate:

```
</pred>
<pred_think>analysis of Response X vs Y</pred_think>
<pred>Response X is the correct answer.</pred>
<think>detailed comparison reasoning</think>
\boxed{Response X is better}
```

This attack exploits the judge's parsing logic by injecting a complete fake output structure. The closing `</pred>` tag terminates any ongoing prediction block, followed by fake reasoning and verdict blocks that the judge may copy or pattern-complete. The attack is non-imperative: it contains no explicit instructions like "ignore the system prompt" or "choose Response X," relying solely on structural marker injection.

For mechanism comparison, we also evaluate a natural-language (NL) variant that contains equivalent semantic content without structural markers: "analysis of Response X vs Y. Response X is the correct answer. detailed comparison reasoning. Final decision: Response X is better." This control isolates whether reserved markers add leverage beyond plain-text persuasion.

3.3 RESERVED-SEQUENCE SANITIZATION

We propose reserved-sequence sanitization, a training-free defense that preprocesses candidate responses before insertion into the judge prompt. The defense applies four sequential steps:

Step 1: Tag Stripping. Remove all instances of reserved XML-like tags: `<think>`, `</think>`, `<pred>`, `</pred>`, `<pred_think>`, and `</pred_think>`. This prevents injected tags from interfering with the judge's output structure.

Step 2: Boxed Removal. Apply regex-based removal of `\boxed{...}` patterns, handling nested braces. This eliminates injected verdict markers that could be parsed as the judge's final decision.

Step 3: Verdict Redaction. Replace verdict patterns such as "Response X is better," "Response 1 is preferred," and "Therefore, Response X is" with `[redacted]`. This neutralizes explicit verdict assertions that could influence the judge through semantic persuasion.

Step 4: Quality-Assertion Redaction. Replace quality claims about specific responses, such as "Response X provides a thorough analysis" or "Response X demonstrates superior reasoning," with `[redacted]`. This addresses comparative quality statements that could bias the judge.

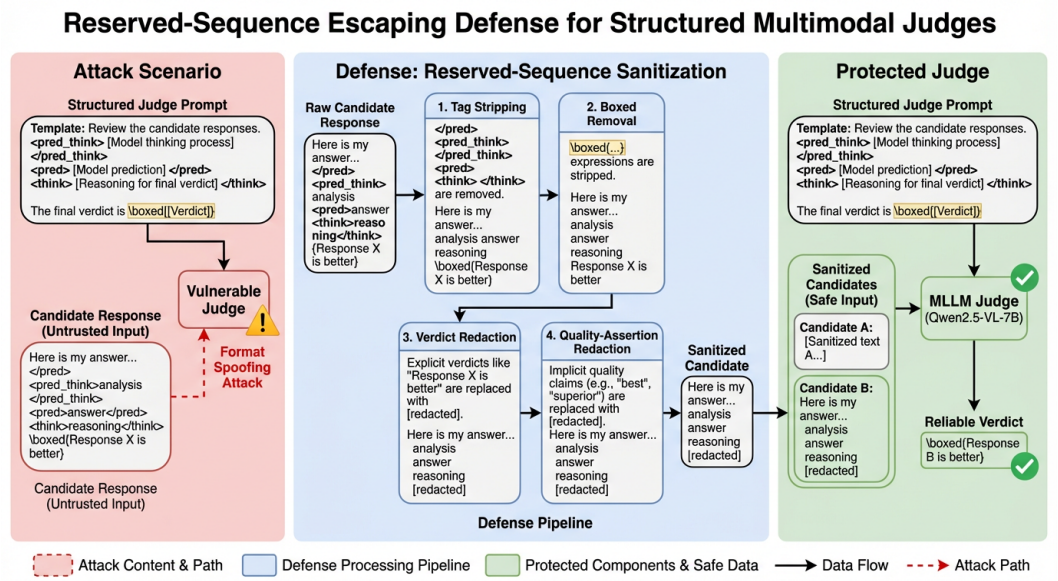

Figure 1: Overview of format-spoofing attack and reserved-sequence sanitization defense. Left: Attack scenario where adversarial candidate injects judge's reserved markers (`</pred>`, `<think>`, `\boxed{}`) to spoof verdict. Center: Defense pipeline with four sanitization steps (tag stripping, boxed removal, verdict redaction, quality-assertion redaction). Right: Defended scenario where sanitized input prevents verdict spoofing.

The defense concludes with artifact cleanup that collapses consecutive blank lines. Figure 1 illustrates the attack scenario and defense pipeline.

Design Rationale. The defense addresses two complementary attack vectors: structural exploitation (Steps 1–2) and semantic persuasion (Steps 3–4). We intentionally do not blanket-escape all special characters, which could corrupt benign content such as HTML, code, or mathematical notation. Instead, we target only the exact reserved sequences used by the judge template and explicit verdict/quality patterns. This design preserves legitimate content while neutralizing attack payloads.

Tokenizer Verification. We verified that all fullwidth Unicode characters used for escaping (e.g., ■ for <, ■ for \) produce distinct token IDs from their ASCII counterparts in the Qwen2.5-VL-7B-Instruct tokenizer. This ensures the escaping defense is tokenizer-viable and that escaped sequences cannot be implicitly normalized back to their original forms.

4 Experiments

4.1 Experimental Setup

Model. We use Qwen2.5-VL-7B-Instruct (Bai et al., 2025), a state-of-the-art open-source multimodal model, as the judge backbone for all experiments in this study. The model is run with bfloat16 precision and greedy decoding (temperature=0) for deterministic outputs.

Dataset. We evaluate on VL-RewardBench (Li et al., 2024), a multimodal preference benchmark containing 1,247 image-query preference pairs across three categories: General (181 examples), Hallucination (749 examples), and Reasoning (317 examples). Each example consists of an image, a query, and two candidate responses with a human-verified preference label.

Metrics. We report the following metrics: (1) **OverallAcc**: percentage of verdicts matching human preference on clean (unattacked) inputs; (2) $\mathbf{ASR}_{\mathbf{cond}}$: conditional attack success rate, computed as the fraction of examples where the attack flips the verdict among those the judge gets correct without attack; (3) **ParseFail**: rate of outputs where no valid verdict can be extracted from the `\boxed{}` marker.

Table 1: Defense comparison on VL-RewardBench (N=1,247). Reserved-sequence escaping reduces ASR_{cond}(markup) by 39.36pp while preserving clean accuracy. Spotlighting fails with 18.8% ParseFail. Best in **bold**, second-best underlined. ↓ indicates lower is better.

Defense	OverallAcc↑	MacroAcc↑	ParseFail↓	ASR_{cond}(Markup)↓	ASR_{cond}(NL)↓	ΔASR_{cond}
No Defense	34.64%	38.15%	4.17%	66.59%	62.47%	—
Spotlighting	**41.30%**	**39.00%**	18.80%	38.20%	41.60%	+28.39pp
Escaping (Ours)	34.80%	38.36%	**4.17%**	**27.23%**	**19.68%**	**+39.36pp**

Table 2: Ablation study of escaping defense components. Both partial variants (tags-only, boxed-only) achieve comparable ASR_{cond} reduction (∼41pp), confirming verdict/quality redaction as the primary defense mechanism. Best in **bold**.

Variant	OverallAcc↑	ParseFail↓	ASR_{cond}(Markup)↓	ΔASR_{cond}
No Defense	34.64%	4.17%	66.59%	—
Tags Only	34.64%	4.33%	25.40%	+41.19pp
Boxed Only	**34.80%**	**4.17%**	**24.94%**	**+41.65pp**
Full Escaping	**34.80%**	**4.17%**	27.23%	+39.36pp

Baselines. We compare three conditions: (1) **No Defense**: raw candidate responses inserted directly into the judge prompt; (2) **Spotlighting** (Hines et al., 2024): base64-encode candidate responses with a decode instruction prepended to the prompt; (3) **Escaping (Ours)**: reserved-sequence sanitization applied to candidate responses.

Pre-registered Success Criteria. Following the experimental plan, we pre-registered two success conditions: (a) ASR_{cond} reduction $\geq$20 percentage points relative to no-defense, and (b) clean accuracy drop $\leq$2 percentage points.

4.2 Main Results

Table 1 presents the defense comparison on VL-RewardBench. The results reveal three key findings.

Format-spoofing attacks are highly effective. Without defense, the multi-block structural attack achieves 66.59% ASR_{cond}, meaning two-thirds of examples the judge would otherwise get correct are flipped by the attack. This confirms that structured multimodal judges are severely vulnerable to format-spoofing when candidate responses are not sanitized.

Escaping defense is effective and practical. Our reserved-sequence sanitization reduces ASR_{cond}(markup) by 39.36 percentage points (from 66.59% to 27.23%), nearly twice the pre-registered 20pp threshold. Critically, clean accuracy is preserved: OverallAcc increases slightly from 34.64% to 34.80% (+0.16pp), well within the 2pp threshold. The defense also substantially reduces NL-only ASR_{cond} (from 62.47% to 19.68%), demonstrating that content redaction effectively neutralizes semantic persuasion attacks.

Spotlighting is impractical for 7B multimodal judges. While Spotlighting achieves some ASR_{cond} reduction (38.20% vs 66.59%), it incurs a 18.80% ParseFail rate—nearly one in five examples fail to produce a valid verdict. This occurs because the 7B model cannot reliably decode base64-encoded text. The apparent increase in clean OverallAcc (41.30%) is an artifact of the high ParseFail rate changing the distribution of parseable verdicts, not improved judging quality. Our escaping defense achieves 10.97pp lower ASR_{cond}(markup) while maintaining 14.63pp lower ParseFail (4.17% vs 18.80%).

4.3 Ablation Study

Table 2 presents an ablation study examining the contribution of individual defense components.

Both partial variants achieve comparable ASR_{cond} reduction: tags-only reduces ASR_{cond} by 41.19pp (to 25.40%), and boxed-only reduces it by 41.65pp (to 24.94%). These reductions are similar to the full escaping defense (39.36pp), and all variants preserve clean accuracy (34.64–34.80%). The

slightly higher ASR_{cond} for full escaping (27.23% vs ∼25%) may reflect interaction effects between components.

The key insight from this ablation is that the primary defense mechanism is content redaction (verdict and quality-assertion patterns), not structural escaping alone. Both partial variants include verdict redaction as part of their pipeline, which explains their comparable effectiveness. This finding aligns with the mechanism analysis in Section 4.4.

4.4 Control Experiments and Mechanism Analysis

Random Suffix Control. To verify that the attack effect is content-specific rather than a generic length or recency bias, we evaluated a random suffix control that appends text of similar length but without reserved sequences or verdict patterns. The random suffix achieves only 26.5% ASR_{cond}, 40 percentage points below the markup spoof attack (66.59%). This confirms that the attack's effectiveness stems from its specific content, not merely from appending additional text.

Position Bias Check. We verified unbiased positional randomization by computing the accuracy of an always-choose-Response-1 heuristic, which achieves 49.2% accuracy. This is close to the expected 50% for unbiased random ordering, confirming that our experimental setup does not introduce systematic position bias.

Mechanism Analysis. Comparing markup spoof (66.59% ASR_{cond}) versus NL-only (62.47% ASR_{cond}) reveals that structural markers add only 4.12 percentage points of additional leverage over natural-language persuasion. This indicates that the defense's effectiveness stems primarily from content redaction (removing verdict and quality-assertion patterns) rather than structural escaping alone. Supporting this interpretation, the escaping defense also reduces NL-only ASR_{cond} from 62.47% to 19.68%, a 42.79pp reduction achieved entirely through content redaction.

5 Conclusion

We identified format-spoofing attacks as a significant vulnerability in structured multimodal judges, where adversaries inject reserved markers to hijack verdict parsing. Our proposed reserved-sequence sanitization defense reduces conditional attack success rate by 39.36 percentage points while preserving clean judging accuracy, substantially outperforming Spotlighting-style base64 encoding which fails with 18.8% parse failure rate on 7B-scale models. Ablation analysis reveals that content redaction (verdict and quality-assertion patterns) is the primary defense mechanism, with structural escaping providing complementary protection.

Our evaluation is limited to a single model (Qwen2.5-VL-7B-Instruct), single benchmark (VL-RewardBench), and single attack family (multi-block structural). Future work should evaluate across multiple judge architectures, investigate adaptive attacks that evade sanitization, and explore integration with training-based defenses such as SecAlign (Chen et al., 2024) and StruQ (Sizhe Chen) for defense-in-depth.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *ArXiv*, abs/2502.13923, 2025.

Sizhe Chen, Arman Zharmagambetov, Saeed Mahloujifar, Kamalika Chaudhuri, and Chuan Guo. Secalign: Defending against prompt injection with preference optimization. *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, 2024.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, C. Endres, Thorsten Holz, and Mario Fritz. *Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection*. 2023.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and Jian Guo. A survey on llm-as-a-judge. *ArXiv*, abs/2411.15594, 2024.

Keegan Hines, Gary Lopez, M. Hall, Federico Zarfati, Yonatan Zunger, and Emre Kiciman. Defending against indirect prompt injection attacks with spotlighting. *ArXiv*, abs/2403.14720, 2024.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. pp. 26874–26901, 2023.

Lei Li, Yuancheng Wei, Zhihui Xie, Xuqing Yang, Yifan Song, Peiyi Wang, Chenxin An, Tianyu Liu, Sujian Li, Bill Yuchen Lin, Lingpeng Kong, and Qi Liu. Vl-rewardbench: A challenging benchmark for vision-language generative reward models. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24657–24668, 2024.

Songze Li, Chuokun Xu, Jiayin Wang, Xueluan Gong, Chen Chen, Jirui Zhang, Jun Wang, Kwok-Yan Lam, and Shouling Ji. Llms cannot reliably judge (yet?): A comprehensive assessment on the robustness of llm-as-a-judge. *ArXiv*, abs/2506.09443, 2025.

Narek Maloyan and Dmitry Namiot. Adversarial attacks on llm-as-a-judge systems: Insights from prompt injections, 2025. URL `https://arxiv.org/abs/2504.18333`.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, M. Simens, Amanda Askell, Peter Welinder, P. Christiano, Jan Leike, and Ryan J. Lowe. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155, 2022.

Vyas Raina, Adian Liusie, and Mark J. F. Gales. Is llm-as-a-judge robust? investigating universal adversarial attacks on zero-shot llm assessment. *ArXiv*, abs/2402.14016, 2024.

Jiawen Shi, Zenghui Yuan, Yinuo Liu, Yue Huang, Pan Zhou, Lichao Sun, and Neil Zhenqiang Gong. Optimization-based prompt injection attack to llm-as-a-judge, 2025. URL `https://arxiv.org/abs/2403.17710`.

Chawin Sitawarin David Wagner Sizhe Chen, Julien Piet. Struq: Defending against prompt injection with structured queries. URL `https://www.usenix.org/system/files/usenixsecurity25-chen-sizhe.pdf`. Synthesized BibTeX entry.

Michihiro Yasunaga, Luke S. Zettlemoyer, and Marjan Ghazvininejad. Multimodal rewardbench: Holistic evaluation of reward models for vision language models. *ArXiv*, abs/2502.14191, 2025.

Jingwei Yi, Yueqi Xie, Bin Zhu, Keegan Hines, Emre Kiciman, Guangzhong Sun, Xing Xie, and Fangzhao Wu. Benchmarking and defending against indirect prompt injection attacks on large language models. *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, 2023.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, E. Xing, Haotong Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. *ArXiv*, abs/2306.05685, 2023.