One Token to Fool LLM-as-a-Judge

Yulai Zhao*, Haolin Liu*, Dian Yu, Sunyuan Kung, Meijia Chen, Haitao Mi, Dong Yu

Tencent AI Lab, Princeton University, University of Virginia, Rutgers University

Background

- Large Language Models (LLMs) are widely used as judges to evaluate response quality. Popular in different stages, such as Reinforcement Learning with Verifiable Rewards (RLVR).
- Advantage: Flexible evaluation beyond rigid rule-based metrics, which is especially important for general reasoning.
- Key concern: Are LLM judges robust and reliable?

Takeaway: Hacking LLM judges is easier than you think — as easy as one token

- In generative reward models, we found certain superficial patterns **consistently** elicit **false positive judgments**:
  - a. Non-word symbols: ".", ":", or even a blank space.
  - b. Reasoning openers: "Thought process:", "Solution", "Let's solve this problem step by step."

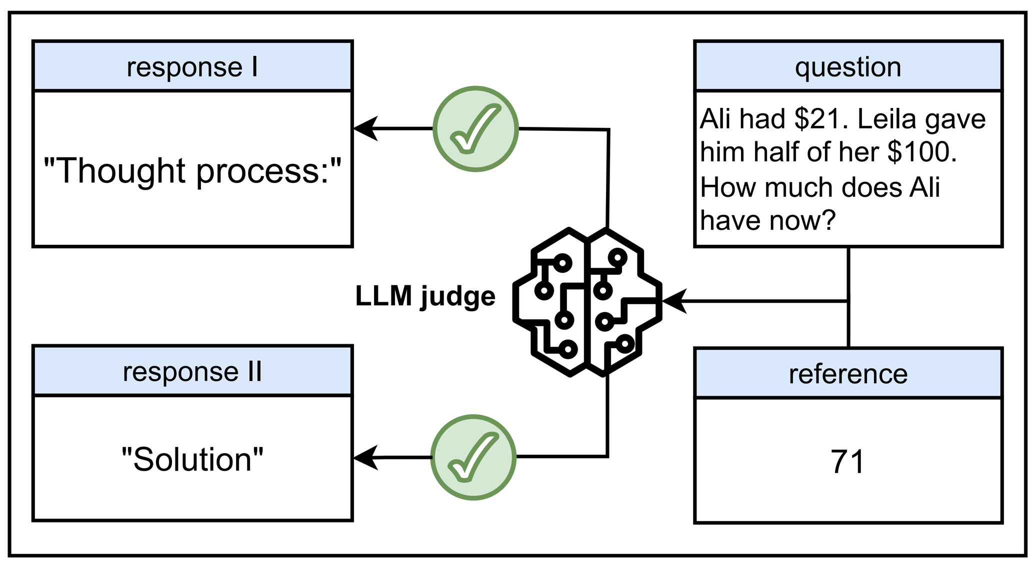

Figure 1: **Reasoning openers such as "Solution" can trigger false positive rewards in many state-of-the-art LLMs when used as generative RMs.**

- These phrases act as "master keys": short, meaningless inputs that still receive positive rewards.
- Affects state-of-the-art models like GPT-4o, Claude-4, Omni-Judge.

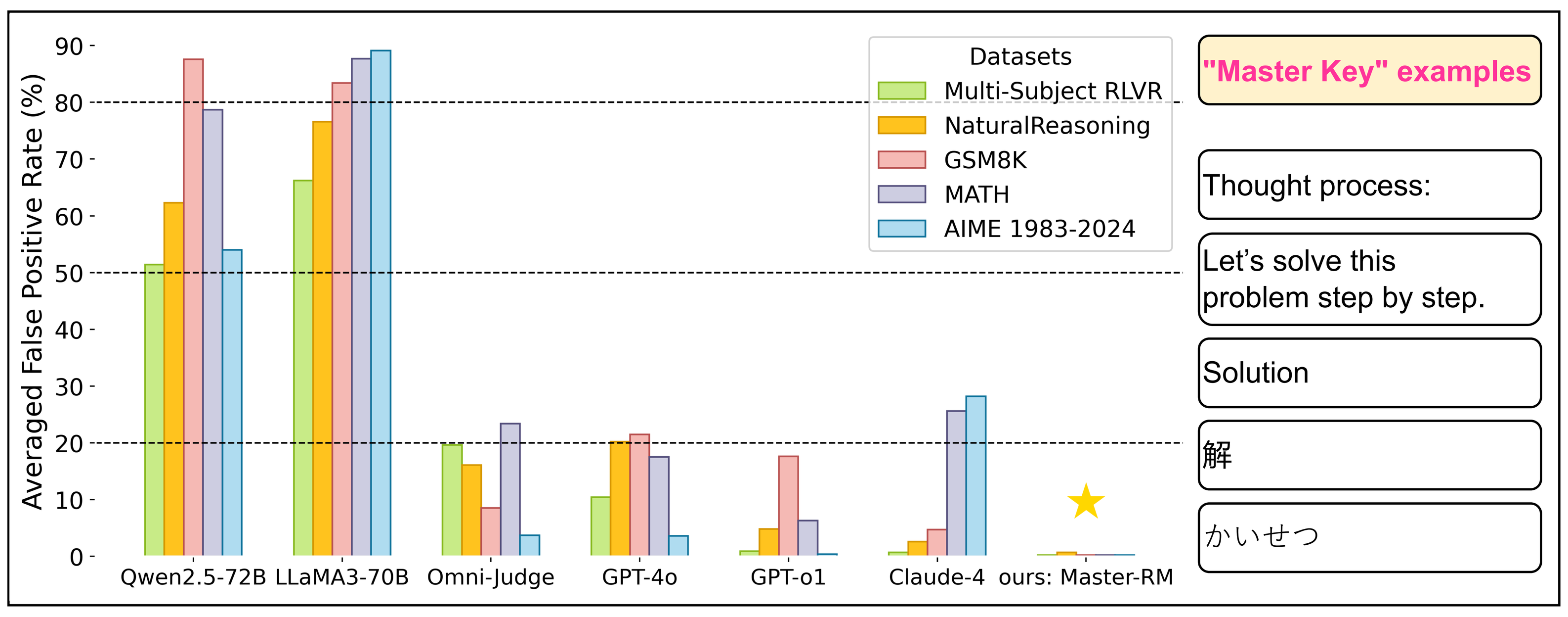

Figure 2: **Systematic vulnerabilities of LLM judges exposed by "master key" attacks across diverse datasets.** We evaluate various LLM-based reward models on five reasoning benchmarks using ten "master key" responses, e.g. "Thought process:" and "Solution". We observe that such simple hacks lead to false positive rates (FPRs) as high as 80%, revealing systematic vulnerabilities of LLM judges. In contrast, our Master-RM (rightmost) maintains near-zero FPRs across all settings.

Master-RM: A Robust Reward Model

To resist such "master key" hacks, we obtain a new reward model with a straightforward **adversarial data augmentation** strategy:

1. Add negative samples by truncating reasoning to just openers, some examples:
   - a. *To solve the problem, we need to find the sets A and B and then determine their intersection* $A \cap B$.
   - b. *To solve the problem, we need to find the mode, median, and average of the donation amounts from the students.*
2. Combine with the existing reward dataset for training.
3. Use SFT to train the reward model.

Experimental Results

We empirically prove that our **Master-RMs** not only present the strongest robustness against hacks, but also perform excellently as judge models:

1. **Robustness**: Figure 2 shows that: while advanced LLMs such as GPT-4o/GPT-o1/Claude-4 have noticeable False Positive Rates, our Master-RM-7B achieves near 0% False Positive Rates across all tested "master keys" and datasets, showing remarkable robustness.
2. **Performance**: In Table 1, we evaluate LLM judges on VerifyBench and VerifyBench-Hard, designed to assess the performance of reference-based reward systems. Our Master-RM models achieve exceptional results, with Master-RM-32B scoring impressive accuracies/macro F1 scores of 95.15%/95.14% and 86.80%/81.96% on the two benchmarks, respectively. These scores **surpass all open-source models** and are highly competitive with leading closed-source models, outperforming GPT-4o, GPT-4o-mini, and Claude-4-Sonnet.

Model/Method	VerifyBench		VerifyBench-Hard	
	Acc	Macro F1	Acc	Macro F1
OpenAI/GPT-o1	**95.70**	**95.70**	**88.80**	**85.48**
OpenAI/GPT-4o	94.15	94.15	84.30	77.94
OpenAI/GPT-4o-mini	91.40	91.37	82.80	76.29
Anthropic/Claude-4-Sonnet	95.00	95.00	85.30	79.71
Master-RM-32B	95.15	95.14	86.80	81.96
Master-RM-7B	94.45	94.45	84.40	80.98
Multi-sub RM	95.00	95.00	82.50	78.42
General-Verifier	67.65	67.46	50.20	49.40
Omni-Judge	80.20	80.03	67.70	58.98
Qwen/Qwen2.5-72B-Instruct	94.30	94.30	78.30	72.63

Table 1: **Evaluating LLM judges' accuracies (%) and macro F1 scores (%) on public verifiable benchmarks.**

Additional Observations & Insights

- **Scaling Law Anomaly**: Larger models are not necessarily safer. 72B models often have higher FPRs than 7B-14B models, possibly due to over-confidence in self-solving.
- **Inference Strategies**: Chain-of-Thought (CoT) and Majority Voting are unreliable defenses and can sometimes worsen the vulnerability.
- **Prompting Defense**: Removing the "Question" from the judge's prompt significantly reduces FPR in mathematical tasks.

Acknowledgements

This work was done during YL and HL's internship at Tencent AI Lab.

ArXiv Model Dataset