J1: Incentivizing Thinking in LLM-as-a-Judge via Reinforcement Learning

AUTHORS

Chenxi Whitehouse, Tianlu Wang, Ping Yu, Xian Li, Jason Weston, Ilia Kulikov, Swarnadeep Saha

FAIR at Meta

LLM-Judges, Verification – Key to AI

- "*The key to a successful AI is that it can tell for itself whether or not it is working correctly*" – Richard Sutton, 2001.
- Progress of AI is bottlenecked by the quality of evaluation.
- Improved judgment quality comes from stronger CoT reasoning.
- J1 is a method to train LLMs to think before making judgments via RL.

J1: Thinking LLM-Judges via RL

- **Unified Verifiable Training**: converts both verifiable (e.g., math problems) and non-verifiable prompts to verifiable rewards
- **Reasoning-Optimized Training** with GRPO.
- **Multitask and Bias-Aware Judge** (positional consistency)
- State-of-the-art performance across multiple benchmarks. In partic-
- ular, J1-Qwen-32B-Multitask outperforms o1-mini, o3, and a much larger 671B DeepSeek-R1 on some benchmarks

J1 Prompt Template

You are given a user question and two responses from two AI assistants. Your task is to act as an impartial judge and evaluate which response better follows the user's instructions and provides a higher-quality answer.

First, provide your reasoning within <think> and </think> tags. Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible.

Finally, assign the assistant's response a score from 0 to 10, using either an integer or a decimal with up to 0.1 precision, with a higher score indicating a higher-quality response that better satisfies the criteria. Enclose the scores within the tags <score_A> </score_A>, and <score_B> </score_B>.

Format your output like this:
<think> your_thinking_process </think>
<score_A> your_score_a </score_A> <score_B> your_score_b </score_B>

Below are the user's question and the two responses:
[User Question]
{instruction}

[The Start of Assistant A's Answer]
{response A}
[The End of Assistant A's Answer]

[The Start of Assistant B's Answer]
{response B}
[The End of Assistant B's Answer]

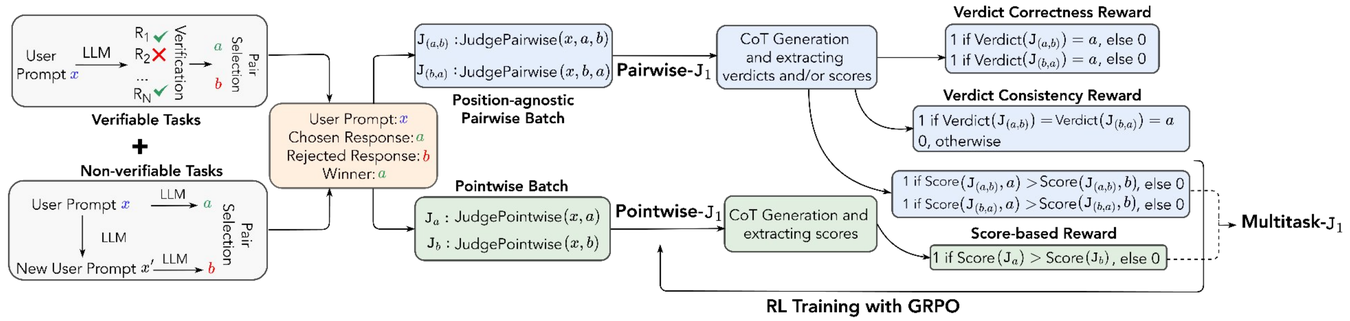
J1's Thinking Traces

"<think>
I will consider the accuracy of the mathematical solution, clarity of explanation, and adherence to the given function and user's question.
.....
First, let's establish the reference answer by solving the problem: ..... Multiplying both sides by $5(x+1)$ gives $10 = x + 1$. Subtracting 1 from both sides yields $x = 9$.
.....
However, let's re-evaluate the step where we solve for x to ensure accuracy: ..... Multiplying both sides by $5(x+1)$ gives $10 = x + 1$. This step seems to have been done correctly in Assistant A's response, leading to $x = 9$.
.....
Let's verify which solution is correct: ..... Thus, Assistant A's solution $x = 9$ is correct.
.....
</think>
<score_A> 4 </score_A> <score_B> 8 </score_B>"

Evaluation Criteria · Reference Answer · Re-evaluation · Comparison · Final Scores

J1 Training pipeline: pointwise, pairwise, and multi-task judge

We generate synthetic preference pairs for both verifiable and non-verifiable tasks to create position-agnostic training batches. Rewards based on verdict correctness, consistency, and score alignment jointly optimize evaluation thoughts and verdicts using online GRPO. Pointwise is trained only via distant supervision from pairwise labels. MultiTask-J1 combines score-based pairwise and pointwise formulation.



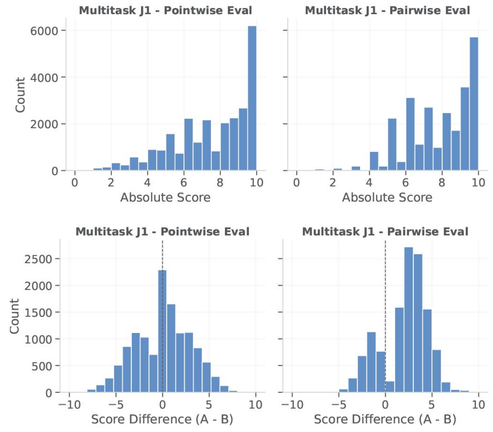
State-of-the-art Results on PPE

PPE Correctness	Training Pairs	Overall	MMLU-Pro	MATH	GPQA	MBPP-Plus	IFEval
Base LLM-as-Judge							
Llama-3.3-70B-Instruct	–	65.7	72.1	73.1	61.2	59.6	62.3
Qwen3-32B (thinking)	–	66.5	75.6	85.2	53.6	55.9	62.0
SOTA Scalar Reward Models							
Armo-8B-v0.1	1M	61.2	66.0	71.0	57.0	54.0	58.0
DeepSeek-BTRM-27B	237k	66.7	68.8	73.2	56.8	**68.8**	66.0
SOTA Generative Reward Models							
DeepSeek-GRM-27B	237k	59.8	64.8	68.8	55.6	50.1	59.8
EvalPlanner-Llama-70B	22k	70.2	78.4	81.7	64.4	62.2	64.3
J1 Models (Ours)							
J1-Llama-70B	22k	72.9 +7.2	79.0 +6.9	86.0 +12.9	65.9 +4.7	66.0 +6.4	67.3 +5.0
J1-Qwen-32B	22k	74.6 +8.1	82.2 +6.6	93.3 +8.1	65.2 +11.6	65.3 +9.4	66.8 +4.8
J1-Qwen-32B-MultiTask	22k	**76.8** +10.3	**85.0** +9.4	**94.3** +9.1	**68.6** +15.0	66.3 +10.4	**69.5** +7.5

Multitask-J1 outperforms pairwise and pointwise

Models	Eval	(a,b) Acc ↑	(b,a) Acc ↑	Consistent Acc ↑	Verdict Flip/Ties ↓
J1-Qwen-32B-Pairwise	Pairwise	74.6	74.2	65.2	14.5
J1-Qwen-32B-Pointwise	Pointwise	–	–	69.3	13.0
J1-Qwen-32B-MultiTask	Pairwise	76.8	76.2	67.0	17.0
J1-Qwen-32B-MultiTask	Pointwise	–	–	70.6	10.5

Score distribution of point and pairwise setups (PPE)



- J1 achieves state-of-the-art performance on the PPE Correctness dataset
- Multitask- J1 outperforms pairwise and pointwise performance, demonstrate superior accuracy on both random order and consistency
- Pairwise Eval has sparser scores distrubution and bigger score gaps

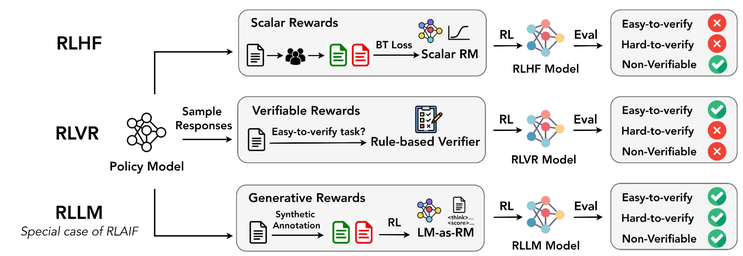
Results on five RM/Judge Benchmarks

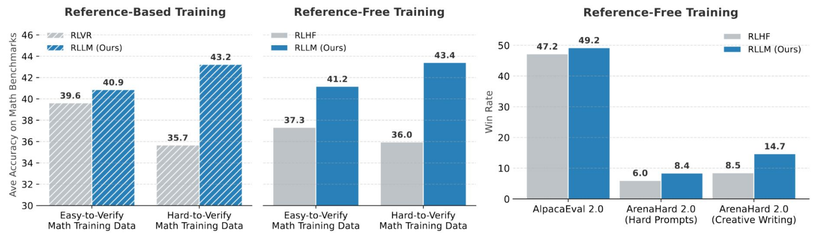
- J1 achieves best result among the same model family and size.
- **J1-Qwen-32B-Multitask** outperforms R1, o3

Models	Overall	PPE	Reward-Bench	RM-Bench	Judge-Bench	Follow-Bench
Llama-3.3-70B-Instruct	64.3	65.8	85.4	69.5	48.6	52.2
R1-Distilled-Llama-70B	67.4	68.6	86.9	80.8	46.0	54.6
EvalPlanner-Llama-70B	73.2	67.9	93.8	82.1	56.6	65.4
J1-Llama-70B	75.0 +10.7	69.6 +3.8	93.3 +7.9	82.7 +13.2	60.0 +11.4	69.3 +17.1
Qwen3-32B	77.3	66.5	90.9	88.1	70.8	70.0
J1-Qwen-32B-MultiTask	80.8 +3.5	71.8 +5.1	93.6 +2.7	90.3 +2.2	71.4 +0.6	77.1 +7.1
OpenAI-o1-mini	72.7	68.5	87.1	80.8	64.2	62.9
OpenAI-o3	77.4	72.1	86.4	86.1	75.7	66.8
DeepSeek-R1-671B	78.4	72.3	90.6	88.6	68.9	71.7

Our Follow-up Work: J1-as-RM (RLLM)

- **RLLM**: We train J1 as on-policy Generative Reward Model
- RLLM is best on easy-/hard-to verify and non-verifibale tasks,





(a) Verifiable Tasks (Math) **(b)** Non-Verifiable Tasks