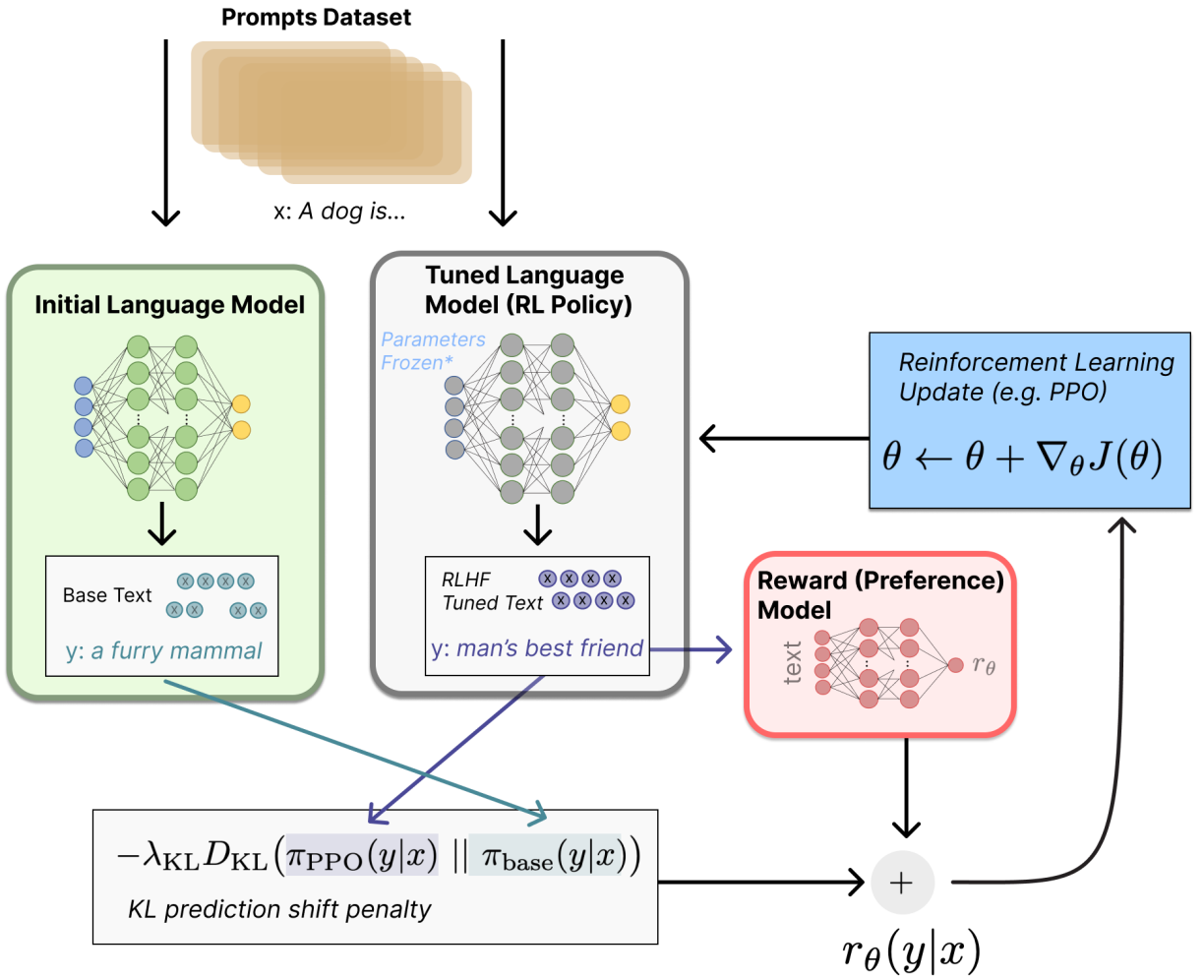
RM-R1: Reward Modeling as Reasoning

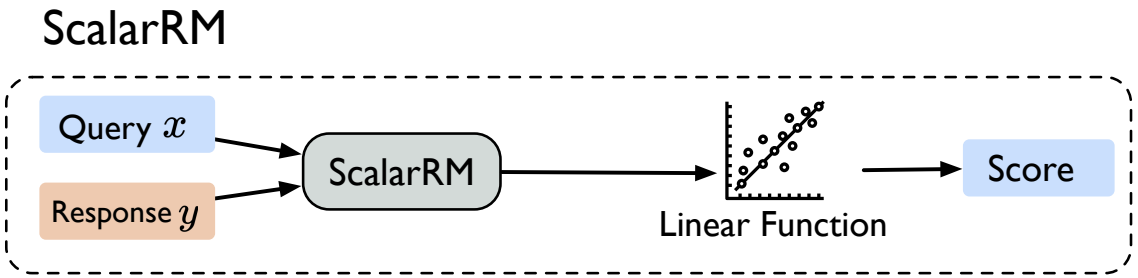
Xiusi Chen*, Gaotang Li*, Ziqi Wang*,
Bowen Jin, Cheng Qian, Yu Wang, Hongru Wang, Yu Zhang, Denghui Zhang,
Tong Zhang, Hanghang Tong, Heng Ji

Reinforcement Learning with Human Feedback (RLHF)

- Passing the fine-grained feedback learned from the reward model to the supervised fine-tuned language model
- Yields the final model that generates even better response
- RLHF is widely used in preference/trustworthy/safety alignment



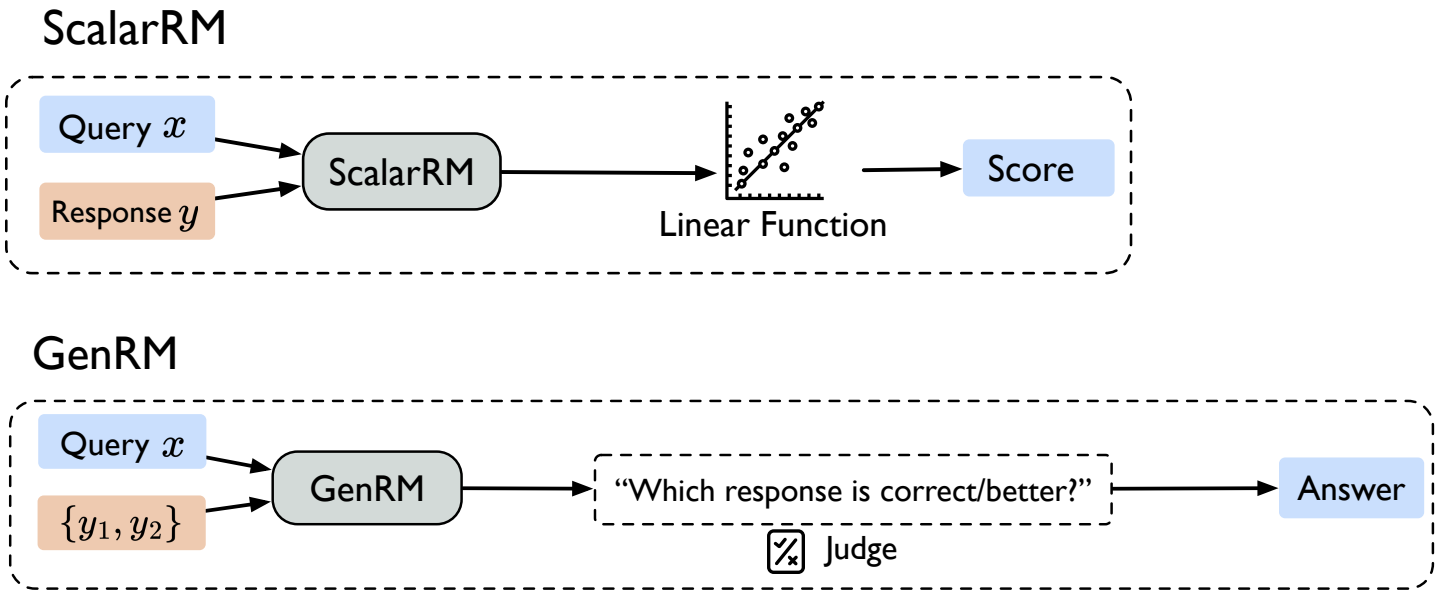
Reward Modeling as *Regression*



- ScalarRMs map each model generated response to a scalar score.

Chen et al., "RM-R1: Reward Modeling as Reasoning." ICLR 2026.

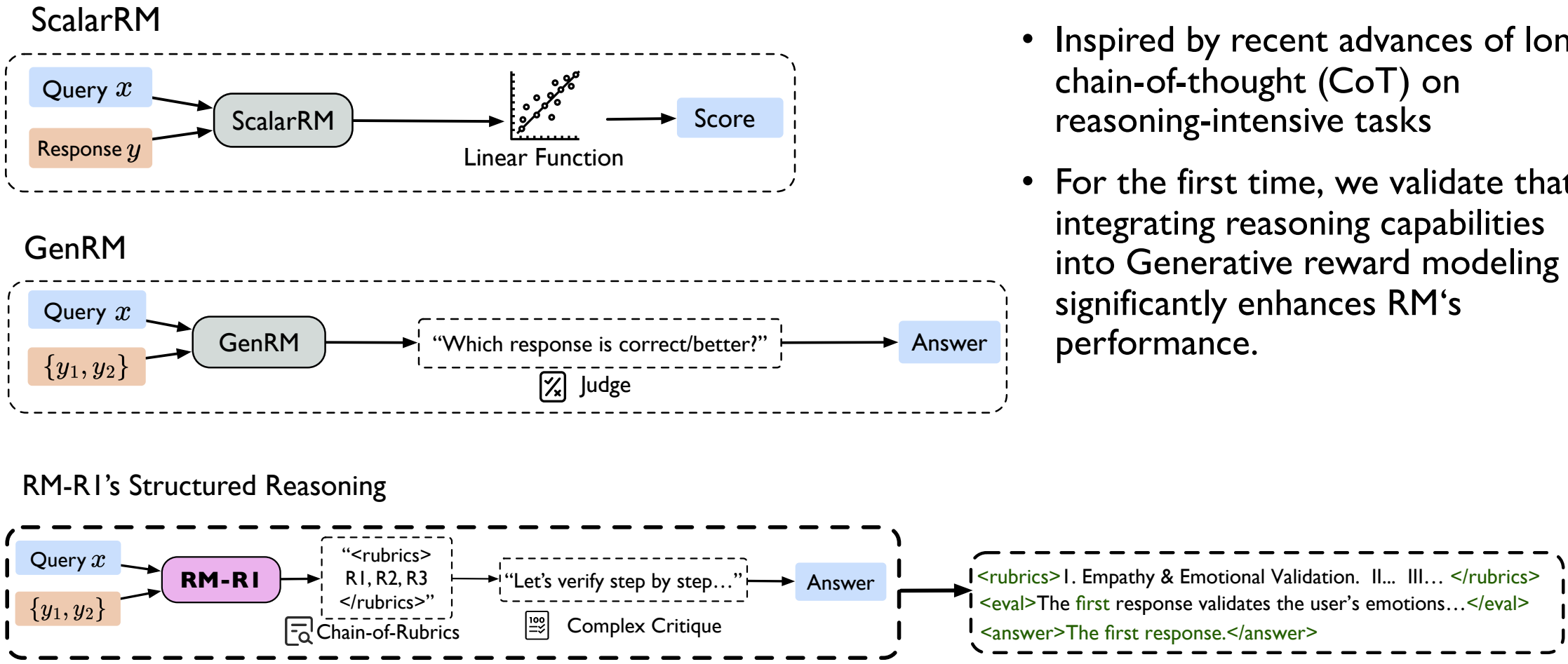
Reward Modeling as *Generation*



- ScalarRMs map each model generated response to a scalar score.
- GenRMs leverages the generative capability of decoder-only models by instructing.

Chen et al., "RM-R1: Reward Modeling as Reasoning." ICLR 2026.

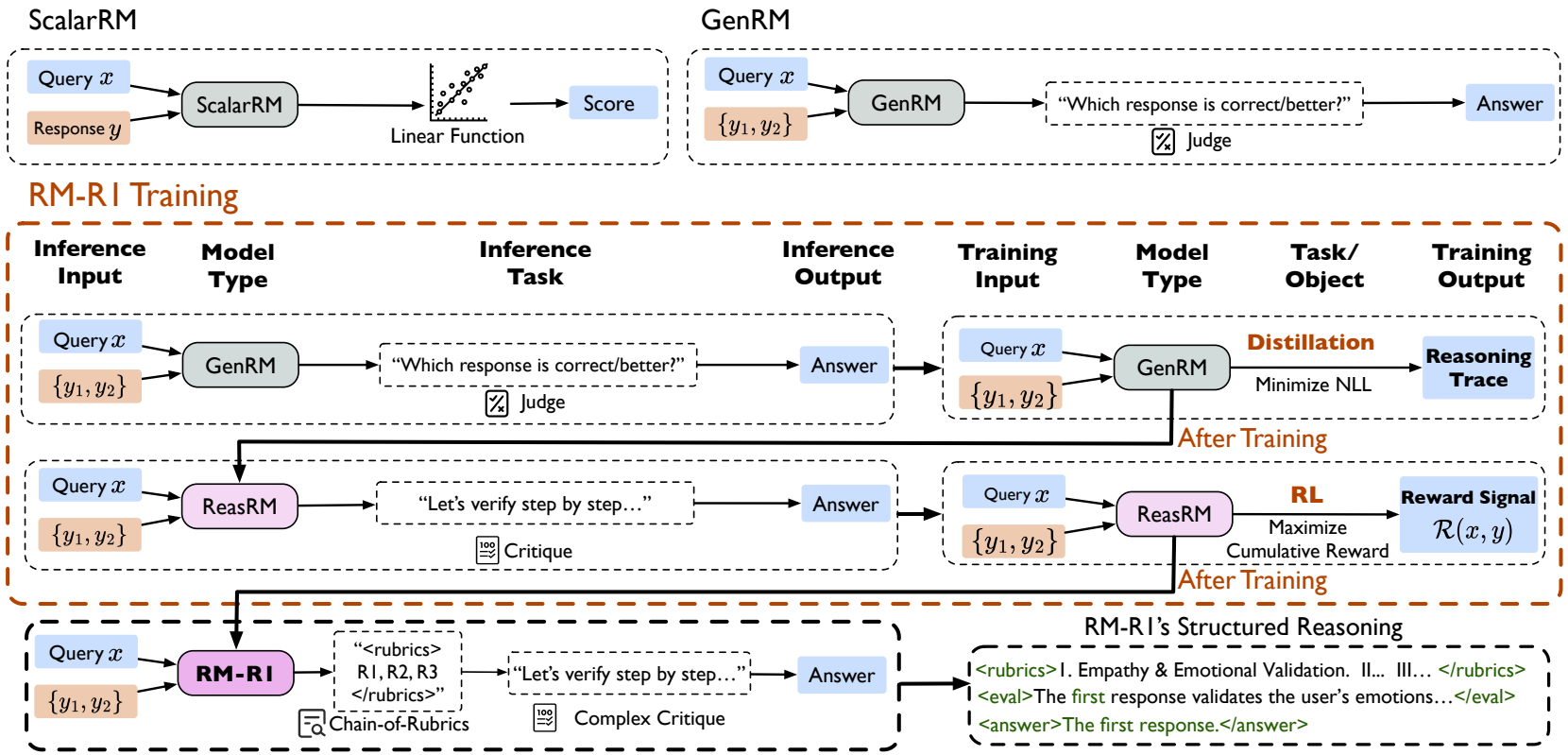
Reward Modeling as *Reasoning*



- Inspired by recent advances of long chain-of-thought (CoT) on reasoning-intensive tasks
- For the first time, we validate that integrating reasoning capabilities into Generative reward modeling significantly enhances RM's performance.

Chen et al., "RM-R1: Reward Modeling as Reasoning." ICLR 2026.

RM-R1: Training pipeline



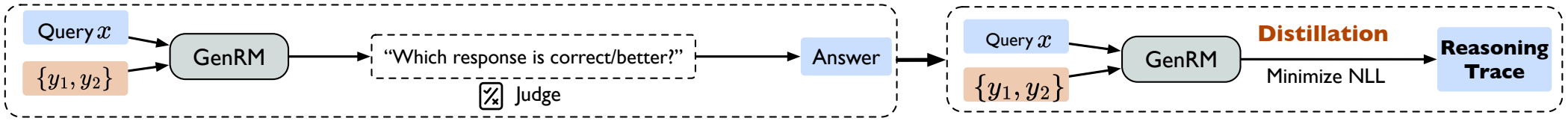
The training consists of two key stages:

(1) distillation of high-quality reasoning chains

(2) reinforcement learning with verifiable rewards.

Chen et al., "RM-R1: Reward Modeling as Reasoning." ICLR 2026.

RM-R1 Training Stage 1: Distillation

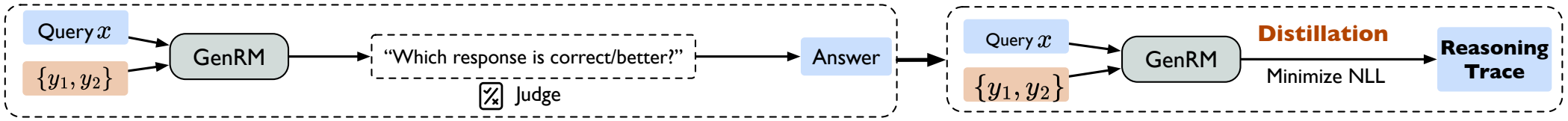


- Why distillation? Without fine-tuning on specialized reasoning traces, an off-the-shelf models may struggle to conduct consistent judgments.
- The Distillation process is resembles Imitation Learning
- We minimize the negative log-likelihood (NLL) loss:

$$\mathcal{L}_{\text{distill}}(\theta) = -\sum_{(x,y)\in\mathcal{D}_{\text{distill}}} \sum_{t\in[|y|]} \log r_\theta\left(y_t \mid x, y_{<t}\right)$$

Chen et al., "RM-R1: Reward Modeling as Reasoning." ICLR 2026.

Distillation Data Synthesis



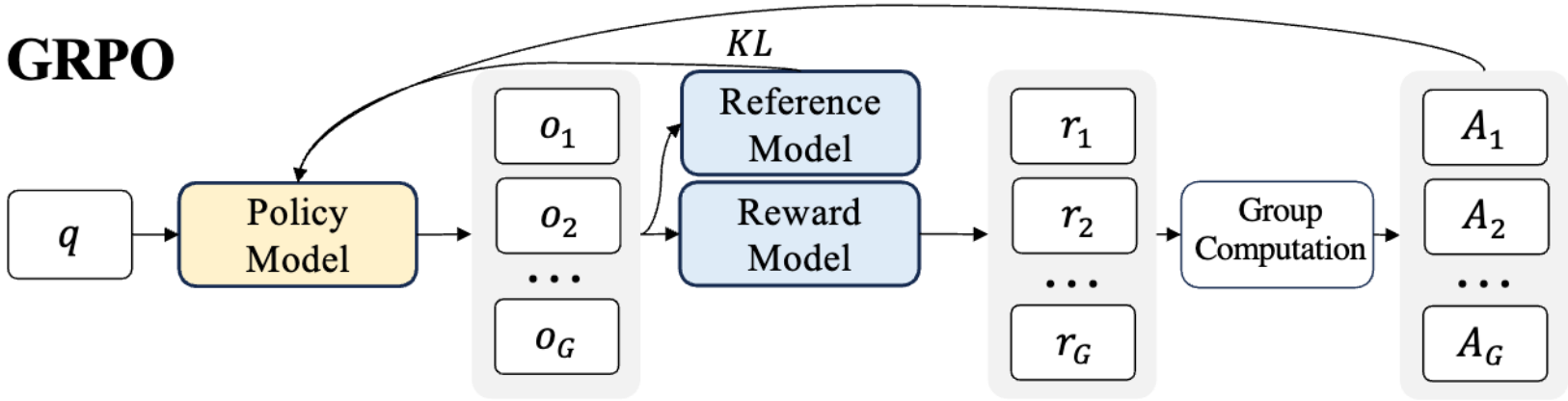
- Subsample from preference data $\mathcal{D}_{\text{sub}} \subset \mathcal{D}$
- For each $(x^{(i)}, y_a^{(i)}, y_b^{(i)}, l^{(i)}) \in \mathcal{D}_{\text{sub}}$, generate reasoning trace (rationales) $r^{(i)}$
- Construct Distillation data

$$y_{\text{trace}}^{(i)} = r^{(i)} \oplus l^{(i)}$$

$$\mathcal{D}_{\text{distill}} = \{(x^{(i)}, y_{\text{trace}}^{(i)})\}_{i=1}^{M}$$

Chen et al., "RM-R1: Reward Modeling as Reasoning." ICLR 2026.

RM-R1 Training Stage 2: Reinforcement learning



- The training consists of two key stages:
  - (1) distillation of high-quality reasoning chains
  - (2) reinforcement learning with verifiable rewards.
- Why RL?
  - Sole distillation often suffers from overfitting to certain patterns in the offline data
  - Constrains the model's ability to generalize its reasoning abilities for critical thinking
  - RL is known for better generalization

Chen et al., "RM-R1: Reward Modeling as Reasoning." ICLR 2026.

Chain-of-Rubrics Rollout

- Chain-of-Rubrics (CoR) enables the model to self-generate grading rubrics before thinking
- Splits ***Chat*** and ***Reasoning*** types of questions
  - ***Chat***: the model generates a set of evaluation rubrics
  - ***Reasoning***: the model solves the problem itself, and use its own solution as the rubric
- Evaluate the responses and give judgement

Chain-of-Rubrics (CoR) Rollout for Instruct Models

Please act as an impartial judge and evaluate the quality of the responses provided by two AI Chatbots to the Client's question displayed below.

First, classify the task into one of two categories: <type> Reasoning </type> or <type> Chat </type>.
- Use <type> Reasoning </type> for tasks that involve math, coding, or require domain knowledge, multi-step inference, logical deduction, or combining information to reach a conclusion.
- Use <type> Chat </type> for tasks that involve open-ended or factual conversation, stylistic rewrites, safety questions, or general helpfulness requests without deep reasoning.

If the task is Reasoning:
1. Solve the Client's question yourself and present your final answer within <solution> ... </solution> tags.
2. Evaluate the two Chatbot responses based on correctness, completeness, and reasoning quality, referencing your own solution.
3. Include your evaluation inside <eval> ... </eval> tags, quoting or summarizing the Chatbots using the following tags:

- <quote_A> ... </quote_A> for direct quotes from Chatbot A
- <summary_A> ... </summary_A> for paraphrases of Chatbot A
- <quote_B> ... </quote_B> for direct quotes from Chatbot B
- <summary_B> ... </summary_B> for paraphrases of Chatbot B

4. End with your final judgment in the format: <answer>[[A]]</answer> or <answer>[[B]]</answer>

If the task is Chat:
1. Generate evaluation criteria (rubric) tailored to the Client's question and context, enclosed in <rubric>...</rubric> tags.
2. Assign weights to each rubric item based on their relative importance.
3. Inside <rubric>, include a <justify>...</justify> section explaining why you chose those rubric criteria and weights.
4. Compare both Chatbot responses according to the rubric.
5. Provide your evaluation inside <eval>...</eval> tags, using <quote_A>, <summary_A>, <quote_B>, and <summary_B> as described above.
6. End with your final judgment in the format: <answer>[[A]]</answer> or <answer>[[B]]</answer>

Important Notes:
- Be objective and base your evaluation only on the content of the responses.
- Do not let response order, length, or Chatbot names affect your judgment.
- Follow the response format strictly depending on the task type.

Chen et al., "RM-R1: Reward Modeling as Reasoning." ICLR 2026.

Single Correctness Reward is Enough

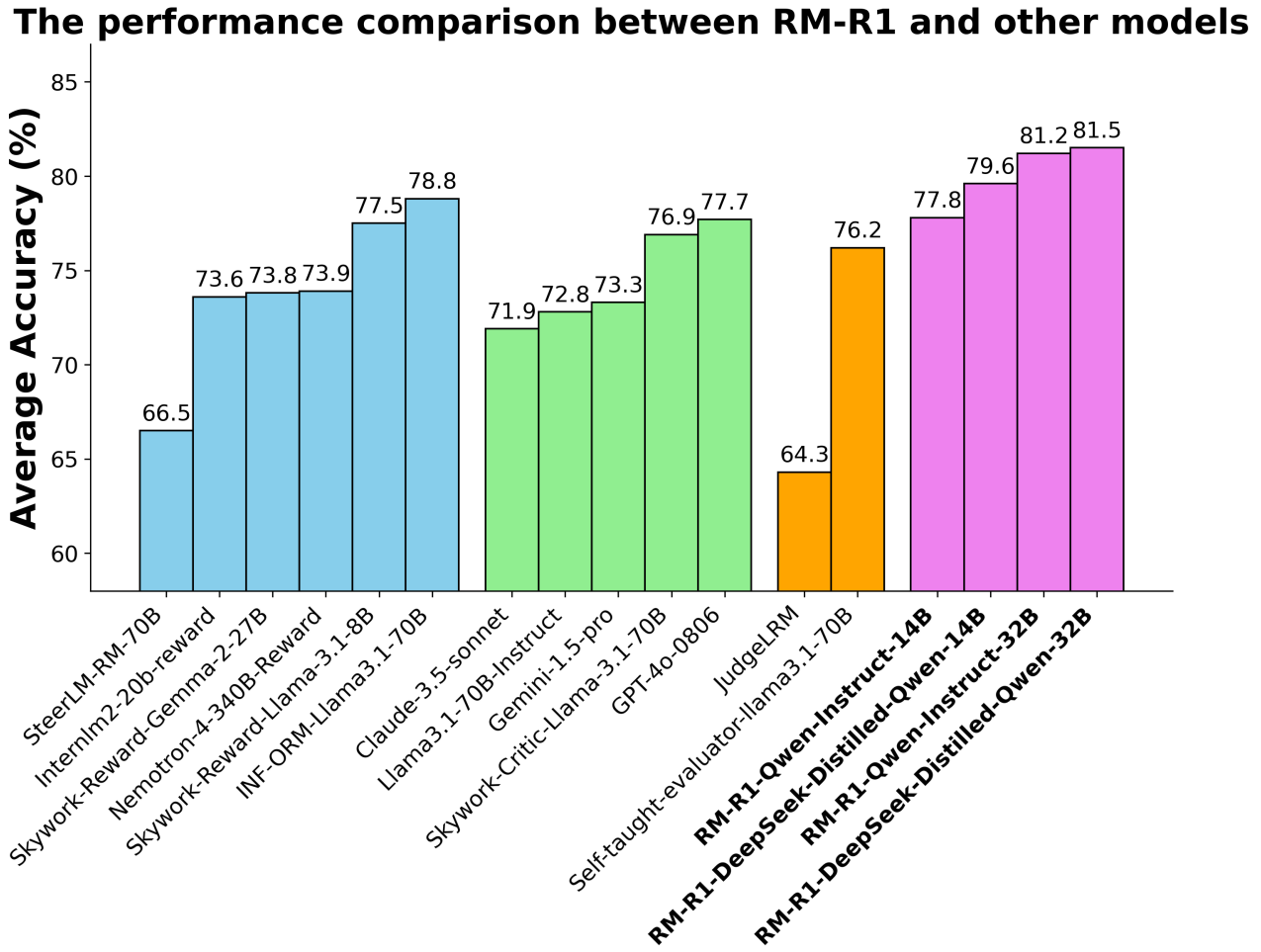
$$\mathcal{R}(x, j|y_a, y_b) = \begin{cases} 1 & \text{if } \hat{l} = l, \\ -1 & \text{otherwise.} \end{cases}$$

- Rule-based reward has demonstrated by DeepSeek-R1 to be effective for stimulating reasoning
- We mainly focus on correctness and omit others like format rewards
  - The distilled models have already learned to follow instructions and formatting.
- Use GRPO/PPO to train RM-R1.

Chen et al., "RM-R1: Reward Modeling as Reasoning." ICLR 2026.

Rubric-based Reasoning Improves Reward Accuracy

- Empirical results show that RM-R1 achieves sota or near sota performance of generative RMs on RewardBench, RM-Bench and RMB, outperforming much larger open-weight models (e.g., Llama3.1-405B) and proprietary ones (e.g., GPT-4o) by up to 4.9%.



Chen et al., "RM-R1: Reward Modeling as Reasoning." ICLR 2026.

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Thank you!

Xiusi Chen

University of Illinois at Urbana-Champaign