ICLR

Linking Process to Outcome: Conditional Reward Modeling for LLM Reasoning

Zheng Zhang, Ziwei Shan, Kaitao Song, Yexin Li, Kan Ren
ShanghaiTech University
State Key Laboratory of General Artificial Intelligence, BIGAI

北京通用人工智能研究院
Beijing Institute for General Artificial Intelligence

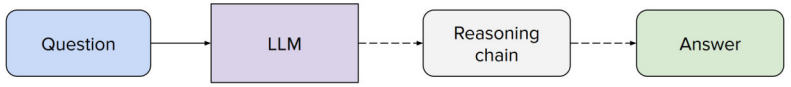
上海科技大学
ShanghaiTech University

Background: LLM Reasoning

Step-by-Step Reasoning

- LLM derive final answers through explicit step-by-step reasoning.

$$\text{Input} \rightarrow \underbrace{\text{Step}_1 \rightarrow \text{Step}_2 \rightarrow \cdots \rightarrow \text{Step}_n}_{\text{Reasoning chain}} \rightarrow \text{Answer}$$



Verifiable Reward

- Checking model outputs against ground-truth.
- Rely on ground-truth, whose acquisition is costly.
- Sparse rewards, lacking fine-grained supervision.

Problem 1:
Question: Two trains running in opposite directions cross a man standing on the platform in 27 seconds and 17 seconds respectively and they cross each other in 23 seconds. The ratio of their speeds is:
Options: A) 3/7 B) 3/2 C) 3/88 D) 3/8 E) 2/2
Rationale: Let the speeds of the two trains be x m/sec and y m/sec respectively. Then, length of the first train = 27x meters, and length of the second train = 17 y meters. (27x + 17y) / (x + y) = 23 → 27x + 17y = 23x + 23y → 4x = 6y → x/y = 3/2.
Correct Option: B

Process Reward Models

- Provide granular feedback at individual steps.

Limitations of Existing PRMs

Isolated step modeling

- Each step evaluated independently.
- Ignores sequential dependency

Limited outcome awareness

- Process rewards not aligned with final correctness.

Ambiguous credit assignment

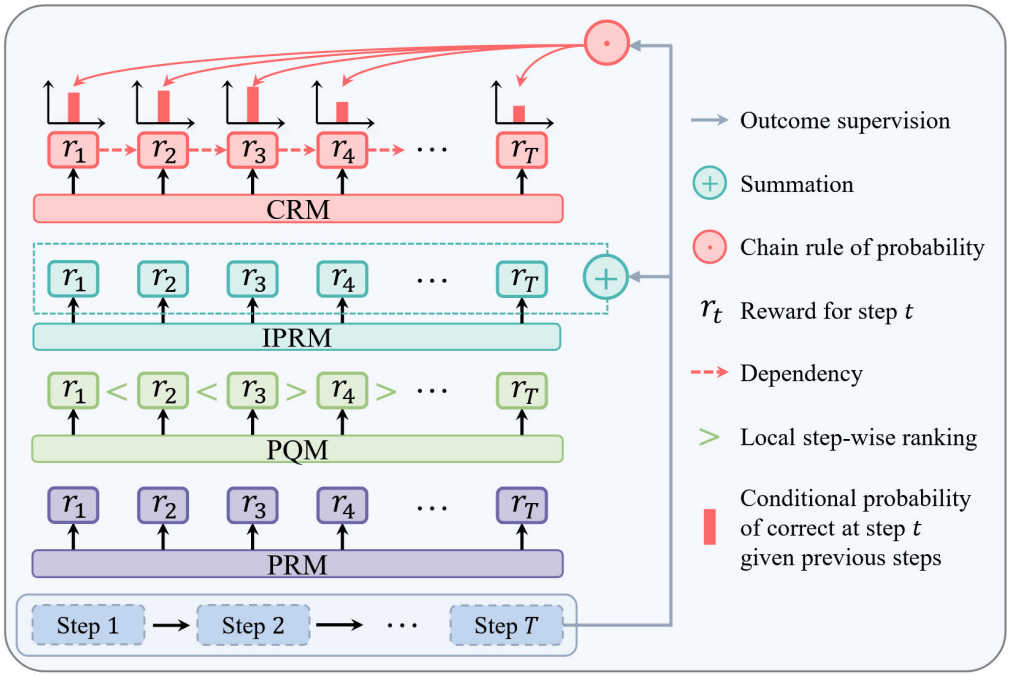
- Imprecise rewards make optimization prone to reward hacking.

Conditional Reward Modeling

ICLR

Key idea

- Framing reasoning as a temporal process through which an LLM progressively approaches the correct final answer.
- Condition each step on previous steps.
- Explicitly link each step to final outcome.



Conditional Reward Modeling

ICLR

Defining the Wrong State

- Reasoning trajectory can no longer yield the correct answer.
- z is the index of the first wrong step.
- $p(z)$ denotes the probability of a wrong state occurring at step z.
- $W(t)$ denotes the probability that a wrong state has already occurred at or before step t.
- $S(t)$ denotes the probability of maintaining correct reasoning up to step t.

$$W(t) = \Pr(z \leq t) = \sum_{z=1}^{t} p(z)$$

$$S(t) = \Pr(z > t) = 1 - W(t) = \sum_{z=t+1}^{\infty} p(z)$$

Conditional Reward Modeling

ICLR

Conditional Dependency

- $h(t)$ denotes the probability that step t enters a wrong state, given all previous steps were correct.

$$h(t) = \Pr(z = t | z \geq t) = \frac{\Pr(z = t)}{\Pr(z \geq t)} = \frac{p(t)}{S(t-1)}$$

Chain rule of probability

- Establish relationships among the variables.

$$S(t) = \Pr(z > t) = \prod_{k=1}^{t} (1 - h(k))$$

$$W(t) = \Pr(z \leq t) = 1 - S(t) = 1 - \prod_{k=1}^{t} (1 - h(k))$$

Conditional Reward Modeling

ICLR

Obtaining Process Rewards

- Applying Potential-Based Reward Shaping (PBRS) yields a dense, step-wise reward mathematically tied to the outcome:

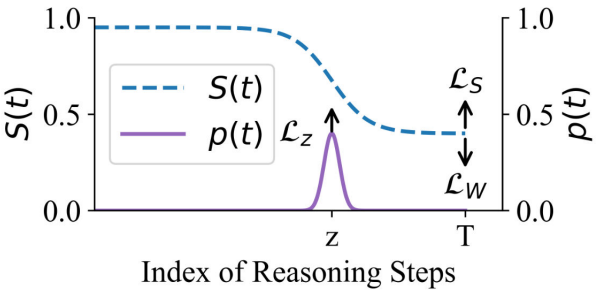
$$r_t = \log(1 - h(t))$$

- The final correct outcome probability $S(T)$ (where T is the total number of steps) equals the product of exponentiated step-wise rewards.

$$S(T) = \prod_{t=1}^{T}(1 - h(t)) = \prod_{t=1}^{T} e^{r_t}$$

- Explicitly modeling the relationship between process rewards and outcome.
- Resolving credit assignment ambiguity.

Conditional Reward Modeling



Training Objectives

- Conditional reward model (CRM) is trained to predict $h(t)$.

- **Maximize Success**: For correct trajectories, we maximizes the probability of reaching the correct answer i.e. $S(T)$.

$$\mathcal{L}_S(x_i, y_i) = -\log \Pr(z_i > T) = -\log S(T) = -\log \left[\prod_{t=1}^{T} (1 - h(t)) \right]$$

- **Minimize Failures**: For flawed trajectories, we minimize $S(T)$.

$$\mathcal{L}_W(x_i, y_i) = -\log \Pr(z_i \leq T) = -\log(1 - S(T)) = -\log \left[1 - \prod_{t=1}^{T} (1 - h(t)) \right]$$

- **Pinpoint Errors**: Maximizes the probability of the wrong state occurring at step z.

$$\mathcal{L}_z(x_i, y_i, z_i) = -\log \Pr(z_i) = -\log p(z_i) = -\log \left[h(z_i) \prod_{t=1}^{z_i - 1} (1 - h(t)) \right]$$

Experiments

Best-of-N Sampling

Best-of-N sampling generates N responses for a given question and selects the optimal one using a reward model.

Table 1: Best-of-N accuracy across models. **Bold** and underlined values denote the top two results.

Models	Methods	GSM-Plus					MATH500				
		@8	@16	@32	@64	@128	@8	@16	@32	@64	@128
	Major@N	66.3	65.6	64.7	64.2	64.2	46.2	46.0	45.2	45.0	44.8
	Pass@N (Oracle)	79.2	81.2	82.2	83.3	85.5	73.8	79.2	84.6	87.4	90.2
Qwen2.5-3B-Instruct	ORM	66.8	67.2	66.4	65.7	65.7	51.6	51.4	51.8	49.0	49.2
	PRM	67.6	67.9	67.7	66.9	66.7	**54.2**	55.2	55.2	54.2	54.6
	PQM	**68.5**	**69.2**	**68.5**	68.2	68.0	53.2	54.4	54.8	54.8	55.8
	IPRM	65.5	66.2	66.8	66.5	66.2	52.4	52.0	52.0	52.2	53.0
	CRM (ours)	67.8	68.6	67.9	**68.4**	**68.7**	53.0	**56.4**	**56.6**	**55.8**	**56.6**
LLaMA3.1-8B	ORM	66.9	67.4	67.1	67.2	66.6	47.4	46.4	44.6	45.2	45.6
	PRM	**67.9**	68.2	68.5	**68.8**	**68.9**	48.0	48.0	49.8	49.0	47.6
	PQM	66.4	67.0	66.2	66.5	67.2	**51.0**	**51.4**	48.8	49.0	48.4
	IPRM	65.1	64.3	63.9	63.5	63.7	48.4	45.8	44.2	46.0	45.8
	CRM (ours)	67.8	**68.8**	**69.1**	68.6	68.5	49.4	50.6	**50.6**	**49.8**	**50.6**

Key Finding

- CRM demonstrates stronger trajectory-level selection in Best-of-N sampling.

Experiments

Beam Search

For each question, beam search initiates by sampling $\mathbf{N}$ responses. Subsequently, a beam of $\mathbf{b}$ candidates with the highest rewards is maintained and expanded during the generation process.

Table 2: Beam Search accuracy on MATH500 and Gaokao2023.

Models	Methods	MATH500				GAOKAO2023			
		$N=4$	$N=8$	$N=20$	$N=100$	$N=4$	$N=8$	$N=20$	$N=100$
Qwen2.5-Math-1.5B	ORM	50.73	54.80	56.80	58.07	35.58	38.18	38.44	40.17
	PRM	51.80	55.73	56.87	58.00	34.72	37.84	38.70	38.96
	PQM	52.67	56.60	58.87	58.80	36.88	38.61	40.61	39.83
	IPRM	44.27	47.27	48.33	47.47	32.55	34.46	35.32	34.55
	CRM (ours)	**54.07**	**58.40**	**61.00**	**63.00**	**38.70**	**39.74**	**41.04**	**43.55**
Qwen2.5-Math-7B	ORM	51.87	57.67	59.47	60.73	37.49	40.26	44.07	43.72
	PRM	52.13	55.67	59.93	60.13	37.58	40.52	41.04	43.81
	PQM	52.73	57.87	59.20	61.13	37.84	40.61	42.60	43.29
	IPRM	49.53	54.07	54.20	52.60	35.67	38.53	40.26	39.57
	CRM (ours)	**56.07**	**60.60**	**62.87**	**64.07**	**39.83**	**42.77**	**46.49**	**48.40**
Llama3.1-8B	ORM	38.13	38.80	38.93	36.67	25.63	26.93	**29.35**	27.62
	PRM	37.87	39.67	40.13	39.53	26.84	**28.66**	27.97	27.97
	PQM	39.20	40.47	41.00	**41.27**	26.58	27.71	28.31	28.23
	IPRM	37.07	38.67	37.13	34.13	26.49	25.89	26.15	24.42
	CRM (ours)	**40.20**	**41.00**	**42.07**	41.00	**26.93**	28.40	28.74	**29.96**

Key Finding

- CRM provides effective and consistent step-level guidance for beam search.

Experiments

RL Optimization

The reward model provides step-level dense rewards to guide the policy model toward generating improved reasoning trajectories.

Table 3: Pass@1 accuracy evaluated on six mathematical reasoning benchmarks.

VR from outcome ground-truth	Method	MATH 500	Minerva Math	Olympiad Bench	AIME25	AIME24	AMC23
VR Disabled	PURE	76.0	30.8	36.7	13.3	26.6	**70.0**
	PRM	71.6	36.3	32.5	13.3	10.0	57.5
	PQM	72.0	34.1	34.3	13.3	13.3	52.5
	CRM (ours)	**77.8**	**40.0**	**39.3**	**23.3**	**43.3**	67.5
VR Enabled	Prime	81.2	29.4	40.8	16.6	26.6	**72.5**
	PURE	**82.4**	40.0	41.3	23.3	23.3	70.0
	VR	76.2	38.6	38.0	16.6	30.0	62.5
	CRM + VR	80.4	**43.0**	**42.1**	**26.6**	**33.3**	**72.5**

Key Finding

- CRM boosts RL performance without VR (Verifiable Reward).

Take-away

Main takeaway messages

- CRM frames LLM reasoning as a temporal probabilistic process.
- CRM explicitly models the causal dependencies between steps and links process to the outcome.
- CRM is robust to reward hacking.
- Experiments across Best-of-N sampling, beam search, and RL demonstrate that CRM consistently outperforms strong baselines.

https://foundation-model-research.github.io/CRM/