

# Let's Verify Step by Step

(Hunter Lightman et al., 2023)

melon讨论班 陈楚岩

2025.05.21

# Let's Verify Step by Step

**Hunter Lightman\***

**Vineet Kosaraju\***

**Yura Burda\***

**Harri Edwards**

**Bowen Baker**

**Teddy Lee**

**Jan Leike**

**John Schulman**

**Ilya Sutskever**

**Karl Cobbe\***

**OpenAI**

# PART 01



## Introduction

# Review of PPO, DPO and GRPO

- **Goal:** Optimize a language model to **adhere to human preferences**.

- **PPO:** 
$$\mathcal{L}_{\text{PPO}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_{\theta_{\text{old}}}} \frac{1}{|y|} \sum_{t=1}^{|y|} \frac{\pi_{\theta}(y_t | x, y_{<t})}{\pi_{\theta_{\text{old}}}(y_t | x, y_{<t})} \boxed{A_t} \text{r}(\cdot) \text{V}_{\phi}(\cdot)$$

- **DPO:** 
$$\mathcal{L}_{\text{DPO}}(\pi_{\theta}; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[ \log \sigma \left( \beta \log \frac{\pi_{\theta}(y_w | x)}{\pi_{\text{ref}}(y_w | x)} - \beta \log \frac{\pi_{\theta}(y_l | x)}{\pi_{\text{ref}}(y_l | x)} \right) \right]$$

- **GRPO:** 
$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}, \{y_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(x)} \left[ \frac{1}{G} \sum_{i=1}^G \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \left( \frac{\pi_{\theta}(y_{i,t} | x_i, y_{i,<t})}{\pi_{\text{old}}(y_{i,t} | x_i, y_{i,<t})} \boxed{\hat{A}_{i,t}} \text{r}(\cdot) - \beta \mathbb{D}_{\text{KL}}(\pi_{\theta} || \pi_{\text{ref}}) \right) \right]$$

Method	Reference Model $\pi_{\text{ref}}(\cdot)$	Old Model $\pi_{\text{old}}(\cdot)$	Policy Model $\pi_{\theta}(\cdot)$	Reward Model $\text{r}(\cdot)$	Value Function $\text{V}_{\phi}(\cdot)$
PPO	Frozen	Frozen	Trainable	Frozen	Trainable
DPO	Frozen	×	Trainable	×	×
GRPO	Frozen	Frozen	Trainable	Frozen	×

# A unified paradigm for different training methods




- Three core components for existing algorithms
- We focus on how to get a good **reward model** as a reward function.




Methods	Data Source	Reward Function	Gradient Coefficient
SFT	$q, o \sim P_{sft}(Q, O)$	-	1
RFT	$q \sim P_{sft}(Q), o \sim \pi_{sft}(O q)$	Rule	Equation 10
DPO	$q \sim P_{sft}(Q), o^+, o^- \sim \pi_{sft}(O q)$	Rule	Equation 14
Online RFT	$q \sim P_{sft}(Q), o \sim \pi_{\theta}(O q)$	Rule	Equation 10
PPO	$q \sim P_{sft}(Q), o \sim \pi_{\theta}(O q)$	Model	Equation 18
GRPO	$q \sim P_{sft}(Q), \{o_i\}_{i=1}^G \sim \pi_{\theta}(O q)$	Model	Equation 21

# From ORM to PRM




- Outcome-supervised **R**eward **M**odel (ORM): feedback only on **final result**
- Process-supervised **R**eward **M**odel (PRM): feedback on **each reasoning step**


The denominator of a fraction is 7 less than 3 times the numerator. If the fraction is equivalent to  $2/5$ , what is the numerator of the fraction? (Answer: )

   Let's call the numerator  $x$ .

   So the denominator is  $3x-7$ .

   We know that  $x/(3x-7) = 2/5$ .

   So  $5x = 2(3x-7)$ .

    $5x = 6x - 14$ .

   So  $x = 7$ .

- **Outcome-supervised Reward Model (ORM):** feedback only on **final result**
- **Process-supervised Reward Model (PRM):** feedback on **each reasoning step**
- **Limitation of ORM**
  - **Lack of Stepwise Feedback:** ORMs only learn from the final outcome, leading to coarse and ambiguous training signals.
  - **Misaligned Reasoning Paths:** Without process-level supervision, models may exploit incorrect reasoning to arrive at correct answers, reinforcing undesirable behaviors.
  - **Limited Interpretability and Alignment:** ORMs provide no insight into decision steps, making it harder for humans to audit reasoning and ensure adherence to chains of thought.



- Outcome-supervised **R**eward **M**odel (ORM): feedback only on **final result**
- Process-supervised **R**eward **M**odel (PRM): feedback on **each reasoning step**
- **What did this paper do?**

Despite these advantages, Uesato et al. (2022) found that outcome supervision and process supervision led to similar final performance in the domain of grade school math. We conduct our own detailed comparison of outcome and process supervision, with three main differences: we use a more capable base model, we use significantly more human feedback, and we train and test on the more challenging MATH dataset (Hendrycks et al., 2021).



# PART 02

---




## Methods: PRM in Practice




- **Goal:** Train the most reliable reward model possible.
- **Base Model:** Fine-tuned GPT-style model (e.g. GPT-4).
- **Generator:** **Base Model** trained for step-by-step solutions.
- **Dataset:** PRM800K, 800K step-level labels across 75K solutions.
- **Evaluation:**
  - We evaluate a reward model by its ability to perform **best-of-N search** over uniformly sampled solutions from the generator.
  - For each test problem, we **select the solution** ranked highest by the reward model, automatically grade it **based on its final answer**, and report the fraction that are correct. A reward model that is more reliable will select the correct solution more often.

# Methods: Training Label and Loss




- **ORM:** Determine correctness by automatically checking the final answer.
- **PRM:** Assign each step in the solution a label of positive, negative, or neutral.



The denominator of a fraction is 7 less than 3 times the numerator. If the fraction is equivalent to  $2/5$ , what is the numerator of the fraction? (Answer: )

   Let's call the numerator  $x$ .

   So the denominator is  $3x-7$ .

   We know that  $x/(3x-7) = 2/5$ .

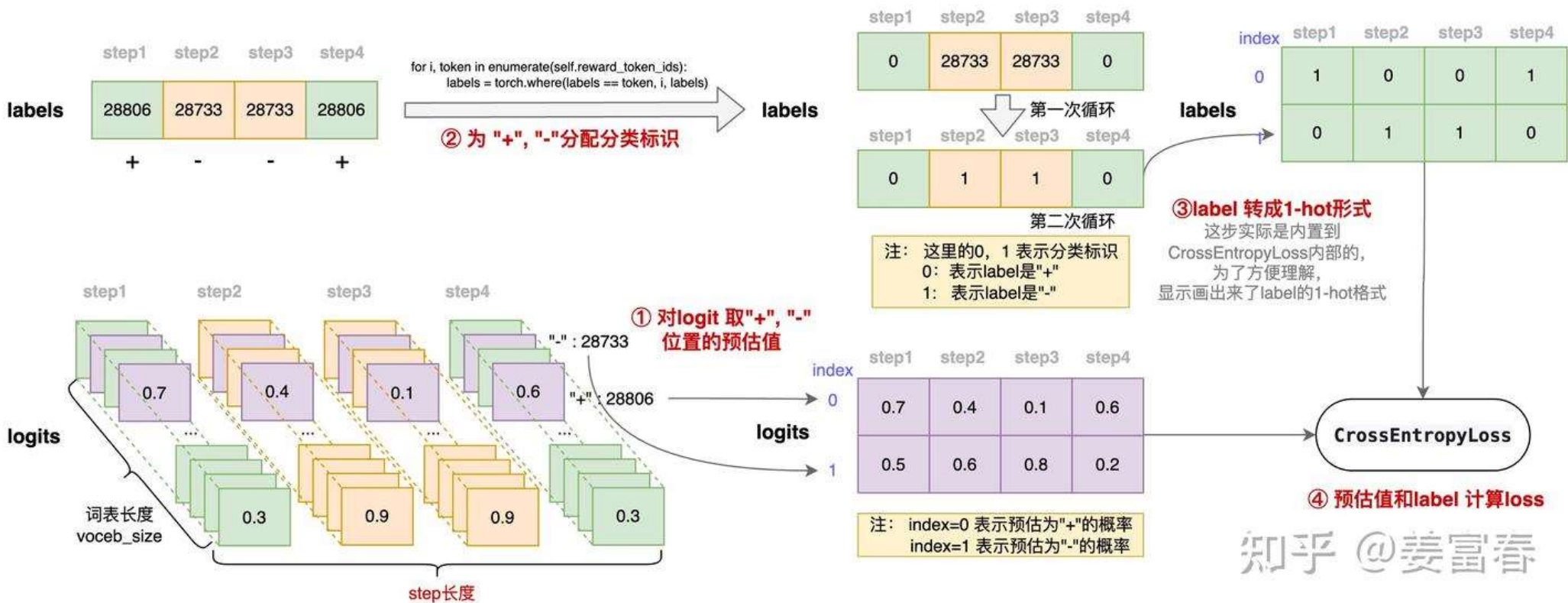
   So  $5x = 2(3x-7)$ .

    $5x = 6x - 14$ .

   So  $x = 7$ .

# Methods: Training Label and Loss

- **ORM**: Determine correctness by automatically checking the final answer.
- **PRM**: Assign each step in the solution a label of positive, negative, or neutral.



知乎 @姜富春

# Methods: Scoring Strategy

- **ORM:** Final token as the overall score for the solution.
- **PRM:** The product of the correctness probabilities for each step.

Let

$$x^8 + 3x^4 - 4 = p_1(x)p_2(x) \cdots p_k(x),$$

where each non-constant polynomial  $p_i(x)$  is monic with integer coefficients, and cannot be factored further over the integers. Compute  $p_1(1) + p_2(1) + \cdots + p_k(1)$ .

I notice that the given polynomial has even degree and only even powers of  $x$ , so I can try to make a substitution to simplify it.

Let  $y = x^4$ , then the polynomial becomes  $y^2 + 3y - 4$ , which is a quadratic equation.

I can factor this quadratic equation as  $(y + 4)(y - 1)$ , so the original polynomial is  $(x^4 + 4)(x^4 - 1)$ .

Now I need to factor each of these factors further, if possible, over the integers.

For the first factor,  $x^4 + 4$ , I recall the Sophie Germain identity, which says that  $a^4 + 4b^4 = (a^2 + 2b^2 + 2ab)(a^2 + 2b^2 - 2ab)$ .

Applying this identity with  $a = x$  and  $b = 1$ , I get  $x^4 + 4 = (x^2 + 2x + 2)(x^2 - 2x + 2)$ .

For the second factor,  $x^4 - 1$ , I can use the difference of squares formula, which says that  $a^2 - b^2 = (a + b)(a - b)$ .

Applying this formula with  $a = x^2$  and  $b = 1$ , I get  $x^4 - 1 = (x^2 + 1)(x^2 - 1)$ .

Then I can apply the difference of squares formula again with  $a = x$  and  $b = 1$ , to get  $x^2 - 1 = (x + 1)(x - 1)$ .

So the original polynomial is  $(x^4 + 4)(x^4 - 1) = (x^2 + 2x + 2)(x^2 - 2x + 2)(x^2 + 1)(x + 1)(x - 1)$ .

These are all the irreducible factors over the integers, so  $p_1(x) = x^2 + 2x + 2$ ,  $p_2(x) = x^2 - 2x + 2$ ,  $p_3(x) = x^2 + 1$ ,  $p_4(x) = x + 1$ , and  $p_5(x) = x - 1$ .

To find the sum of their values at  $x = 1$ , I just need to plug in  $x = 1$  into each factor and add them up.

I get  $p_1(1) = 1^2 + 2 \cdot 1 + 2 = 5$ ,  $p_2(1) = 1^2 - 2 \cdot 1 + 2 = 1$ ,  $p_3(1) = 1^2 + 1 = 2$ ,  $p_4(1) = 1 + 1 = 2$ , and  $p_5(1) = 1 - 1 = 0$ .

So the final answer is  $5 + 1 + 2 + 2 + 0 = 10$ .

Answer: 10

Let

$$x^8 + 3x^4 - 4 = p_1(x)p_2(x) \cdots p_k(x),$$

where each non-constant polynomial  $p_i(x)$  is monic with integer coefficients, and cannot be factored further over the integers. Compute  $p_1(1) + p_2(1) + \cdots + p_k(1)$ .

This problem is about factoring a polynomial over the integers, which can be a tricky task.

One possible strategy is to look for rational roots of the polynomial, using the rational root theorem.

However, that would involve checking a lot of possible factors of  $-4$ , and there is no guarantee that the polynomial has any rational roots at all.

Another possible strategy is to use some algebraic tricks to simplify the polynomial or reduce its degree.

I notice that the polynomial has only even powers of  $x$ , so I can try to substitute  $y = x^2$  and see what happens.

Then the polynomial becomes  $y^4 + 3y^2 - 4$ , which is a quadratic equation in  $y^2$ .

I can try to factor this quadratic equation by looking for two numbers that multiply to  $-4$  and add to  $3$ .

Those numbers are  $4$  and  $-1$ , so I can write  $y^4 + 3y^2 - 4 = (y^2 + 4)(y^2 - 1)$ .

Now I can substitute back  $x^2 = y$  and get  $x^8 + 3x^4 - 4 = (x^4 + 4)(x^4 - 1)$ .

This looks promising, because I have reduced the degree of the factors from  $8$  to  $4$ .

I can try to factor each of these factors further, using the difference of squares formula.

I get  $x^8 + 3x^4 - 4 = (x^2 + 2)(x^2 - 2)(x^2 + 1)(x^2 - 1)$ .

I can apply the difference of squares formula again to the last factor and get  $x^8 + 3x^4 - 4 = (x^2 + 2)(x^2 - 2)(x^2 + 1)(x + 1)(x - 1)$ .

Now I have factored the polynomial completely into monic linear and quadratic factors with integer coefficients.

These are the  $p_i(x)$ 's that the problem is asking for.

To find the sum of their values at  $x = 1$ , I just need to plug in  $x = 1$  into each factor and add them up.

I get  $p_1(1) + p_2(1) + \cdots + p_k(1) = (1^2 + 2)(1^2 - 2)(1^2 + 1)(1 + 1)(1 - 1)$ .

Simplifying, I get  $p_1(1) + p_2(1) + \cdots + p_k(1) = (3)(-1)(2)(2)(0)$ .

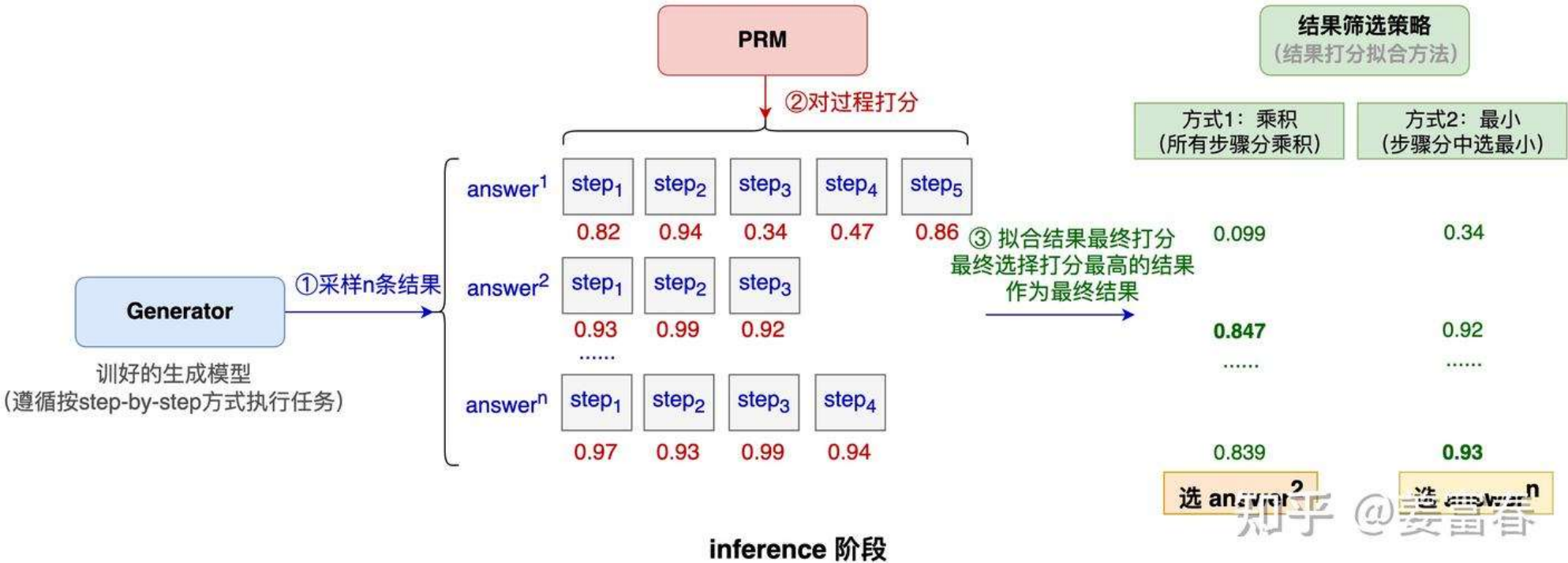
Multiplying, I get  $p_1(1) + p_2(1) + \cdots + p_k(1) = 0$ .

Answer: 0



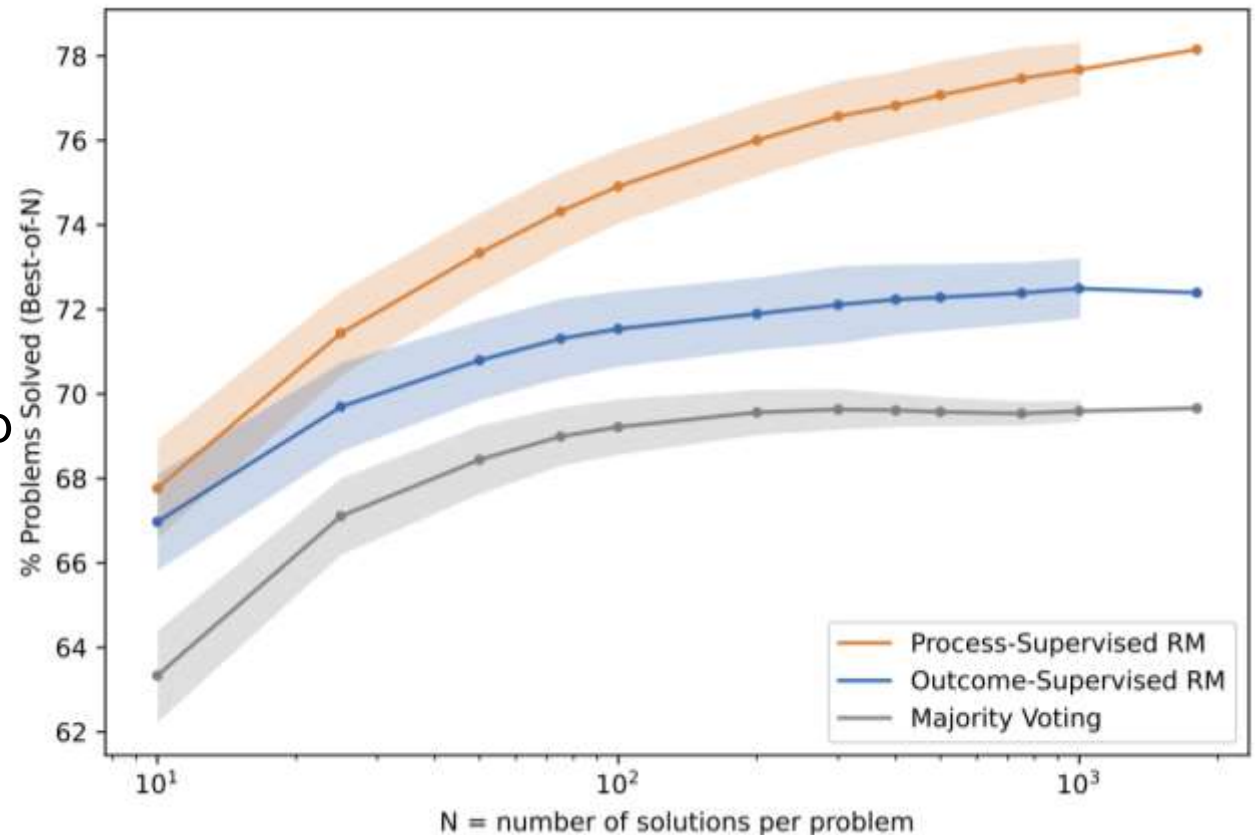
# Methods: Scoring Strategy

- **ORM**: Final token as the overall score for the solution.
- **PRM**: The product of the correctness probabilities for each step.



# Experiment: Large-Scale Supervision

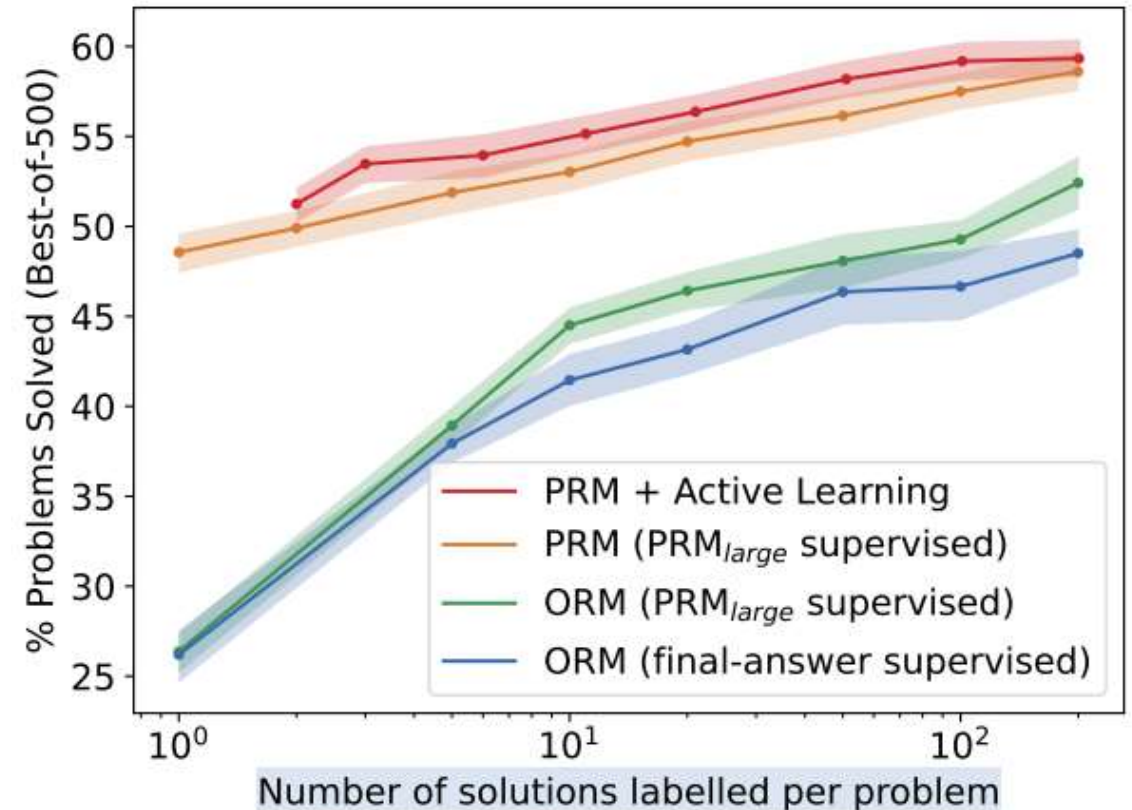
- **Majority Voting:** Sample  $N$  solutions and decide the answer with majority vote.
- **Variance:** Variance of subsamples of the 1860 solutions.
- Performance gap widens with larger  $N$  (up to 1860 samples)
- PRM outperforms across all difficulty quintiles





# Experiment: Small-Scale Synthetic Supervision

- **PRM\_large**: A newly trained large reward model used to annotate training data
- **Number of Solutions Labeled per Problem**:  
The count of annotations per problem produced by PRM\_large
- **Active Learning**:
  - Retrains a small reward model to identify error samples that confuse the small model and **improve data quality**
  - Achieves  $\sim 2.6\times$  data efficiency improvement
  - Focuses human effort on most informative samples



## Experiment: OOD Generalization

- PRM shows robust performance under distribution shift

	ORM	PRM	Majority Vote	# Problems
AP Calculus	68.9%	<b>86.7%</b>	80.0%	45
AP Chemistry	68.9%	<b>80.0%</b>	71.7%	60
AP Physics	77.8%	<b>86.7%</b>	82.2%	45
AMC10/12	49.1%	<b>53.2%</b>	32.8%	84
Aggregate	63.8%	<b>72.9%</b>	61.3%	234

## PART 03

---

# Discussion & Conclusion & Limitation

- **Credit Assignment:** Process supervision simplifies credit assignment by giving precise, **step-level feedback** instead of only a coarse final signal.
- **Alignment Impact:** By directly rewarding human-endorsed reasoning paths, process supervision boosts interpretability and safety **without imposing an alignment tax**.
- **Test Set Contamination:** Although some MATH problems may overlap with pretraining data, low solve-rates and uncontaminated generalization tests indicate minimal impact on evaluation.

- Process supervision yields significantly more reliable reward models.
- Active learning makes data collection efficient.
- Released PRM800K dataset to catalyze research.
- Difficulty in defining universally applicable **fine-grained reasoning steps**.
- Challenges in **judging the correctness** of each intermediate step—automated annotation is unreliable and manual labeling doesn't scale.
- Susceptibility to **reward hacking**, plus added resource demands and pipeline complexity from retraining the reward model.

# Thank you!

[\[2305.20050\] Let's Verify Step by Step](#)