Is PRM Necessary? Problem-Solving RL Implicitly Induces PRM Capability in LLMs

Zhangyin Feng*
Huawei Technologies Ltd.
fengzhangyin@huawei.com

Qianglong Chen*
Huawei Technologies Ltd.
chenqianglong.ai@gmail.com

Ning Lu
HKUST
nluab@cse.ust.hk

Yongqian Li
Huawei Technologies Ltd.
liyongqian2@huawei.com

Siqi Cheng
Huawei Technologies Ltd.
chengsiqi4@huawei.com

Shuangmu Peng
Huawei Technologies Ltd.
pengshuangmu@huawei.com

Duyu Tang
Huawei Technologies Ltd.
tangduyu@huawei.com

Shengcai Liu†
Guangdong Provincial Key Laboratory of Brain-Inspired Intelligent Computation, Department of CSE, SUSTech
liusc3@sustech.edu.cn

Zhirui Zhang†
Huawei Technologies Ltd.
zrustc11@gmail.com

Abstract

The development of reasoning capabilities represents a critical frontier in large language models (LLMs) research, where reinforcement learning (RL) and process reward models (PRMs) have emerged as predominant methodological frameworks. Contrary to conventional wisdom, empirical evidence from DeepSeek-R1 demonstrates that pure RL training focused on mathematical problem-solving can progressively enhance reasoning abilities without PRM integration, challenging the perceived necessity of process supervision. In this study, we conduct a systematic investigation of the relationship between RL training and PRM capabilities. Our findings demonstrate that problem-solving proficiency and process supervision capabilities represent complementary dimensions of reasoning that co-evolve synergistically during pure RL training. In particular, current PRMs underperform simple baselines like majority voting when applied to state-of-the-art models such as DeepSeek-R1 and QwQ-32B. To address this limitation, we propose Self-PRM, an introspective framework in which models autonomously evaluate and rerank their generated solutions through self-reward mechanisms. Although Self-PRM consistently improves the accuracy of the benchmark (particularly with larger sample sizes), analysis exposes persistent challenges: The approach exhibits low precision (<10%) on difficult problems, frequently misclassifying flawed solutions as valid. These analyses underscore the need for continued RL scaling to improve reward alignment and introspective accuracy. Overall, our findings suggest that PRM may not be essential for enhancing complex reasoning, as pure RL not only improves problem-solving skills but also inherently fosters robust PRM capabilities. We hope these findings provide actionable insights for building more reliable and self-aware complex reasoning models.

*Equal contribution.
†Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

1 Introduction

Recent advances in reinforcement learning (RL) scaling have substantially improved the ability of large language models (LLMs) to perform complex reasoning tasks. This progress is exemplified by the emergence of state-of-the-art (SOTA) models such as Kimi 1.5 [25], OpenReasonerZero [10], DeepSeek-R1 [8], QwQ-32B [32, 26], and Claude-3.7-Sonnet [1], all of which demonstrate remarkable problem-solving capabilities achieved through pure RL training.

Alongside this RL-driven progress, process reward models (PRMs) [16, 30, 2] have emerged as an alternative paradigm for reasoning supervision, with the aim of evaluating and optimizing the quality of intermediate reasoning steps. Despite their conceptual appeal, PRMs encounter three fundamental limitations: (1) the inherent ambiguity in defining granular reasoning steps, (2) prohibitive annotation costs, and (3) systemic vulnerability to reward hacking mechanisms. In particular, the DeepSeek-R1 technical report highlights these limitations [8], demonstrating that PRMs did not yield significant gains in the performance of complex reasoning. However, recent work continues to explore the potential of PRMs. For example, PRIME [4] introduces implicit reward mechanisms to bypass the need for labeled steps, GenPRM [37] uses code verification and relative progress estimation to improve PRM performance, and Qwen PROCESSBENCH [38] offers a benchmark to assess the quality of the reasoning trace. These efforts underscore the ongoing interest in PRMs as a path to better reasoning supervision. However, a crucial question remains largely unexplored: *Is explicit PRM training truly necessary for developing models capable of process-level reasoning?* In other words, can RL training alone – in the absence of any process supervision – lead to models that are both strong problem solvers and capable evaluators of reasoning quality?

To answer this, in this work, we present the first systematic empirical study exploring the intrinsic connection between RL training and PRM capabilities. Our key insight is that problem-solving and process supervision are not orthogonal skills, but rather complementary competencies that co-evolve during pure RL training. Through a series of experiments, we examine how reasoning models trained solely with rule-based rewards develop strong process-level judgement capabilities, without access to fine-grained supervision. Through a series of controlled experiments on math reasoning tasks, we demonstrate that RL-trained models like DeepSeek-R1 and QwQ-32B exhibit strong process judgement abilities, exceeding those of models explicitly trained with PRMs.

In addition, we find that existing PRMs provide little to no benefit when applied to these models for reranking, often underperforming compared to simple heuristics such as majority voting. To address this, we introduce Self-PRM, an introspective approach where a model leverages its own internal reward signal to re-rank sampled outputs. Self-PRM consistently improves performance – particularly at higher sample sizes – by aligning candidate selection with the model's own reasoning preferences. Despite these improvements, a detailed analysis reveals that Self-PRM suffers from low precision on challenging problems, often misclassifying incorrect solutions as correct. Stronger models, such as DeepSeek-R1 and QwQ-32B, exhibit more reliable self-evaluation, but the issue persists, highlighting the limitations of introspective scoring when reward alignment is imperfect. These findings challenge conventional assumptions about the necessity of process supervision and suggest that lightweight, introspective training signals – whether through self-evaluation or continued RL scaling – may provide a viable path forward.

In summary, our main contributions are as follows:

- We provide the first systematic study demonstrating that RL training alone – without any process-level supervision – can endow models with strong PRM capabilities.
- We empirically show that existing PRMs fail to improve performance when applied to strong RL-trained models, and often underperform simple baselines such as majority voting.
- We introduce the Self-PRM framework, in which models rerank their own outputs using internal reward signals, achieving consistent performance gains – while also revealing limitations in precision that highlight the need for improved reward alignment.

2 Preliminaries

Reinforcement Learning. RL provides a framework in which an agent learns to make sequential decisions by interacting with an environment to maximize cumulative rewards. Formally, RL is

Table 1: Evaluation results of different LLMs on PROCESSBENCH. We report the error, correct and F1 score of the respective accuracies on erroneous and correct samples.

Model	GSM8K			MATH			OlympiadBench			OmniMATH			Average
	Error	Correct	F1	Error	Correct	F1	Error	Correct	F1	Error	Correct	F1	
Proprietary LLMs (Critic)													
GPT-4o-0806	70.0	91.2	79.2	54.4	76.6	63.6	45.8	58.4	51.4	45.2	65.6	53.5	61.9
o1-mini	88.9	97.9	93.2	83.5	95.1	88.9	80.2	95.6	87.2	74.8	91.7	82.4	87.9
*DeepSeek R1 and R1 Distilled Models, prompted as **Generative PRM***													
R1-Distill-Qwen-7B	45.9	90.2	60.8	51.5	82.8	63.5	35.6	73.5	47.9	27.8	70.1	39.8	53.0
R1-Distill-Qwen-32B	74.4	98.4	84.7	71.5	89.4	79.5	64.3	85.5	73.4	59.0	83.8	69.3	76.7
DeepSeek-R1	84.1	95.3	89.3	82.3	91.1	86.5	78.2	86.4	82.1	71.7	80.9	76.0	83.5
*Qwen-series Models, prompted as **Generative PRM***													
Qwen2.5-Math-7B-Instruct	15.5	100.0	26.8	14.8	96.8	25.7	7.7	91.7	14.2	6.9	88.0	12.7	19.9
Qwen2.5-Math-72B-Instruct	49.8	96.9	65.8	36.0	94.3	52.1	19.5	97.3	32.5	19.0	96.3	31.7	45.5
Qwen2.5-7B-Instruct	40.6	33.2	36.5	30.8	45.1	36.6	26.5	33.9	29.7	26.2	28.6	27.4	32.6
Qwen2.5-14B-Instruct	54.6	94.8	69.3	38.4	87.4	53.3	31.5	78.8	45.0	28.3	76.3	41.3	52.2
Qwen2.5-32B-Instruct	49.3	97.9	65.6	36.7	95.8	53.1	25.3	95.9	40.0	24.1	92.5	38.3	49.3
Qwen2.5-72B-Instruct	62.8	96.9	76.2	46.3	93.1	61.8	38.7	92.6	54.6	36.6	90.9	52.2	61.2
QwQ-32B	84.1	97.4	90.2	83.0	90.4	86.5	75.9	87.3	81.2	71.1	83.8	77.0	83.7
*Other Models with Training PRM data, as **Discriminative PRM***													
Skywork-PRM-1.5B	50.2	71.5	59.0	37.9	65.2	48.0	15.4	26.0	19.3	13.6	32.8	19.2	36.4
Math-Shepherd-PRM-7B	32.4	91.7	47.9	18.0	82.0	29.5	15.0	71.1	24.8	14.2	73.0	23.8	31.5
RLHFlow-PRM-Mistral-8B	33.8	99.0	50.4	21.7	72.2	33.4	8.2	43.1	13.8	9.6	45.2	15.8	28.4
RLHFlow-PRM-Deepseek-8B	24.2	98.4	38.8	21.4	80.0	33.8	10.1	51.0	16.9	10.9	51.9	16.9	26.6
Skywork-PRM-7B	61.8	82.9	70.8	43.8	62.2	53.6	17.9	31.9	22.9	14.0	41.9	21.0	42.1
Qwen2.5-Math-7B-Math-Shepherd	46.4	95.9	62.5	18.9	96.6	31.6	7.4	93.8	13.7	4.0	95.0	7.7	28.9
Qwen2.5-Math-7B-PRM800K	53.1	95.3	68.2	48.0	90.1	62.6	35.7	87.3	50.7	29.8	86.1	44.3	56.5
RetrievalPRM-7B	64.7	88.1	74.6	67.2	75.6	71.1	56.0	65.2	60.2	52.8	62.7	57.3	65.8
Qwen2.5-Math-PRM-7B	72.0	96.4	82.4	68.0	90.4	77.6	55.7	85.5	67.5	55.2	83.0	66.3	73.5
Qwen2.5-Math-PRM-72B	78.7	97.9	87.3	74.2	88.2	80.6	67.9	82.0	74.3	64.8	78.8	71.1	78.3

typically modeled as a Markov Decision Process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space, $\mathcal{P}$ the state transition dynamics, $\mathcal{R}$ the reward function, and γ the discount factor. In the context of LLMs, RL is widely used to align model behavior with human preferences. One prominent approach is Reinforcement Learning from Human Feedback (RLHF) [20], where a reward model (RM) is trained to approximate human judgements over generated outputs. The base language model is then optimized using RL to maximize the reward provided by this RM. Several methods have been proposed to implement this optimization process effectively, including PPO [22], DPO [21], GRPO [23], DAPO [35] and VAPO [36] etc. These methods differ in how explicitly they use reward signals, but all share the goal of training LLMs to produce outputs that are more helpful, truthful, and aligned. Notably, recent RL-based models – including DeepSeek-R1 [8], Kimi1.5 [25], QwQ-32B [26] , OpenReasonerZero [10] – achieve state-of-the-art performance on complex reasoning tasks without process-level supervision. Meanwhile, some works [7, 12, 24, 14, 5] are exploring RL with tools for efficient reasoning. While Retool [5] use DAPO-MATH-17k and DAPO for RL training to guide LLMs towards optimal strategies for leveraging external computational tools during reasoning, ToRL [14] leverage 7B-base to improve the complex reasoning ability.

Process Reward Models. PRMs [2, 15, 16, 17, 29, 30, 39, 27, 13] are designed to evaluate the reasoning process of LLMs, not just the final answer. They provide rewards or scores for each intermediate step in a reasoning path, helping models learn to think in a more structured, logical, and interpretable way. While PRM800K [15] is proposed to train the reward model, AlphaMath [2] leverage Monte Carlo Tree Search (MCTS) to unleash the potential of a well-pretrained LLM to autonomously enhance its mathematical reasoning. PRM is especially useful for tasks that require multi-step problem solving, such as math or logical reasoning. Compared to traditional outcome-supervised reward models [3, 34] that focus only on whether the final answer is correct, PRMs can give more detailed feedback, making them valuable for improving model alignment and transparency. They are typically trained on labeled reasoning examples, learning to distinguish good reasoning steps from incorrect or irrelevant ones. Despite their potential, PRMs face challenges such as limited high-quality reasoning data, difficulties in defining stepwise correctness, and risks of models exploiting the reward function. To improve PRMs, researchers are exploring better data collection methods, combining them with reinforcement learning algorithms (e.g., PPO [22]), and even letting models evaluate their own reasoning to enhance learning [2] or extend them to visual reasoning [29].

Table 2: Chi-square test results for model performance on PROCESSBENCH, including GSM8K, MATH, OlympiadBench, and OmniMath datasets. We classify the math problems into `True` and `False` categories, depending on whether the LLMs can provide correct solutions. Similarly, we categorize the judgment results into `Correct` and `Error` categories, based on whether the LLMs deliver the correct judgments.

Model	Solution	GSM8K			MATH			OLYMPIADBENCH			OMNIMATH			ALL		
		Correct	Error	p	Correct	Error	p	Correct	Error	p	Correct	Error	p	Correct	Error	p
Qwen2.5-32B	True	327	52	2.9e-5	599	201	0	298	131	0	243	161	0	1467	545	0
	False	11	10		63	137		220	351		134	462		428	960	
R1-Distill-Qwen-32B	True	315	47	0.071	755	159	3.08e-3	439	146	3.21e-3	444	171	0	1953	559	0
	False	29	9		33	17		276	139		206	179		544	344	
QwQ-32B	True	338	42	0.011	826	147	0.125	507	110	0.023	543	149	3.9e-5	2214	448	0
	False	14	6		20	7		292	91		204	104		530	208	
DeepSeek-R1	True	330	44	0.025	831	141	0.039	476	122	1.15e-3	501	167	0.012	2138	474	0
	False	19	7		20	8		284	118		224	108		547	241	

3 Intrinsic Connection between RL Training and PRM

3.1 Experimental Setup

We evaluate the PRM capabilities of various models using PROCESSBENCH, a publicly available benchmark designed to assess the reasoning abilities of LLMs across diverse domains of mathematical problem-solving. PROCESSBENCH comprises four distinct datasets: GSM8K [3], MATH, OLYMPIADBENCH [9], and OMNIMATH [6]. For each dataset, we report three metrics: Error Rate, Correct Rate, and F1 Score. We categorize the evaluated models into three main groups:

- Proprietary LLMs (Critic Models): This group includes closed-source models, such as GPT-4o-0806 [11] and o1-mini [19], which serve as reference points.
- Qwen-series and DeepSeek-series Models (Prompted as Generative PRM): These models are not directly fine-tuned on RM or PRM datasets; instead, they are prompted as PRMs. Representative models include Qwen2.5-Math-7B-Instruct [33] and DeepSeek-R1 [8].
- Models Fine-Tuned with PRM Data (Discriminative PRM): These models are explicitly trained on datasets constructed with PRM-style supervision. Models include Math-Shepherd-PRM [28], Skywork-PRM-7B [18], RetrievalPRM-7B [40], and RLHFlow-PRM-Mistral-8B [31].

Additionally, to further validate the emergence of PRM capabilities during RL training, we conduct experiments using Qwen2.5-7B-Base with DAPO-Math-17k [35], employing DAPO as the policy optimization algorithm. The hyperparameters are set as follows: learning rate of 1e-6, batch size of 256, prompt length of 2048, output length of 10240, group size of 16, clipping ratio (high) of 0.28, overlong buffer length of 4096, and an overlong penalty factor of 1.0.

3.2 Empirical Analysis on PROCESSBENCH

Evaluation Results on PROCESSBENCH. The evaluation results of different LLMs on PROCESSBENCH are presented in Table 1, which summarizes the PRM performance of various models across the GSM8K, MATH, OLYMPIADBENCH, and OMNIMATH datasets. Our findings indicate that models trained purely with reinforcement learning (RL), such as DeepSeek-R1 and QwQ-32B, exhibit strong inherent PRM capabilities, despite not being explicitly supervised with PRM-labeled data. These RL-trained models consistently outperform discriminative PRM models that were fine-tuned directly on PRM-annotated datasets. For instance, DeepSeek-R1 achieves an average F1 score of 83.5, while QwQ-32B achieves 83.7, both surpassing all models trained with PRM labels in a discriminative setting, including Skywork-PRM-7B (42.1), Qwen2.5-Math-PRM-7B (73.5), and Qwen2.5-Math-PRM-72B (78.3). This suggests that PRM capabilities can emerge implicitly through RL, without the need for explicit PRM supervision.

Problem-Solving V.S. Judgment: A Statistical Perspective. We further conduct chi-square tests to evaluate the correlation between problem-solving proficiency and process supervision capabilities in the same model. Specifically, we categorize the mathematical problems in `ProcessBench` into two groups for each model: (1) `True`: Problems for which the LLM generates correct solutions; (2) `False`: Problems where the LLM fails to produce correct solutions. Similarly, we classify the model's

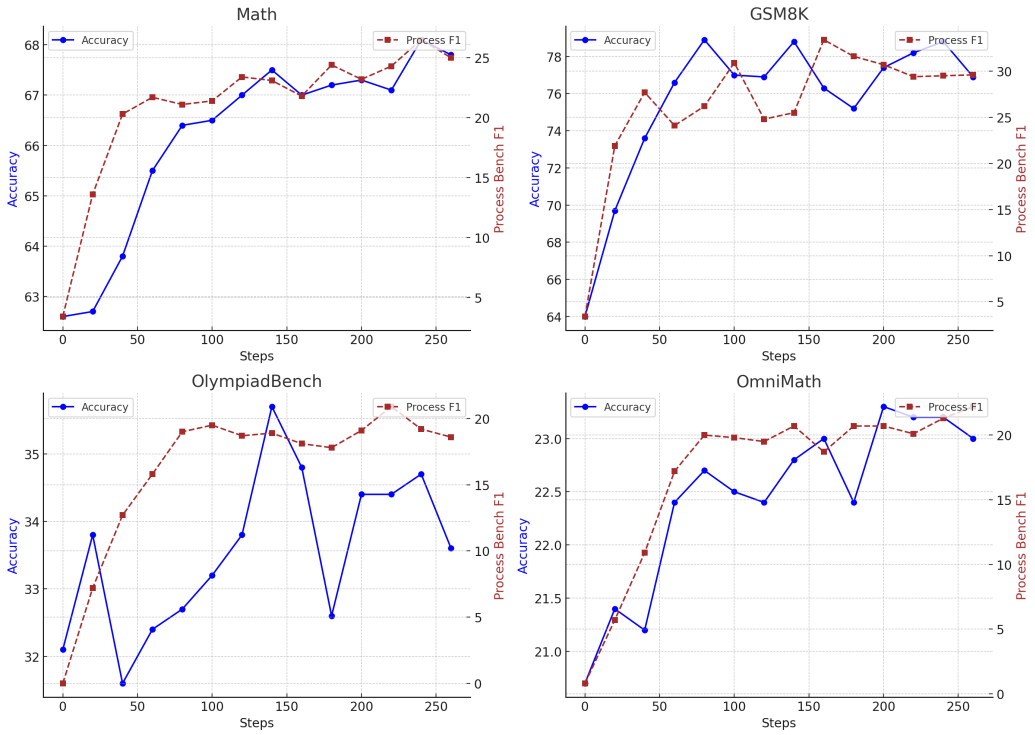

Figure 1: Problem Solving Accuracy and F1 Results of the RL-trained model on ProcessBench. We leverage Qwen2.5-7b-base for RL training.

judgment outcomes into: (1) `Correct`: Cases where the LLM accurately evaluates the correctness of the paired problem-solution instances in `ProcessBench`; (2) `Error`: Cases where the LLM's judgment is incorrect. Based on these categorizations, we count the total number of instances in each of the four groups and calculate the p-value to test the hypothesis that problem-solving proficiency and process supervision capabilities are independent of each other. As summarized in Table 2, all models exhibit statistically significant correlations ($p < 0.05$) in the aggregate analysis, strongly rejecting the null hypothesis. This indicates a strong correlation between problem-solving proficiency and process supervision capabilities. Specifically, for the GSM8K dataset, three models achieve significant results: Qwen2.5-32B ($p = 2.9 \times 10^{-5}$), QwQ-32B ($p = 0.011$), and DeepSeek-R1 ($p = 0.025$), whereas R1-Distill-Qwen-32B did not reach statistical significance ($p = 0.071$). For the MATH dataset, all models except QwQ-32B exhibited significance: Qwen2.5-32B ($p = 0$), R1-Distill-Qwen-32B ($p = 3.08 \times 10^{-3}$), and DeepSeek-R1 ($p = 0.039$). QwQ-32B's performance is non-significant ($p = 0.125$). For the OLYMPIADBENCH and OMNIMATH datasets, all models achieve statistically significant results. The chi-square tests conclusively validate that problem-solving proficiency and process supervision capabilities are intrinsically linked in LLMs, and this correlation persists across diverse mathematical reasoning tasks.

Takeaway 1:

Empirical results demonstrate that RL-trained models (e.g., DeepSeek-R1 and QwQ-32B) develop robust PRM capabilities without explicit PRM supervision, consistently surpassing discriminative PRM baselines. Chi-square analysis ($p < 0.05$) confirms a statistically significant positive correlation between process-level judgment accuracy and problem-solving proficiency.

3.3 Can RL Training Improve PRM Capability?

Since the RL-trained models shows strong PRM capability, we further conduct the experiment to observe the change of PRM capability during RL training. As illustrated in Figure 1, we analyze the learning curves of the RL-trained model through four mathematical reasoning benchmarks (GSM8K,

MATH, OLYMPIADBENCH, and OMNIMATH), evaluating both final-answer accuracy (solid blue lines) and ProcessBench F1 scores (red dashed lines). The results demonstrate a consistent and parallel enhancement in both final-answer accuracy and process judgement ability as training steps increase. A salient trend is that gains in process judgement often precede or surpass those in accuracy, particularly in the early training phases. For instance, on the MATH and GSM8K benchmarks, the F1 score on ProcessBench increases sharply within the first 80 steps, while accuracy improves more gradually. This suggests that RL enables the model to internalize and identify valid reasoning structures before fully mastering the corresponding solution strategies. In more challenging problems, such as OLYMPIADBENCH, the F1 score on ProcessBench improves consistently, whereas accuracy exhibits greater volatility. This divergence highlights the value of process-level learning as a stable and transferable signal, even when final-answer correctness is harder to optimize. Similarly, in OMNIMATH, where domain generalization is required, the model achieves modest but consistent improvements in both metrics, indicating the scalability of process supervision under domain shift. These findings indicate that problem-solving and process supervision are not orthogonal skills, but rather complementary competencies that co-evolve during pure RL training.

Takeaway 2:

RL improves both the accuracy of problem solving and the PRM capability in a coordinated manner. Crucially, improvements in process judgement often emerge earlier and more steadily than accuracy, highlighting RL's capacity to implicitly foster interpretable reasoning even in complex or out-of-domain tasks.

4 Explore the Performance Limit of Reasoning Models

4.1 Experimental Setup

In order to further explore whether a model's own problem-solving ability can enhance its capability as a PRM, we propose the self-reference (Self-REF) paradigm, where the model's generated solutions are used as reference signals to supervise PRM alignment. Using Qwen2.5-32B-Instruct, R1-Distill-Qwen-32B, QwQ-32B, and DeepSeek-R1 as the base model, we evaluate two variants on the PROCESSBENCH benchmark (§4.2): (1) we prompt these models as generative PRMs, and (2) a Self-REF-enhanced PRM incorporating the model's own solutions as supervisory signals.

Meanwhile, we evaluate whether PRMs can enhance the performance of strong reasoning models using two experimental settings (§4.3): (1) BoN w/ PRM, where we use a Qwen2.5-MATH-PRM-72B to select the best output from $k = 8, 16, 32, 64$. (2) BoN w/ Self-PRM, where each model (QwQ-32B, DeepSeek-R1) reranks its own outputs using internally derived reward scores. Evaluations are conducted on AIME24, AIME25, and CNMO24, with performance compared across direct sampling (Pass@k), majority voting, BoN with external PRM, and BoN with Self-PRM. To analyze the limitations of Self-PRM, we further compute the number of solutions labeled correct by Self-PRM (S_{PRM}) and the subset that are truly correct (S_{TP}), allowing us to quantify precision across individual problem indices.

4.2 Improving PRM Capability with Self-REF

To understand when self-generated reasoning traces serve as effective supervisory signals for PRM alignment, we access the impact of Self-REF across models with distinct training strategies as shown in Table 3. We observe a clear trend: Self-REF substantially improves PRM performance only for models that have not undergone RL. Specifically, the F1 score of Qwen2.5-32B-Instruct, a purely instruction-tuned model, improves significantly from 45.1 $\rightarrow$ 63.9 (+18.8) with Self-REF. This gain is most pronounced on MATH (+15.6 F1) and OlympiadBench (+13.9 F1), indicating that Self-REF can inject meaningful structure into process modeling where it is otherwise absent. In contrast, RL-trained models such as QwQ-32B and DeepSeek-R1 exhibit marginal or negative changes in F1 when using Self-REF (e.g., QwQ-32B: 83.7 $\rightarrow$ 83.0; DeepSeek-R1: 83.5 $\rightarrow$ 81.1). This suggests that these models have already internalized process reasoning through reinforcement objectives, and additional weakly supervised signals from self-generated traces may introduce noise rather than improve fidelity. Interestingly, the distilled variant R1-Distill-Qwen-32B, which inherits process judgement behaviors through distillation rather than direct RL training, does benefit modestly from

Table 3: Evaluation results of several models on PROCESSBENCH with solutions as references, including Qwen2.5-32B-Instruct, DeepSeek-R1-Distill-Qwen-32B, QwQ-32B, DeepSeek-R1.

Model	GSM8K			MATH			OlympiadBench			OmniMATH			Average
	Error	Correct	F1	Error	Correct	F1	Error	Correct	F1	Error	Correct	F1	
Qwen2.5-1.5B-Instruct	23.2	17.6	20.0	18.9	27.6	22.4	12.4	24.5	16.5	12.3	24.9	16.4	18.8
w/ Self-REF	17.4	10.4	13.0	14.0	19.7	16.4	7.4	33.6	12.1	9.0	27.8	13.6	13.8
Qwen2.5-7B-Instruct	44.9	43.0	43.9	30.1	48.0	37.0	26.5	33.3	29.5	27.1	28.2	27.7	34.5
w/ Self-REF	45.4	82.9	58.7	26.1	80.0	39.4	15.6	72.9	25.7	15.8	70.5	25.8	37.4
Qwen2.5-32B-Instruct	43.0	97.9	59.8	33.3	95.6	49.0	22.4	90.0	35.9	22.4	87.6	35.7	45.1
w/ Self-REF	72.9	96.9	83.2	50.7	88.9	64.6	35.4	83.8	49.8	24.5	79.3	37.4	63.9
R1-Distill-Qwen-32B	74.4	98.4	84.7	71.5	89.4	79.5	64.3	85.5	73.4	59.0	83.8	69.3	76.7
w/ Self-REF	77.3	96.4	85.8	76.1	92.4	83.4	68.8	90.9	78.3	58.0	87.1	69.6	79.3
QwQ-32B	84.1	97.4	90.2	83.0	90.4	86.5	75.9	87.3	81.2	71.1	83.8	77.0	83.7
w/ Self-REF	82.6	93.8	87.8	81.6	88.9	85.1	77.2	85.3	81.0	71.8	83.8	77.3	83.0
DeepSeek-R1	84.1	95.3	89.3	82.3	91.1	86.5	78.2	86.4	82.1	71.7	80.9	76.0	83.5
w/ Self-REF	81.2	93.8	87.0	83.3	87.7	85.5	74.0	79.9	76.8	70.2	79.7	74.6	81.1

Self-REF (F1: 76.7 $\rightarrow$ 79.3), further supporting that Self-REF is effective in settings where explicit reward-driven reasoning supervision is absent or weak. For the small language models, we can observe that qwen2.5-7B-instruct with Self-REF shows improvements on GSM8K and MATH, while its F1 scores decrease on OlympiadBench and OmniMATH. In contrast, qwen2.5-1.5B-instruct with Self-REF exhibits a decline in performance across all test sets. These results are consistent with our expectations: the effectiveness of Self-REF does depend on the base model's reasoning ability. A weaker instruction-tuned model would generate lower-quality reasoning traces, which would act as noisy or incorrect supervisory signals.

Takeaway 3:

Self-Reference (Self-REF) improves PRM performance primarily for instruction-tuned models lacking RL supervision (e.g., +18.8 F1 for Qwen2.5-32B-Instruct), while RL-trained models (QwQ-32B, DeepSeek-R1) show minimal or negative gains, suggesting limited benefit when process reasoning is already reinforced.

4.3 Self-PRM: Can Current PRM Models further Enhance Reasoning Models?

Table 4: Comparison of sampling@k (k = 8, 16, 32, 64) across QwQ-32B and DeepSeek-R1-on three benchmarks under three settings: direct sampling (Pass@), majority voting, BoN with External PRM (BoN w/ PRM), and BoN with Self-PRM.

Method	@k	QwQ-32B			DeepSeek-R1		
		AIME24	AIME25	CNMO24	AIME24	AIME25	CNMO24
Pass	1	73.3	66.7	77.8	80.0	63.3	77.8
	8	90.0	73.3	88.9	93.3	86.7	88.9
	16	90.0	80.0	94.4	93.3	90.0	88.9
	32	93.3	86.7	94.4	93.3	90.0	88.9
	64	93.3	93.3	94.4	93.3	93.3	88.9
Majority Voting	8	80.0	76.7	83.3	80.0	76.7	77.8
	16	83.3	76.7	83.3	83.3	76.7	83.3
	32	86.7	76.7	83.3	86.7	76.7	83.3
	64	86.7	76.7	83.3	86.7	76.7	83.3
BoN w/ PRM	8	83.3	73.3	76.7	83.3	76.7	77.8
	16	83.3	73.3	83.3	83.3	76.7	77.8
	32	86.7	76.7	83.3	86.7	76.7	77.8
	64	86.7	76.7	83.3	86.7	76.7	83.3
BoN w/ Self-PRM	8	83.3	76.7	83.3	83.3	76.7	77.8
	16	86.7	**80.0**	**88.9**	86.7	76.7	83.3
	32	**90.0**	**80.0**	**88.9**	86.7	**83.3**	83.3
	64	**90.0**	**80.0**	83.3	86.7	**83.3**	83.3

To evaluate whether current state-of-the-art PRMs can enhance the performance of reasoning models in complex mathematical reasoning tasks, we conduct experiments on AIME24, AIME25, and CNMO24, using Qwen2.5-Math-PRM-72B as the PRM for reranking. Specifically, we apply the Best-of-N with PRM (BoN w/ PRM) strategy, which selects the highest-scoring solution based on the PRM's minimum-step reward among the candidates. As shown in Table 4, models that already exhibit strong reasoning abilities through RL, such as QwQ-32B and DeepSeek-R1, do not benefit from external PRM-based reranking compared to majority voting. Across all datasets and sampling sizes $k = 8, 16, 32, 64$, BoN w/ PRM achieves performance that is comparable to or slightly worse than majority voting. For example, on AIME25 with QwQ-32B, the accuracy plateaus at 76.7 for both BoN w/ PRM and majority voting across all values of k. Similarly, DeepSeek-R1 gains no measurable improvement from PRM reranking on any benchmark. These findings highlight a key limitation of current PRMs: while they are effective in guiding weaker models, they do not effectively enhance or complement already-aligned reasoning models, and may even misalign with their internal reward signals.

Since reasoning models demonstrate strong PRM performance, we introduce a **Self-PRM** method, where each model (e.g., QwQ-32B or DeepSeek-R1) serves as its own PRM to rerank its sampled outputs. As shown in Table 4, the BoN with Self-PRM strategy consistently outperforms both Pass@k and Majority Voting, particularly at larger sample sizes(e.g., $k = 32, 64$). For instance, QwQ-32B on AIME24 improves from 86.7 (majority voting) to 90.0 (BoN w/ Self-PRM) at $k = 32$ and maintains this gain at $k = 64$. Similar improvements are observed across other datasets and for DeepSeek-R1. These improvements suggest that the model's internal reward signal is better aligned with its reasoning behavior than those of externally trained PRMs, enabling a more effective utilization of its latent reasoning capabilities. Thus, Self-PRM offers a more introspective and model-consistent evaluation strategy for complex reasoning tasks.

Takeaway 4:

External PRMs provide little to no benefit for RL-based reasoning models like QwQ-32B and DeepSeek-R1, and may even perform worse than majority voting. In contrast, Self-PRM, where models use their own internal scoring as process reward to rerank outputs, consistently improves accuracy, particularly at higher sampling sizes. This approach demonstrates better alignment with the model's problem-solving abilities in complex reasoning tasks and greater effectiveness in leveraging its latent PRM capabilities.

4.4 The Limitations of Self-PRM

Table 5: Model performance comparison across different datasets and problem indices.

Model	Dataset	AIME24				AIME25 I						AIME25 II			CNMO24		
	index	62	63	81	89	6	9	10	12	13	14	4	12	14	1	7	17
QwQ-32B	Difficulty	–	0/64	0/64	3/64	–	3/64	–	4/64	2/64	0/64	–	1/64	0/64	3/64	2/64	0/64
	S_{PRM}	–	10	14	18	–	25	–	6	2	8	–	5	2	55	13	16
	S_{TP}	–	0	0	1	–	1	–	0	0	0	–	0	0	2	2	0
DeepSeek-R1	Difficulty	1/64	0/64	0/64	10/64	22/64	–	14/64	0/64	1/64	0/64	34/64	–	6/64	19/64	0/64	0/64
	S_{PRM}	16	5	1	32	44	–	11	7	1	1	2	–	2	39	7	34
	S_{TP}	0	0	0	3	18	–	4	0	0	0	1	–	1	10	0	0

Note: AIME24 problem indices are sourced from the `math-ai/aime24` dataset on Hugging Face.

The indices in Table 5 were specifically chosen from problems that the respective models (QwQ-32B and DeepSeek-R1) initially failed to solve correctly (i.e., failed at pass@1). This selection allows us to analyze the behavior of Self-PRM on what are considered 'hard problems' for each model. For these hard problems, we sampled 64 solutions to see how many the Self-PRM would judge as correct (S_{PRM}), and out of those, how many were truly correct (S_{PRM}). The "-" symbol indicates that the problem was not a hard problem for that particular model (i.e., the model solved it correctly on its first try). For instance, in AIME24 index 62, QwQ-32B succeeded initially, so it was excluded from this failure analysis and marked with "-".

To better understand the limitations of Self-PRM, we conduct a fine-grained analysis at the problem level using $N = 64$ sampled solutions per problem instance. For each case, we define S_{PRM} as the number of solutions labeled as correct by the Self-PRM, and $S_{\text{TP}} \subseteq S_{\text{PRM}}$ as the subset that are truly correct. Thus, the ratio $S_{\text{TP}}/S_{\text{PRM}}$ reflects the precision of Self-PRM's judgment. As shown in Table 5, while Self-PRM often selects a substantial number of candidate solutions as correct (e.g., 55 on CNMO24-index-1 for QwQ-32B), the precision is often low. For example, on CNMO24-index-1, only 2 of the 55 selected solutions are truly correct, yielding a precision of 3.6%. This trend is especially pronounced for high-difficulty problems, such as CNMO24 and certain AIME25 instances, where Self-PRM misclassifies a large number of incorrect reasoning traces as valid. In contrast, DeepSeek-R1, while selecting fewer candidates overall, tends to show higher precision in Self-PRM filtering. On the same CNMO24-index-1 problem, DeepSeek-R1 selects 39 solutions as correct, 10 of which are true positives, resulting in a much higher precision of 25.6%. This suggests that stronger solvers exhibit more reliable self-PRM behaviors, possibly due to more coherent internal alignment between their reasoning and scoring mechanisms. These observations highlight a key challenge: although Self-PRM improves performance at the aggregate level, its fine-grained precision can vary substantially, especially on difficult tasks, where the model tends to overestimate the correctness of its own outputs. Future work may focus on strengthening a model's Self-PRM capabilities through joint training with fine-grained process supervision or continued RL scaling, enabling tighter alignment between the model's problem-solving ability and PRM capabilty.

Takeaway 5:

Self-PRM also suffers from low precision on hard problems, misclassifying many incorrect solutions as correct. Stronger models like DeepSeek-R1 exhibit more reliable self-evaluation. Improving Self-PRM may require joint training with PRM-related tasks or continued RL scaling to enhance internal alignment.

5 Conclusion

This work challenges the prevailing assumption that explicit process-level supervision is necessary for enabling effective reasoning in large language models. Through systematic empirical analysis, we demonstrate that RL alone – without access to fine-grained reasoning annotations – can endow models with strong PRM capabilities. RL-trained models such as DeepSeek-R1 and QwQ-32B not only solve complex problems with high accuracy but also implicitly learn to distinguish between valid and flawed reasoning steps. Furthermore, we find that existing state-of-the-art PRMs fail to enhance the performance of slow-thinking reasoning models, and in some cases, underperform compared to simple baselines like majority voting. In contrast, leveraging internal signals, i.e., Self-PRM, consistently improves performance, particularly at larger sampling sizes. These findings reveal a tight coupling between problem-solving and process-level judgement in RL-trained models. Overall, our findings suggest that PRM may not be essential for enhancing complex reasoning, as pure RL not only improves problem-solving skills but also inherently fosters robust PRM capabilities. We hope these findings provide actionable insights for building more reliable and self-aware complex reasoning models.

6 Limitations

In this work, since mathematical problems offer clear, objective correctness criteria and structured, step-by-step solutions, we chose the math reasoning domain for studying the co-evolution of problem-solving and process supervision capabilities. Our primary goal was to first systematically establish and validate the existence of the strong positive correlation between a model's problem-solving accuracy and its process judgment F1 score, a phenomenon we demonstrate persists across multiple datasets as RL training progresses. However, a deeper investigation of the underlying mechanism is a valuable direction.

Acknowledgments

This work was supported by National Key Research and Development Program of China under Grant 2022YFA1004102, and in part by National Natural Science Foundation of China under Grant 62502192, and National Natural Science Foundation of China under Grant 62272210.

References

[1] Anthropic. Introducing claude. *https://www.anthropic.com/index/introducing-claude/*, 2023.

[2] Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. Alphamath almost zero: Process supervision without process. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 27689–27724. Curran Associates, Inc., 2024.

[3] Karl Cobbe, Vineet Kosaraju, Mo Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168, 2021.

[4] Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

[5] Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms, 2025.

[6] Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. Omni-math: A universal olympiad level mathematic benchmark for large language models. *ArXiv*, abs/2410.07985, 2024.

[7] Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. ToRA: A tool-integrated reasoning agent for mathematical problem solving. In *The Twelfth International Conference on Learning Representations*, 2024.

[8] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[9] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Annual Meeting of the Association for Computational Linguistics*, 2024.

[10] Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, and Heung-Yeung Shum Xiangyu Zhang. Open-reasoner-zero: An open source approach to scaling reinforcement learning on the base model. `https://github.com/Open-Reasoner-Zero/Open-Reasoner-Zero`, 2025.

[11] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[12] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

[13] Wendi Li and Yixuan Li. Process reward model with q-value rankings. In *The Thirteenth International Conference on Learning Representations*, 2025.

[14] Xuefeng Li, Haoyang Zou, and Pengfei Liu. Torl: Scaling tool-integrated rl, 2025.

[15] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *ArXiv*, abs/2305.20050, 2023.

[16] Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen Zhou. Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling. *arXiv preprint arXiv:2502.06703*, 2025.

[17] Qianli Ma, Haotian Zhou, Tingkai Liu, Jianbo Yuan, Pengfei Liu, Yang You, and Hongxia Yang. Let's reward step by step: Step-level reward model as the navigators for reasoning. *arXiv preprint arXiv:2310.10080*, 2023.

[18] Skywork o1 Team. Skywork-o1 open series. `https://huggingface.co/Skywork`, November 2024.

[19] OpenAI. Learning to reason with llms, 2024.

[20] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155, 2022.

[21] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *ArXiv*, abs/2305.18290, 2023.

[22] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017.

[23] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Jun-Mei Song, Mingchuan Zhang, Y. K. Li, Yu Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *ArXiv*, abs/2402.03300, 2024.

[24] Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025.

[25] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

[26] Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown, November 2024.

[27] Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

[28] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[29] Weiyun Wang, Zhangwei Gao, Lianjie Chen, Zhe Chen, Jinguo Zhu, Xiangyu Zhao, Yangzhou Liu, Yue Cao, Shenglong Ye, Xizhou Zhu, et al. Visualprm: An effective process reward model for multimodal reasoning. *arXiv preprint arXiv:2503.10291*, 2025.

[30] Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models. *arXiv preprint arXiv:2408.00724*, 2024.

[31] Wei Xiong, Hanning Zhang, Nan Jiang, and Tong Zhang. An implementation of generative prm. https://github.com/RLHFlow/RLHF-Reward-Modeling, 2024.

[32] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[33] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

[34] Fei Yu, Anningzhe Gao, and Benyou Wang. OVM, outcome-supervised value models for planning in mathematical reasoning. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 858–875, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[35] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Honglin Yu, Weinan Dai, Yuxuan Song, Xiang Wei, Haodong Zhou, Jingjing Liu, Wei Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yong-Xu Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale. 2025.

[36] Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, et al. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025.

[37] Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, et al. Genprm: Scaling test-time compute of process reward models via generative reasoning. *arXiv preprint arXiv:2504.00891*, 2025.

[38] Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Processbench: Identifying process errors in mathematical reasoning. *arXiv preprint arXiv:2412.06559*, 2024.

[39] Jianyuan Zhong, Zeju Li, Zhijian Xu, Xiangyu Wen, and Qiang Xu. Dyve: Thinking fast and slow for dynamic process verification. *ArXiv*, abs/2502.11157, 2025.

[40] Jiachen Zhu, Congmin Zheng, Jianghao Lin, Kounianhua Du, Ying Wen, Yong Yu, Jun Wang, and Weinan Zhang. Retrieval-augmented process reward model for generalizable mathematical reasoning. *arXiv preprint arXiv:2502.14361*, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In this paper, we conduct a systematic investigation of the relationship between RL training and PRM capabilities. Our findings demonstrate that problem-solving proficiency and process supervision capabilities represent complementary dimensions of reasoning that co-evolve synergistically during pure RL training.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See in Section 6

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: Not Applicable

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See in Section 3.1 and Section 4.1

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [NA]

Justification: Not Applicable

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See in Section 4.1

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: Not Applicable

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See in Section 3.1 and Section 4.1

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research presented in this paper conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Not Applicable

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Not Applicable

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All external assets, including datasets, code libraries, and pre-trained models, are properly credited in the paper, primarily in the experiments section and the references.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Not Applicable

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Not Applicable

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Not Applicable

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We utilized a pre-trained Large Language Model and the model version explicitly detailed in Section 3.1 and 4.1.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.