P2S: Probabilistic Process Supervision for General-Domain Reasoning Question Answering

Wenlin Zhong[1], Chengyuan Liu[2], Yiquan Wu[3*], Bovin Tan[3], Changlong Sun[3], Yi Wang[4], Xiaozhong Liu[5], Kun Kuang[2]

[1]School of Software Technology, Zhejiang Unirersity
[2]College of Computer Science and Technology, Zhejiang Unirersity
[3]Guanghua Law School, Zhejiang University
[4]Chongqing Ant Consumer Finance Co,. Ltd , Ant Group
[5]Worcester Polytechnic Institute, Worcester, USA
{22451152, liucy1, wuyiquan, bovintan, 11921173, kunkuang}@zju.edu.cn, haonan.wy@myxiaojin.cn, xliu14@wpi.edu

Abstract

While reinforcement learning with verifiable rewards (RLVR) has advanced LLM reasoning in structured domains like mathematics and programming, its application to general-domain reasoning tasks remains challenging due to the absence of verifiable reward signals. To this end, methods like Reinforcement Learning with Reference Probability Reward (RLPR) have emerged, leveraging the probability of generating the final answer as a reward signal. However, these outcome-focused approaches neglect crucial step-by-step supervision of the reasoning process itself. To address this gap, we introduce Probabilistic Process Supervision (P2S), a novel self-supervision framework that provides fine-grained process rewards without requiring a separate reward model or human-annotated reasoning steps. During reinforcement learning, P2S synthesizes and filters a high-quality reference reasoning chain (gold-CoT). The core of our method is to calculate a Path Faithfulness Reward (PFR) for each reasoning step, which is derived from the conditional probability of generating the gold-CoT's suffix, given the model's current reasoning prefix. Crucially, this PFR can be flexibly integrated with any outcome-based reward, directly tackling the reward sparsity problem by providing dense guidance. Extensive experiments on reading comprehension and medical Question Answering benchmarks show that P2S significantly outperforms strong baselines.

Introduction

Large-scale Reinforcement Learning with Verifiable Rewards (RLVR) has emerged as a promising paradigm to advance the reasoning capabilities of Large Language Models (LLMs) (Guo et al. 2025a; Wen et al. 2025; Xu et al. 2025b). This approach has fueled a major leap forward, particularly in structured, verifiable domains such as mathematics and programming (Shao et al. 2024; Havrilla et al. 2024; Kumar et al. 2024; Cao et al. 2024). Within this paradigm, LLMs are trained using verifiable rewards computed directly from the model's own final outcomes, such as matching ground truth answers, passing unit tests, or selecting the correct option in multiple-choice questions (MCQ) (Schulman et al. 2017; Setlur et al. 2024; Xie et al. 2025).

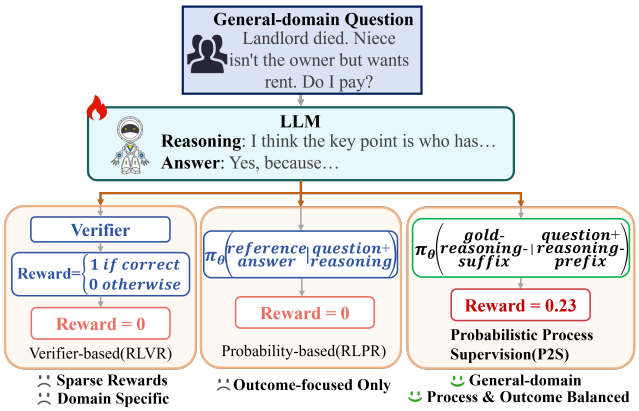

Figure 1: Comparing reward mechanisms: P2S rewards the entire reasoning process.

While RLVR has excelled in specific domains, its success does not readily transfer to general-domain reasoning. The free-form and stylistically diverse nature of answers in these tasks makes designing a direct, verifiable reward signal a challenge. Conventional solutions are inadequate: manually engineering reward functions is unscalable (Zeng et al. 2025; Hu et al. 2025), and training a specialized LLM as a verifier (Ma et al. 2025) demands extensive data annotation, yields unsatisfactory reward quality, and complicates the training pipeline. A more promising direction, Reinforcement Learning with Reference Probability Reward (RLPR) (Xu et al. 2025a; Yu et al. 2025b; Zhou et al. 2025), leverages the generation probability of the final answer as a reward. However, all these outcome-focused methods share critical flaws: they neglect step-by-step process supervision, which can lead models to discover "shortcut" solutions via flawed logic and exacerbates reward sparsity in complex problems.

As shown in Figure 1, we compare P2S with RLVR

*Corresponding author.

and RLPR in general-domain QA. In contrast to domain-specific, sparse-reward verifiers (Figure 1, left) and purely outcome-focused RLPR (Figure 1, center), we argue that the supervisory signal within the reasoning chain itself remains a valuable, untapped resource. Therefore, we aim to design a new reward mechanism that moves beyond sparse outcomes and learns directly from the step-by-step reasoning process, providing more effective and fine-grained supervision for general-domain tasks.

To remedy this oversight, directly supervising the reasoning process is a natural next step. However, prevailing approaches introduce significant burdens. Training a separate reward model necessitates a large corpus of human-annotated or LLM-generated preference data (Lightman et al. 2023), incurring substantial annotation and computational costs. Alternatively, Monte Carlo search-based (Wang et al. 2023) methods, which score each step via multiple rollouts to a terminal state, face severe scalability challenges. The required sample count grows prohibitively with the reasoning chain's length, leading to immense computational overhead. **This highlights a crucial need for a process supervision method that is both low-cost and computationally tractable.**

Our work addresses this challenge by introducing Probabilistic Process Supervision (P2S), a low-cost, self-bootstrapping mechanism that provides fine-grained, process-level supervision by scoring and learning from the model's own reasoning paths, eliminating the need for external reward models or human annotations. To achieve this, we introduce two core techniques.

First, we introduce a dynamic gold-CoT synthesis mechanism. For each problem, we prompt the model with the question and its ground truth answer to generate multiple candidate reasoning paths. These paths are then filtered based on both their final answer's correctness and their internal reasoning quality, creating a high-quality, dynamically updated set of reference chains that adapts to the model's evolving capabilities. Second, we introduce the Path Faithfulness Reward (PFR), our core innovation for dense, step-level supervision. PFR measures how "faithful" a generated reasoning path is to a reference gold-CoT. At each step of the generated path, PFR calculates the conditional probability of completing the rest of the gold-CoT from that point. This step-wise score quantifies whether the model is on a logically sound trajectory. These scores are then aggregated into a sample-level reward that penalizes early deviations and rewards consistent logical progression, thereby directly providing the dense, process-level signal needed to overcome reward sparsity. Finally, P2S operates within a flexible reinforcement learning paradigm. Our process-based PFR can be seamlessly combined with any outcome-based reward, creating a hybrid signal. This joint optimization ensures the model learns not only from successful outcomes but also from the quality of its reasoning process, providing a dense and robust reward signal even when all samples in a batch are incorrect.

Extensive experiments on diverse benchmarks, including general-domain reading comprehension and medical QA, demonstrate that P2S significantly outperforms strong baselines. Our main contributions are summarized as follows:

- We explore the challenging task of reinforcement learning for reasoning in general-domain QA, where traditional verifiable rewards are often unavailable. we identify the limitations of current outcome-focused approaches and propose a new direction centered on process-level supervision derived from the model's own generation probabilities.
- We introduce Probabilistic Process Supervision (P2S), a novel self-supervision framework that generates fine-grained, process-level rewards without costly external reward models or human annotations. At its core, P2S leverages two innovations: a dynamic Gold-CoT synthesis mechanism and our Path Faithfulness Reward (PFR).
- We demonstrate through extensive experiments on diverse benchmarks, including general-domain reading comprehension and medical QA, that P2S consistently and significantly outperforms strong state-of-the-art baselines.

Related Work

Reinforcement Learning for Reasoning

To advance beyond simple prompting for Chain-of-Thought (CoT) reasoning (Kojima et al. 2022; Wei et al. 2022), recent paradigms directly train LLMs, notably via reinforcement learning (RL) on reasoning traces (Shao et al. 2024; He et al. 2025). A successful branch, RLVR, excels in structured domains like math and code by using deterministic, binary outcome rewards from verifiers (Guo et al. 2025a; Yu et al. 2025a; Ye et al. 2025). However, this reliance on verifiers makes RLVR unsuitable for general-domain reasoning, where such clear verification is often impossible.

Reasoning in General Domains

To enable reinforcement learning in general reasoning domains without clear verifiers, research has focused on designing reliable reward signals. One major direction is to train an external generative reward model to act as a judge (Mahan et al. 2024; Ma et al. 2025), which introduces the overhead of developing and maintaining an additional reward model during RL training. A competing approach avoids this by using the policy model's internal feedback as a reward, leveraging signals such as self-certainty or the probability of the ground truth answer as a reward signal. (Xu et al. 2025a; Yu et al. 2025b; Zhou et al. 2025).

Process Reward Supervision

Process supervision improves LLM reasoning consistency by rewarding intermediate steps. While training reward models on human-annotated steps (Li et al. 2024; Lightman et al. 2023) is costly and unscalable, search-based alternatives like Monte Carlo search estimate step values via rollouts (Wang et al. 2023; Guo et al. 2025b). However, these methods incur prohibitive computational costs that scale poorly with reasoning length.

Preliminaries

We first introduce the reasoning optimization with RL, upon which many works build to perform RLVR. Then, we introduce the emerging approach of RLPR (Xu et al. 2025a; Yu et al. 2025b; Zhou et al. 2025).

Reasoning Optimization With RL

In order to enhance the reasoning ability of large models, we adopt Group Relative Policy Optimization (GRPO) (Shao et al. 2024) following the recent advancements such as DeepSeek-R1 (Guo et al. 2025a). Given a question-answer pair (q, a), a behavior policy $\pi_{\theta_{\text{old}}}$ samples a group of G individual responses $\{o_i\}_{i=1}^{G}$. The GRPO objective updates model parameters θ as follows:

$$\begin{aligned}\mathcal{J}_{\text{GRPO}}(\theta) =& \mathbb{E}_{(q,a)\sim D,\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \\ & \left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left\{\min\left[\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}\hat{A}_{i,t},\right.\right.\right. \\ & \left.\left.\left.\text{clip}\left(\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}, 1-\epsilon, 1+\epsilon\right)\hat{A}_{i,t}\right]\right.\right. \\ & \left.\left. - \beta\mathbb{D}_{\text{KL}}(\pi_\theta\|\pi_{\text{ref}})\right\}\right]\end{aligned} \tag{1}$$

The key distinction of GRPO is its advantage estimation for the t-th token in the i-th output, $\hat{A}_{i,t}$. This involves a structured comparison across a group of G outputs $\{\mathbf{o}_i\}_{i=1}^{G}$ sampled for the same prompt. Given corresponding rewards $\{R_i\}_{i=1}^{G}$, the advantage is estimated as:

$$\hat{A}_{i,t} = \frac{r_i - \text{mean}(\{R_i\}_{i=1}^{G})}{\text{std}(\{R_i\}_{i=1}^{G})} \tag{2}$$

In the context of RLVR, the reward r_i is typically a verifiable signal, such as 1 if the final answer is correct and 0 otherwise. This group-normalized formulation steers the policy to assign higher probabilities to trajectories that outperform their peers within the same generation batch.

Reinforcement Learning with Reference Probability Reward (RLPR)

To address the scalability limitations of RLVR, a recent trend in general-domain reasoning is to adopt reinforcement learning paradigms that use probability-based reward signals. It leverages the LLM's own knowledge.

In a typical RLPR setup, for a given input query $\mathbf{q}$, the policy model π_θ first generates a full response $\mathbf{o}$, which includes both a reasoning path $\mathbf{z}$ and a final answer y. The reward is not based on the correctness of the generated answer y. Instead, it is computed from the model's conditional probability of generating the tokens of the ground truth answer y^*, given the generated reasoning path $\mathbf{z}$. This can be formally expressed as the aggregated log-probability:

$$r_{\text{RLPR}} = \sum_{t=1}^{|y^*|}\log\pi_\theta(y_t^*|\mathbf{q},\mathbf{z},y_{<t}^*) \tag{3}$$

where y_t^* is the t-th token of the ground truth answer.

Methodology

In this section, we begin by formally defining the problem, then outline the overall architecture of Probabilistic Process Supervision (P2S) framework, and finally, detail its core components.

Problem Definition

We consider the task of learning a reasoning policy for general-domain question answering. Formally, we are given a dataset $\mathcal{D} = \{(q_i, y_i^*)\}_{i=1}^{N}$, where q_i is a question or prompt, and y_i^* is its corresponding ground-truth final answer. A key characteristic of these tasks is their diversity, spanning multiple domains and featuring answers that are free-form text of varying lengths and styles.

Our goal is to learn a policy π_θ that, given a prompt q, generates a logically sound reasoning path $\mathbf{z} = (\mathbf{z_1}, \mathbf{z_2}, \ldots, \mathbf{z_T},)$ which culminates in a final answer y. This diversity in the target answers y^* makes exact string matching an unsuitable objective. Therefore, our ultimate goal is to maximize the semantic similarity between the generated answer y and the ground-truth y^*.

Overall Architecture

As illustrated in Figure 2, our Probabilistic Process Supervision (P2S) framework operates as a self-improving loop that provides dense, process-level rewards for policy optimization. Firstly, within each iteration of the GRPO, a dynamic Gold-CoT synthesis mechanism leverages the current policy π_θ, guided by a ground truth answer, to generate and filter multiple candidate reasoning paths. This yields a high-quality set of Gold-CoTs specifically tailored for the current learning state. Concurrently, for each generated reasoning trace in the batch, our Path Faithfulness Reward (PFR) is computed by aligning it against a reference Gold-CoT and calculating step-wise conditional probabilities. These step-wise rewards are then weighted and aggregated into a single, sample-level process reward and used to update the policy π_θ, which provides a nuanced score for the entire reasoning path.

Dynamic Gold-CoT Synthesis and Filtering

To ensure a high-fidelity and adaptive supervision signal, P2S dynamically synthesizes and filters reference reasoning paths (Gold-CoTs) in each training iteration. This process involves two main steps: Candidate Synthesis and Quality-Based Filtering.

Candidate Synthesis. To encourage the model to explore paths that lead to the correct answer, we prompt the policy model π_θ with both the query q and the ground truth answer y^* to generate a diverse set of K candidate reasoning paths, $\{\mathbf{o}_k\}_{k=1}^{K}$ during this synthesis stage. This guided generation helps to efficiently sample trajectories within the vicinity of the correct solution space.

$$\{\mathbf{o}_k\}_{k=1}^{K} \sim \pi_\theta(\cdot|q, y^*) \tag{4}$$

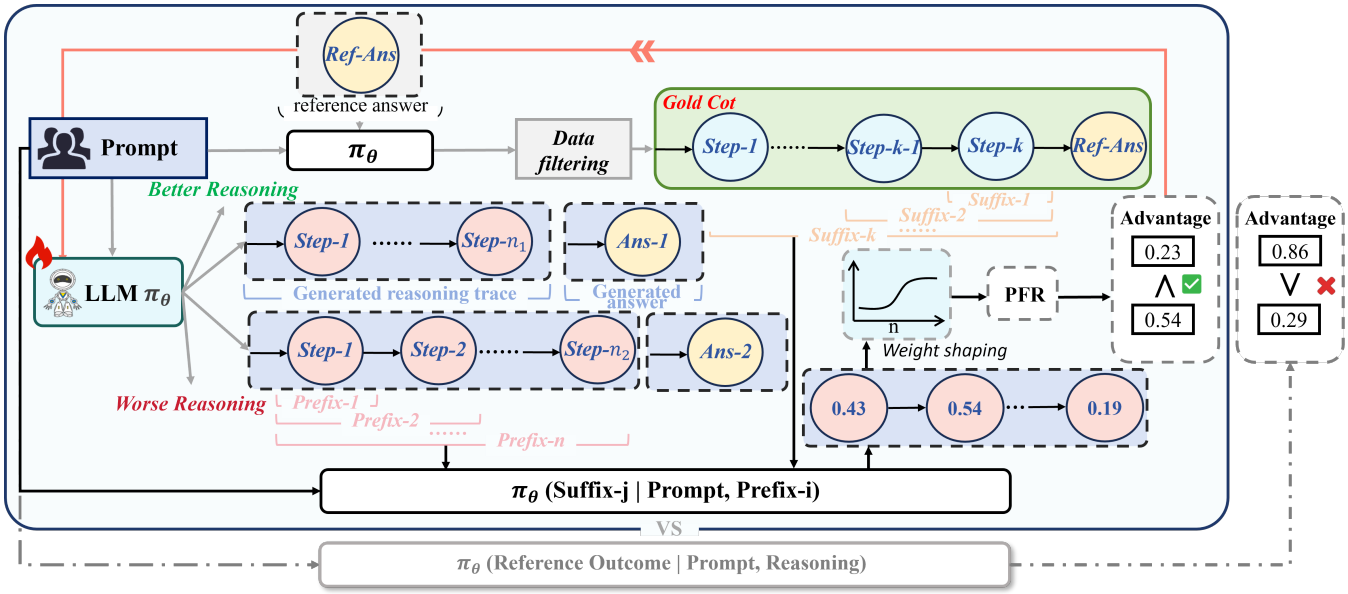

Figure 2: An overview of our Probabilistic Process Supervision (P2S) framework. (1) Gold-CoT Synthesis (Top): A dynamic reference path (Gold-CoT) is created by generating and filtering the policy model's own reasoning outputs. (2) PFR Calculation (Bottom): For each new trace, a step-wise Path Faithfulness Reward (PFR) is computed by aligning it against the Gold-CoT. (3) Reward Shaping & Aggregation: The step-wise rewards are shaped using a sigmoid function to assign progressively higher weights to later reasoning steps. These weighted scores are then summed to produce the final, sample-level Path Faithfulness Reward (PFR) used for policy optimization.

Quality-Based Filtering. Simply generating paths guided by the ground truth answer y^* is insufficient, as they may still be logically flawed, trivial, or fail to reach the correct final answer. Therefore, a filtering stage is crucial to isolate only the highest-quality candidates.

First, we discard any candidate $\mathbf{o}_k$ that does not adhere to a required structural format. Following the standard of (Guo et al. 2025a), this format is <think>Reasoning</think><answer>Answer</answer>. This preliminary step ensures that the reasoning path $\mathbf{z}_k$ and the final answer y_k can be reliably parsed. Let the set of format-correct candidates be $\mathcal{C}_{\text{formatted}}$.

Then, for each candidate in $\mathcal{C}_{\text{formatted}}$, we compute a quality score S_k as the conditional log-probability of generating the ground truth answer y^* given the candidate's reasoning $\mathbf{z}_k$:

$$S_k = \sum_{t=1}^{|y^*|} \log \pi_\theta(y_t^* | \mathbf{q}, \mathbf{z}_k, y_{<t}^*) \tag{5}$$

For each problem q, the definitive gold-CoT $\mathbf{o}^*$ is then selected by finding the candidate that maximizes this score:

$$\mathbf{o}^* = \underset{\mathbf{o}_k \in \mathcal{C}_{\text{formatted}}}{\arg\max}\ S_k$$

The resulting set of candidates $\mathcal{C}_{\text{gold}}$ forms a dynamic and high-quality benchmark for the current training step. This self-improving mechanism creates a virtuous cycle: as the policy model π_θ improves, so does the quality of its self-generated supervision.

Path Faithfulness Reward (PFR)

The core of our P2S framework is the Path Faithfulness Reward (PFR), which provides a dense, step-level reward to guide the model's reasoning process. The central intuition is that a high-quality reasoning prefix should significantly increase the likelihood of generating a subsequent, logically sound reasoning segment from a verified gold-CoT.

We first segment the generated chain $\mathbf{z}$ into a sequence of up to MAX_STEP_NUM equally-sized steps, denoted as $(z_1, z_2, \ldots, z_m)$. This yields a sequence of prefixes $p_1, p_2, \ldots, p_m$, where $p_i = \mathbf{z}[:i]$ is the concatenation of the first i steps. Similarly, we define a suffix of the gold-CoT $\mathbf{o}^*$ starting at step t as $s_t = \mathbf{o}^*[t:]$.

For each intermediate step z_i (where $i < m$), we compute its reward by evaluating the quality of the full prefix $p_i = (z_1, \ldots, z_i)$ that it concludes. This prefix-based evaluation not only assesses z_i within its full contextual history to ensure logical coherence, but also allows the prefix's score to be directly attributed to z_i as the final, decisive step guiding the path forward.

A naive approach would be to measure the conditional probability of generating a gold-CoT suffix given the prefix p_i. However, a high probability might arise simply because the suffix itself is a common or high-probability sequence, regardless of the prefix's quality. Following the work of (Xu et al. 2025a), to isolate the actual contribution of the prefix, we normalize the raw conditional probability by subtracting a baseline. This baseline is defined as the probability of generating the same suffix given the initial question q and a masked version of the prefix p_i, denoted p_{masked}. The re-

sulting score can thus be interpreted as the information gain provided by the final step z_i within the context of its preceding steps.

The reward for step z_i, denoted $r_{\text{step}}(z_i)$, is therefore defined by evaluating its corresponding prefix p_i and finding the maximum log-probability gain over all valid suffixes within the definitive gold-CoT $\mathbf{o}^*$:

$$r_{\text{step}}(z_i) := \max_t \left(\log \pi_\theta(s_t|q, p_i) - \log \pi_\theta(s_t|q, p_{\text{masked}})\right) \tag{6}$$

For the final step z_m, however, the reward is treated differently. This step completes the entire reasoning path z, and its quality is best assessed by its ability to produce the correct final answer. For this terminal step, the objective shifts from measuring information gain to ensuring absolute correctness. Therefore, its reward is defined directly by the conditional log-probability of generating the ground-truth answer y^*, given the full reasoning path z:

$$r_{\text{step}}(z_m) := \log \pi_\theta(y^*|q, z) \tag{7}$$

Time Complexity Analysis. The computational overhead of P2S for a single problem instance is dominated by the number of forward passes (C_{fwd}) through the policy model π_θ. The process involves two main cost components per iteration. First, the Gold-CoT synthesis requires sampling and filtering K candidate paths, incurring a cost proportional to $K \cdot C_{\text{fwd}}$. Second, the PFR calculation for a reasoning path with m steps involves a search over suffixes, resulting in a complexity of approximately $O(m^2 \cdot C_{\text{fwd}})$. Since m is capped by a constant MAX_STEP_NUM, this complexity is well-controlled. Therefore, the total time complexity is $O((K + m^2) \cdot C_{\text{fwd}})$. This is a manageable trade-off, and the computation is highly parallelizable.

Reward Shaping with Step-wise Weighting

A simple averaging of step-wise rewards is suboptimal because it treats all steps equally. Instead, we adopt a strategy that allows the model a "grace period" for initial exploration, such as analyzing the problem or self-correcting from early missteps. To implement this, we introduce a weight shaping mechanism that assigns progressively higher importance to later reasoning steps, thereby focusing supervision on the more converged and critical stages of the reasoning process.

To assign greater importance to later reasoning steps, we compute the final sample-level reward, $R_{\text{PFR-w}}$, as a weighted average of the step-wise rewards $r_{\text{step}}(z_i)$. The weight for each step, w_i, is generated using a monotonically increasing standard sigmoid $\sigma(i)$, ensuring that later steps contribute more significantly to the final reward. The formulation is as follows:

$$R_{\text{PFR-w}} = \frac{\sum_{i=1}^{m} w_i \cdot r_{\text{step}}(z_i)}{\sum_{i=1}^{m} w_i} \tag{8}$$

Hierarchical Reward Integration

A key advantage of our P2S framework is its flexibility, as the Path Faithfulness Reward ($R_{\text{PFR-w}}$) can function either as a standalone process signal or be integrated with other rewards. We present a powerful hierarchical paradigm that combines P2S with an outcome-based reward, assigning scores with a clear priority. First, malformed trajectories are heavily penalized. If any trajectory yields a correct answer, we exclusively use this outcome signal to rapidly amplify the advantage of successful paths. Only when all valid paths fail does our dense PFR serve as a fallback, ensuring a fine-grained learning signal is always available to mitigate reward sparsity.

This hierarchical logic can be formalized concisely. Let $F(i) \in \{0, 1\}$ be an indicator function where $F(i) = 1$ if the format of trajectory i is correct. Let $S_{\mathcal{G}} = \max_{j \in \mathcal{G}} R_{\text{outcome},j}$ be a binary variable indicating whether any trajectory in the group $\mathcal{G}$ was successful. The final reward R_i for trajectory i is then:

$$R_i = \begin{cases} -1 & \text{if } F(i) = 0 \\ R_{\text{outcome},i} & \text{if } F(i) = 1 \text{ and } S_{\mathcal{G}} = 1 \\ R_{\text{PFR-w},i} & \text{if } F(i) = 1 \text{ and } S_{\mathcal{G}} = 0 \end{cases} \tag{9}$$

Cold-Start. To ensure training stability, we adopt a curriculum warm-up strategy (Liu et al. 2025). For the initial S_{warmup} training steps, the model learns the basic task structure using only format-based rewards, with our PFR component deactivated. Subsequently, the full P2S reward mechanism is enabled to refine the logical quality of the reasoning process.

Experiments

Experimental Setup

Datasets We focus on reasoning tasks that lack strict structural verifiers due to their open-ended and stylistically diverse answers, but still possess objectively correct outcomes. Accordingly, we train and evaluate our method on two datasets selected to reflect this challenge. (1) DROP (Dua et al. 2019): A challenging reading comprehension benchmark that requires discrete reasoning over open-domain Wikipedia text, such as arithmetic and sorting. (2) Medical QA (Chen et al. 2024): An open-ended medical question-answering dataset derived from challenging medical exams. For both datasets, we process into a question-answering format and filter to include questions under 2000 and answers between 1-50 characters, creating a 10k/2k random train/test split for each.

Evaluation Metrics Our evaluation employs two complementary metrics for final answers. For lexical similarity, we use **ROUGE-1 F1** to measure overlap with the ground truth answers. To assess semantic correctness, we use LLM-as-a-Judge (Gu et al. 2024) to judge semantic equivalence, including: Claude 4 Sonnet ($\mathbf{ACC_{Claude}}$), GPT-4o ($\mathbf{ACC_{GPT}}$), and a trained 1.5B general-domain Verifier ($\mathbf{ACC_{Verifier}}$) (Ma et al. 2025). Finally, we report the mean of these three accuracy scores, $\mathbf{ACC_{Avg}}$, as a single, robust measure of correctness.

Baselines We compare our method against several baselines, all built upon the **Qwen2.5-1.5B-Instruct** model. Full implementation details for all experiments are provided in

Model	Drop					MedicalQA				
	ROUGE	ACC_{Claude}	ACC_{GPT}	$ACC_{Verifier}$	ACC_{Avg}	ROUGE	ACC_{Claude}	ACC_{GPT}	$ACC_{Verifier}$	ACC_{Avg}
Qwen2.5-1.5B-Instruct	42.23	51.67	50.75	49.15	50.52	40.30	19.20	19.20	27.00	21.80
Prompt-Based										
COT	41.97	45.33	49.00	48.85	47.73	40.09	20.40	21.60	26.60	22.87
Self-Consistency	45.51	51.17	52.67	52.35	52.06	38.76	14.10	17.13	23.75	18.33
Fine-tuning and RL methods										
Full-Sft	71.44	66.00	64.50	63.42	64.64	50.92	20.80	20.04	22.65	21.28
GRPO	70.89	60.00	62.25	62.12	61.46	46.21	17.67	21.00	25.50	21.39
GRPO+SFT-loss	66.18	59.50	63.00	58.90	60.47	45.79	21.00	20.40	24.15	21.85
SFT+GRPO	75.28	66.50	<u>70.14</u>	<u>68.55</u>	<u>68.40</u>	50.57	<u>23.33</u>	20.00	23.80	22.38
General Reasoner	73.03	<u>67.89</u>	65.32	66.30	66.50	<u>51.57</u>	19.18	17.20	**27.45**	21.28
RLPR methods										
DRO	74.85	66.28	67.17	66.65	66.70	50.52	20.11	19.20	23.50	20.94
RLPR	<u>75.48</u>	67.18	68.04	67.57	67.60	51.14	21.16	20.75	<u>26.92</u>	<u>22.94</u>
VeriFree	71.98	64.42	62.17	63.40	63.33	51.46	21.98	<u>21.68</u>	22.85	22.17
P2S	**76.78**	**69.11**	**72.14**	**70.85**	**70.70**	**52.90**	**24.33**	**22.67**	25.85	**24.28**

Table 1: Performance comparison of various Reasoning methods on general-domian QA task. **Bold** and <u>underline</u> indicate the best and second-best results, respectively.

Appendix A. And our baselines are grouped into three categories. (1) **Prompt-based methods** that require no fine-tuning: Chain-of-Thought (CoT) (Wei et al. 2022) and Self-Consistency (Wang et al. 2022). (2) **Fine-tuning and RL methods**, including full supervised fine-tuning (Full-SFT) and several GRPO (Shao et al. 2024) variants. Standalone **GRPO**, the two-stage **SFT+GRPO**, and **GRPO+SFT-loss** (which integrates off-policy knowledge via an auxiliary SFT loss) all use ROUGE-1 F1 as their outcome-based reward. In contrast, **General Reasoner** (Ma et al. 2025) also employs GRPO but replaces this reward with judgments from a trained 1.5B LLM verifier that assesses semantic equivalence. (3) **RLPR-based methods**, which leverage the model's own probabilities for reward, including DRO (Xu et al. 2025a), the original RLPR (Yu et al. 2025b), and VeriFree (Zhou et al. 2025). To ensure a fair comparison and mitigate reward collapse during RL phases, P2S along with the General Reasoner and RLPR-based baselines, adheres to a same cold-start Supervised Fine-Tuning paradigm before RL training (Guo et al. 2025a).

Main Results

Main Results in Table 1 show our method, P2S, outperforms all baselines on both the DROP and MedicalQA datasets. We can draw several key conclusions from the results:

1) P2S significantly improves general-domain reasoning performance. On DROP, it reaches an ACC_{Avg} of 70.70, exceeding the strongest fine-tuned baseline (SFT+GRPO at 68.40) by 2.3 points. This leadership extends to MedicalQA, where P2S achieves an ACC_{Avg} of 24.28, outperforming the next best method (RLPR at 22.94) by over 1.3 points.

2) Our core hypothesis—that dense process supervision is critical—is validated by these results. P2S's superiority is particularly clear against RLPR-based methods (e.g., RLPR, VeriFree). On DROP, for instance, P2S surpasses the strongest RLPR-based method (RLPR) by 1.3 points in ROUGE (76.78 vs. 75.48) and by over 3 points in ACC_{Avg} (70.70 vs. 67.60). This dual improvement proves that our process-focused supervision not only mitigates the reward sparsity of outcome-only approaches but also guides the model to produce answers superior in both form and substance.

3) P2S outperforms representative fine-tuning and RL paradigms, highlighting the efficacy of verifier-free rewards. On DROP, P2S surpasses all GRPO and RLPR variants. More notably, it outperforms General Reasoner by a significant margin of over 4 points in ACC_{Avg} (70.70 vs. 66.50), which uses a 1.5B LLM verifier for its reward signal. This is a crucial finding: our internal, process-based rewards are more effective than guidance from a costly external verifier. Furthermore, the reliability of such verifiers is questionable, as evidenced on MedicalQA. General Reasoner's $ACC_{Verifier}$ score (27.45) is substantially inflated compared to judgments from large-scale models like Claude (19.18) and GPT (17.20). This discrepancy underscores the robustness and efficiency of our verifier-free P2S framework, especially in new domains.

Ablation Study

Our ablation study on DROP (Table 2) validates the contribution of each key component in the P2S framework by systematically removing them from the full model.

Gold-CoT Filtering (GCF) is Crucial. Replacing our Gold-CoT filtering with random path selection (w/o GCF) causes the most substantial performance drop, reducing ACC_{Avg} by 4.5 points. This confirms that high-quality, faithful reasoning paths are a critical foundation for effective process supervision. Path Faithfulness Reward (PFR) is the

core contribution. Removing our core PFR component (w/o PFR) results in a 2.3-point decrease in ACC_{Avg}. This directly validates the effectiveness of our proposed PFR as a critical component for process supervision. Advanced Reward Mechanisms are Effective. We also validated our reward design choices. Replacing sigmoid-based weight shaping with simple averaging (w/o RS) drops ACC_{Avg} by 2.7 points, confirming the benefit of prioritizing later reasoning steps. Similarly, a naive reward summation (w/o HRI) is less effective than our hierarchical integration, proving the advantage of our dynamic fusion strategy.

Model	ROUGE	ACC_{Claude}	$ACC_{GPT\text{-}4o}$	$ACC_{Verifier}$	ACC_{Avg}
P2S (Full)	**76.78**	**69.11**	**72.14**	**70.85**	**70.70**
w/o GCF	71.46	64.21	67.22	67.21	66.21
w/o PFR	75.28	66.50	70.14	68.55	68.40
w/o RS	74.91	64.10	69.88	69.94	67.97
w/o HRI	76.70	68.20	71.27	70.20	69.89

Table 2: Ablation study of P2S components on DROP. **P2S (Full)** is our complete model; **w/o GCF** removes Gold-CoT filtering; **w/o PFR** removes our core Path Faithfulness Reward; **w/o RS** removes sigmoid-based reward shaping; and **w/o HRI** removes hierarchical reward integration.

Effect of Model Scale

To investigate its scalability, we evaluate P2S against the untuned base model and the strong SFT+GRPO baseline at 1.5B and 3B scales on DROP (Table 3). Results highlight two key findings. First, P2S shows remarkable efficiency: our 1.5B model (70.70 ACC_{Avg}) significantly outperforms the much larger 3B base model (62.77), suggesting our process supervision unlocks capabilities beyond simply scaling parameters. Second, P2S's consistent superiority at both scales confirms it is a robust and effective enhancement across different model sizes.

Model Scale	Base		SFT+GRPO		**P2S (Ours)**	
	ROUGE	ACC_{Avg}	ROUGE	ACC_{Avg}	**ROUGE**	**ACC_{Avg}**
1.5B	42.23	50.52	75.28	68.40	**76.78**	**70.70**
3B	47.96	62.77	81.16	74.41	**82.08**	**77.30**

Table 3: Performance on DROP across model scales.

Analysis on Verifiable Subsets

We study the effectiveness of P2S on domains with readily available verifiers. To this end, we created two verifiable subsets—DROP-verifiable (5k) and MedicalQA-verifiable (2.35k)—by filtering for instances with single-word answers. On these, we compare P2S against two outcome-only baselines: the probabilistic RLPR and the rule-based RLVR.

As shown in Figure 3, P2S consistently outperforms both baselines on both subsets, across both exact match (ACC_{exact}) and verifier-based average accuracy (ACC_{Avg}).This crucial finding proves that our P2S provides

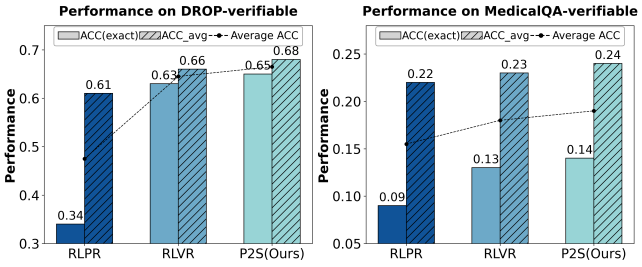

Figure 3: P2S outperforms in verifiable tasks

a fundamentally superior learning signal, extending its benefits far beyond merely overcoming reward sparsity, even in ideal settings for outcome-only methods.

Case Study

Figure 4 provides a case study to illustrate how our Path Faithfulness Reward (PFR) works. Given a Gold-CoT, we analyze two incorrect reasoning paths, z_1 and z_2.

In path z_1, the model makes an early error by analyzing the wrong dates (highlighted in light blue), leading to a low reward score for that step (e.g., 0.12). The error propagates, resulting in even lower scores for subsequent steps (0.09). In contrast, path z_2 correctly identifies the initial entities (highlighted in orange), and our PFR mechanism appropriately assigns a high reward to this correct step (0.87).

Although both paths ultimately fail to produce the correct final answer, our PFR is capable of discerning valuable, correct sub-steps within an overall incorrect reasoning process. This fine-grained reward allows our framework to reinforce partially correct reasoning even within failed attempts.

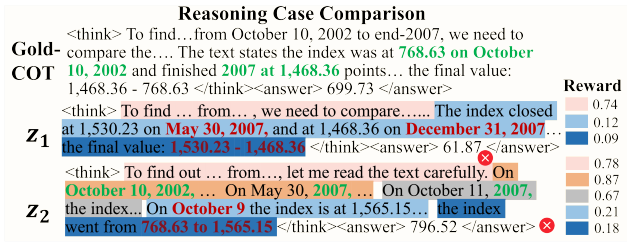

Figure 4: Case Study

Conclusion

In this paper, we introduced Probabilistic Process Supervision (P2S), a novel, low-cost self-supervision framework. At its core, P2S leverages two key innovations: a dynamic mechanism for synthesizing high-quality Gold-CoTs and the Path Faithfulness Reward (PFR), which provides a dense, step-by-step signal by measuring the faithfulness of a generated reasoning path to a reference. Our extensive experiments demonstrated that P2S significantly outperforms strong baselines on challenging reasoning benchmarks. This work proves that it is both feasible and effective to learn directly from the reasoning process itself without external reward models or human annotation.

Acknowledgments

This work was supported in part by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang (2025C02037), the National Natural Science Foundation of China (62376243, 62406287), Key R&D Program of Hangzhou (2025SZDA0254), and Ant Group, Chongqing Ant Consumer Finance Co. All opinions in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Cao, Y.; Zhao, H.; Cheng, Y.; Shu, T.; Chen, Y.; Liu, G.; Liang, G.; Zhao, J.; Yan, J.; and Li, Y. 2024. Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods. *IEEE Transactions on Neural Networks and Learning Systems*.

Chen, J.; Cai, Z.; Ji, K.; Wang, X.; Liu, W.; Wang, R.; Hou, J.; and Wang, B. 2024. HuatuoGPT-o1, Towards Medical Complex Reasoning with LLMs. arXiv:2412.18925.

Dua, D.; Wang, Y.; Dasigi, P.; Stanovsky, G.; Singh, S.; and Gardner, M. 2019. DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs. In *Proc. of NAACL*.

Gu, J.; Jiang, X.; Shi, Z.; Tan, H.; Zhai, X.; Xu, C.; Li, W.; Shen, Y.; Ma, S.; Liu, H.; et al. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Guo, Y.; Xu, L.; Liu, J.; Ye, D.; and Qiu, S. 2025b. Segment policy optimization: Effective segment-level credit assignment in rl for large language models. *arXiv preprint arXiv:2505.23564*.

Havrilla, A.; Du, Y.; Raparthy, S. C.; Nalmpantis, C.; Dwivedi-Yu, J.; Zhuravinskyi, M.; Hambro, E.; Sukhbaatar, S.; and Raileanu, R. 2024. Teaching large language models to reason with reinforcement learning. *arXiv preprint arXiv:2403.04642*.

He, J.; Liu, J.; Liu, C. Y.; Yan, R.; Wang, C.; Cheng, P.; Zhang, X.; Zhang, F.; Xu, J.; Shen, W.; et al. 2025. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*.

Hu, J.; Zhang, Y.; Han, Q.; Jiang, D.; Zhang, X.; and Shum, H.-Y. 2025. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213.

Kumar, A.; Zhuang, V.; Agarwal, R.; Su, Y.; Co-Reyes, J. D.; Singh, A.; Baumli, K.; Iqbal, S.; Bishop, C.; Roelofs, R.; et al. 2024. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*.

Li, J.; Liang, X.; Zhang, J.; Yang, Y.; Feng, C.; and Gao, Y. 2024. PSPO*: An Effective Process-supervised Policy Optimization for Reasoning Alignment. *arXiv preprint arXiv:2411.11681*.

Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; and Cobbe, K. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Liu, Z.; Gong, C.; Fu, X.; Liu, Y.; Chen, R.; Hu, S.; Zhang, S.; Liu, R.; Zhang, Q.; and Tu, D. 2025. GHPO: Adaptive Guidance for Stable and Efficient LLM Reinforcement Learning. *arXiv preprint arXiv:2507.10628*.

Ma, X.; Liu, Q.; Jiang, D.; Zhang, G.; Ma, Z.; and Chen, W. 2025. General-reasoner: Advancing llm reasoning across all domains. *arXiv preprint arXiv:2505.14652*.

Mahan, D.; Van Phung, D.; Rafailov, R.; Blagden, C.; Lile, N.; Castricato, L.; Fränken, J.-P.; Finn, C.; and Albalak, A. 2024. Generative reward models. *arXiv preprint arXiv:2410.12832*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Setlur, A.; Nagpal, C.; Fisch, A.; Geng, X.; Eisenstein, J.; Agarwal, R.; Agarwal, A.; Berant, J.; and Kumar, A. 2024. Rewarding progress: Scaling automated process verifiers for llm reasoning. *arXiv preprint arXiv:2410.08146*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Wang, P.; Li, L.; Shao, Z.; Xu, R.; Dai, D.; Li, Y.; Chen, D.; Wu, Y.; and Sui, Z. 2023. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Wen, L.; Cai, Y.; Xiao, F.; He, X.; An, Q.; Duan, Z.; Du, Y.; Liu, J.; Tang, L.; Lv, X.; et al. 2025. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond. *arXiv preprint arXiv:2503.10460*.

Xie, T.; Gao, Z.; Ren, Q.; Luo, H.; Hong, Y.; Dai, B.; Zhou, J.; Qiu, K.; Wu, Z.; and Luo, C. 2025. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2502.14768*.

Xu, Y.; Chakraborty, T.; Sharma, S.; Nunes, L.; Kıcıman, E.; Lu, S.; and Chandra, R. 2025a. Direct reasoning optimization: Llms can reward and refine their own reasoning for open-ended tasks. *arXiv preprint arXiv:2506.13351*.

Xu, Z.; Yue, X.; Wang, Z.; Liu, Q.; Zhao, X.; Zhang, J.; Zeng, W.; Xing, W.; Kong, D.; Lin, C.; et al. 2025b. Copyright Protection for Large Language Models: A Survey of Methods, Challenges, and Trends. *arXiv preprint arXiv:2508.11548*.

Ye, Y.; Huang, Z.; Xiao, Y.; Chern, E.; Xia, S.; and Liu, P. 2025. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*.

Yu, Q.; Zhang, Z.; Zhu, R.; Yuan, Y.; Zuo, X.; Yue, Y.; Dai, W.; Fan, T.; Liu, G.; Liu, L.; et al. 2025a. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Yu, T.; Ji, B.; Wang, S.; Yao, S.; Wang, Z.; Cui, G.; Yuan, L.; Ding, N.; Yao, Y.; Liu, Z.; et al. 2025b. RLPR: Extrapolating RLVR to General Domains without Verifiers. *arXiv preprint arXiv:2506.18254*.

Zeng, W.; Huang, Y.; Liu, Q.; Liu, W.; He, K.; Ma, Z.; and He, J. 2025. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*.

Zhou, X.; Liu, Z.; Sims, A.; Wang, H.; Pang, T.; Li, C.; Wang, L.; Lin, M.; and Du, C. 2025. Reinforcing General Reasoning without Verifiers. *arXiv preprint arXiv:2505.21493*.