A Survey of Process Reward Models: From Outcome Signals to Process Supervisions for Large Language Models

Congmin Zheng[*1], **Jiachen Zhu**[*1], **Zhuoying Ou**[*1], **Yuxiang Chen**[2], **Kangning Zhang**[1], **Rong Shan**[1] **Zeyu Zheng**[3], **Mengyue Yang**[4], **Jianghao Lin**[1†], **Yong Yu**[1], **Weinan Zhang**[1†]
[1]Shanghai Jiao Tong University, [2]University College London,
[3]Carnegie Mellon University, [4]University of Bristol
{desp.zcm,gebro13,zoeouzy23,linjianghao,wnzhang}@sjtu.edu.cn,

Abstract

Although Large Language Models (LLMs) exhibit advanced reasoning ability, conventional alignment remains largely dominated by outcome reward models (ORMs) that judge only final answers. *Process Reward Models* (PRMs) address this gap by evaluating and guiding reasoning at the step or trajectory level. This survey provides a systematic overview of PRMs through the full loop: how to *generate process data*, *build PRMs*, and *use PRMs* for test-time scaling and reinforcement learning. We summarize applications across math, code, text, multimodal reasoning, robotics, and agents, and review emerging benchmarks. Our goal is to clarify design spaces, reveal open challenges, and guide future research toward fine-grained, robust reasoning alignment. To support these efforts, we accompany this survey with an actively updated GitHub repository (https://github.com/despzcm/Survey-of-Process-Reward-Model).

1 Introduction

The advent of Large Language Models (LLMs) has reshaped alignment for reasoning (Shao et al., 2024; Jaech et al., 2024; Yang et al., 2025a; Bai et al., 2025; He et al., 2025a), shifting attention from *outcome-only* supervision to *process-aware* evaluation. Early pipelines predominantly relied on outcome reward models (ORMs) (Lightman et al., 2023) that judge only final answers, providing a single coarse signal for long chains of thought. As reasoning tasks grow longer and more complex, this static, outcome-centric view struggles to capture stepwise progress, diagnose intermediate errors, or allocate computation adaptively.

To address this gap, the community has begun to move beyond coarse outcome supervision toward process reward models (PRMs), which explicitly assess and guide reasoning at the step or trajectory level. As shown in Figure 1, Process Reward Models coupled with a closed loop: *generate process data* → *train PRMs* → *use PRMs* (test-time scaling or RL) → *produce better data*. This loop transforms reward modeling from a one-shot verdict to an iterative controller of reasoning, enabling finer credit assignment, richer diagnostics, and improved robustness.

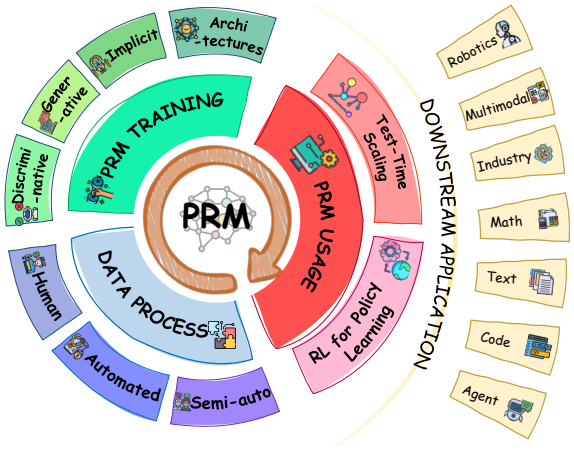

Figure 1: The Process Reward Model (PRM) loop that iteratively *generates data*, *trains PRMs*, and *uses PRMs* to improve policies and produce new data.

The emergence of PRMs marks a pivotal shift. Rather than relying on single-turn or rule-based evaluation, PRMs assess partial solutions and trajectories, leverage context for adaptive "reason-then-rate" verification, and integrate with inference-time controllers and reinforcement learning (RL) objectives. In this paradigm, supervision becomes proactive: it not only evaluates but also *steers* search, reflection, and policy updates across diverse sources of evidence (e.g., retrieved knowledge, programs, or multimodal inputs).

Given these rapid advances, we present a systematic survey of PRMs across the full loop: **how to generate data**, **how to build PRMs**, and **how**

[*]Equal Contribution
[†]Corresponding author

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3591–3607
July 2-7, 2026

to use PRMs. Current discussions mainly focus on either test-time scaling paradigms (Zhang et al., 2025g), broad reward modeling taxonomies (Zhong et al., 2025), or generic deep RL reward design (Yu et al., 2025), whereas our PRM survey uniquely targets step-level process reward modeling by organizing the full loop of data generation, PRM building, and usage (test-time scaling and PRM-guided RL) for fine-grained reasoning supervision.

Specifically, this paper is structured as follows. Sec. 2 (**How to Generate Data**) categorizes process supervision into *human annotation*, *automated supervision*, and *semi-automated pipelines*, highlighting fidelity–scalability trade-offs. Sec. 3 (**How to Build PRMs**) reviews modeling paradigms, including *discriminative* vs. *generative* objectives, *explicit* vs. *implicit* supervision, and architectural innovations. Sec. 4 (**How to Use PRMs**) discusses *test-time scaling* (re-ranking, verification-guided decoding, search) and *PRM-guided RL* (dense step-wise rewards and credit assignment). Sec. 5 includes **applications** spanning math, code, multimodal reasoning, agents, and high-stakes domains, and Sec. 6 summarizes **benchmarks**. Further **discussions** are provided in Sec. 7.

2 How to Generate Data

In this section, we address the question of "how to generate data" for training process reward models (PRMs) and categorize existing approaches into three main paradigms: (1) human annotation, (2) automated supervision, and (3) hybrid methods that combine both. Each paradigm reflects a different trade-off between fidelity and scalability, and recent work often integrates multiple strategies to leverage the strengths of one source while mitigating the weaknesses of another.

2.1 Human Annotation

The earliest and most straightforward form of process supervision comes from direct human annotation, where annotators explicitly verify the correctness of intermediate reasoning steps. PRM800K (Lightman et al., 2023) is a representative example, in which human labelers carefully validated each step of multi-hop reasoning chains. This dataset demonstrated that explicitly capturing human judgments about process correctness can substantially improve PRM training, leading to better alignment and more interpretable reasoning outcomes.

Although resource-intensive and limited in scale, human-curated process data has proven to be a critical foundation: it provides high-fidelity signals, establishes benchmarks for other data generation pipelines, and often serves as seed material to guide more scalable methods.

2.2 Automated Supervision

To overcome the bottlenecks of manual labeling, a large body of research explores fully automated approaches that generate process supervision through symbolic verification, consistency checks, execution feedback, or synthetic self-evolution.

Math-Shepherd (Wang et al., 2023) introduced an automated verification pipeline where mathematical reasoning steps are validated using symbolic tools and consistency-checking heuristics, enabling large-scale process supervision without human annotations. FOVER (Kamoi et al., 2025) uses formal verification tools (e.g., Z3, Isabelle) to automatically generate PRM training data with accurate step-level error labels. OmegaPRM (Luo et al., 2024) extends this paradigm by using a divide-and-conquer style Monte Carlo Tree Search (MCTS) algorithm to efficiently identify the first error in a reasoning chain, providing a scalable alternative to human judgment. URSA (Luo et al., 2025) further advances this line by synthesizing process-level supervision for multimodal mathematical reasoning through a fully automated dual-view pipeline, which employs MCTS-based error localization and misinterpretation insertion engines to construct large-scale process annotations.

Expanding beyond mathematics, MT-RewardTree (Feng et al., 2025b) adapts the MCTS-driven framework to machine translation, leveraging approximate MCTS to generate token-level preference pairs entirely through automatic evaluation and filtering, thereby enabling scalable and fine-grained reward modeling without human annotation. Similarly, CodePRM (Li et al., 2025a) employs automated tree search and execution feedback to derive step-level supervision for code reasoning, achieving fully automatic label generation without human involvement. Search-in-Context (Chen et al., 2025d) introduces Monte Carlo Tree Search with dynamic retrieval, which automatically constructs intermediate reasoning steps without requiring human-annotated reasoning chains or task-specific rewards.

Some approaches take automation even further. In AlphaMath (Chen et al., 2024), researchers

propose an even more radical approach: deriving pseudo-process supervision directly from outcome supervision, thereby eliminating the need for stepwise labels altogether. More structured methods have also been developed, such as Tree-PLV (He et al., 2024), which learns preferences over trees of reasoning trajectories automatically constructed via a best-first search algorithm. Building on this trend, rStar-Math (Guan et al., 2025) and Qwen2.5-Math PRM (Zhang et al., 2025k) adopt self-evolutionary and consensus-filtering strategies respectively to create massive reasoning datasets, while EpicPRM (Sun et al., 2025b) focuses on balancing precision and scale in constructing process-supervised training data.

To improve robustness, SCAN (Ding et al., 2025) introduces a self-denoising annotation framework that automatically detects and corrects noisy labels, and Wang et al. (2025c) proposes a data augmentation strategy based on node merging in the tree structure.

Collectively, these works showcase the promise of automated pipelines: they enable unprecedented scale and efficiency, though they must carefully address error propagation, verifier limitations, and potential misalignment with human reasoning preferences.

2.3 Semi-automated Approaches

Between these two extremes, a growing number of works adopt semi-automated approaches, blending selective human input with scalable automated expansion. In multimodal reasoning, this pattern is especially pronounced: VRPRM (Chen et al., 2025f) and Athena (Wang et al., 2025b) both construct PRM datasets by starting with limited human-curated reasoning steps and then expanding them with automated verification or synthetic generation, significantly improving data efficiency. ViL-Bench (Tu et al., 2025) and VisualPRM (Wang et al., 2025f) adopt similar strategies in vision-language reasoning, mixing curated samples with large-scale synthetic data to create comprehensive benchmarks.

In more specialized domains, MedS^3 (Jiang et al., 2025) adopts a self-evolved "slow thinking" paradigm for medical reasoning: it starts from around 8,000 human-curated examples and then automatically expands them via MCTS-based exploration and rule-verifiable trajectory generation, greatly reducing manual workload while retaining domain reliability. Beyond single-domain settings, VersaPRM (Zeng et al., 2025) generates synthetic reasoning data across multiple domains primarily via auto-labeling, with a small-scale manual evaluation conducted to verify the quality of the auto-labeled data.

Practical task-oriented applications also rely on hybrid pipelines. Web-Shepherd (Chae et al., 2025) supervises web navigation reasoning traces by mixing human oversight with automatic checks, while GUI-Shepherd (Chen et al., 2025a) builds the PRM dataset via a dual-pipeline strategy combining diverse trajectories with hybrid human-GPT annotations. Finally, ActPRM (Duan et al., 2025) exemplifies active learning in PRM training, selectively querying human annotators only when automated signals are uncertain, thereby reducing labeling costs without sacrificing supervision quality.

These hybrid methods illustrate that carefully combining human anchors with automated pipelines not only mitigates the weaknesses of each approach but also opens up broader applications in domains where neither purely human nor purely automated supervision is sufficient.

3 How to Build PRMs

In this section, we answer the question of "how to build PRMs" and categorize PRM training works into four classes: Discriminative PRMs, Generative PRMs, Implicit PRMs, and Other Architectures. Furthermore, we provide detailed discussions of representative methods in each category.

3.1 Discriminative PRMs.

A *discriminative* PRM learns a scoring function over intermediate reasoning states to predict per-step correctness, plausibility, or progress. Given an input x and a partial solution $s_{1:t}$, the model outputs a scalar score as Eq. 1 shows.

$$r_t = \sigma(f_\theta(x, s_{1:t})) \in (0, 1) \tag{1}$$

Pointwise loss. The score r_t can be trained with standard pointwise objectives. Here σ is the sigmoid function, and f_θ denotes the discriminative PRMs. With binary labels $y_t \in \{0, 1\}$ or soft labels $y_t \in [0, 1]$, one typically uses either binary cross-entropy (BCE) or mean squared error (MSE):

$$\mathcal{L}^{\text{BCE}}_{\text{point}} = \mathbb{E}\big[-y_t \log r_t - (1-y_t)\log(1-r_t)\big], \tag{2}$$

$$\mathcal{L}^{\text{MSE}}_{\text{point}} = \mathbb{E}\big[(r_t - y_t)^2\big]. \tag{3}$$

Pairwise (preference) loss. Alternatively, discriminative PRMs can be trained on *relative* preferences between two candidate steps or partial traces u and v. The model predicts the probability that u is preferred to v:

$$\mathbb{P}_\theta(u \succ v) = \sigma\big(f_\theta(u) - f_\theta(v)\big), \tag{4}$$

and minimizes a pairwise (preference) loss such as:

$$\mathcal{L}_{\text{pair}} = \mathbb{E}\big[-\log \mathbb{P}_\theta(u \succ v)\big], \tag{5}$$

which is analogous to the Direct Preference Optimization (DPO) objective used in RLHF.

Discriminative PRMs, viewed as the foundational training paradigm in the history of process-level reward models, have inspired lots of works. DreamPRM (Cao et al., 2025b) alternately trains the PRM and domain weights through a bi-level strategy to generalize across multimodal tasks; PQM (Li and Li, 2024) recasts PRM as a Q-value ranking problem, aligning rewards by relative ordering; ER-PRM (Zhang et al., 2024) injects entropy regularization into the reward objective to avoid overconfident predictions and improve calibration; EDU-PRM (Cao et al., 2025a) uses entropy-based uncertainty sampling and weighting to focus training on ambiguous or difficult reasoning steps; Q-RM (Chen et al., 2025b) introduces token-level discriminative loss to provide finer-grained feedback on intermediate tokens; BiPRM (Zhang et al., 2025e) seamlessly integrates a parallel right-to-left (R2L) evaluation stream with the conventional L2R flow, allowing later reasoning steps to real-time assist in assessing earlier ones;R-PRM (She et al., 2025) designs a loss function that favors logical and structural consistency across reasoning steps; BiRM (Chen et al., 2025e) not only evaluates the correctness of previous steps, but also models the probability of future success; CoLD (Zheng et al., 2025) uses counterfactual guidance to mitigate length bias in reward scoring; and ProgRM (Zhang et al., 2025a) defines dynamic "progress rewards" that proportionally align process rewards with the degree of task completion.

3.2 Generative PRMs.

A *generative* PRM operates in two stages: it first generates a verification or critique chain z_t ("think"), and then judges or scores the original reasoning step based on that chain ("judge"). Concretely, one can write:

$$\begin{aligned} z_t &\sim p_\phi\big(z_t \mid x, s_{1:t}\big) \\ r_t &= h_\psi\big(x, s_{1:t}, z_t\big), \end{aligned} \tag{6}$$

where p_ϕ is the generative verifier or critic model, and h_ψ is a scoring head that maps the generated chain and the step history to a step-level reward r_t. A plausible joint training objective combines a likelihood loss for the verification chain and a supervision term for the step-level reward:

$$\mathcal{L}_{\text{gen}} \;=\; -\log p_\phi\big(z_t^\star \mid x, s_{1:t}\big) \;+\; \lambda\,\mathrm{BCE}\big(r_t, y_t\big), \tag{7}$$

where $z_t^\star$ is a reference (e.g., human or oracle) critique chain, and y_t is the ground-truth (or soft) label for the step.

In many works, h_ψ is simply the confidence of the answer logits. Assume token indices k_{yes} and k_{no} correspond to "yes" and "no" respectively. Then define r_t as the softmax score:

$$r_t = \frac{\exp(q_{k_{\text{yes}}})}{\exp(q_{k_{\text{yes}}}) + \exp(q_{k_{\text{no}}})}. \tag{8}$$

This generative PRM paradigm helps the reward model maintain long reasoning chains (i.e., extended "thinking") and better understand the semantics of the input. ThinkPRM (Lee et al., 2025) uses an internal "thinking" loop to simulate generative reflection and enable dynamic reasoning. GenRM (Zhang et al., 2025f) introduces chain-of-thought at inference and uses voting to pick the highest-scoring reasoning chain to improve consistency. GenPRM (Zhao et al., 2025) applies generative computation scaling at test time to boost the stability of reward predictions. GRAM-R² (Wang et al., 2025a) self-trains a generative foundation reward model that evolves its own reasoning and reward logic. Process-based Self-Rewarding Language Models (Zhang et al., 2025h) allow the model to both generate and assess its own reasoning chains, closing the loop between reasoning and reward. Test-Time Scaling with Reflective Generative Model (Wang et al., 2025g) expands inference-time generative capacity and applies reflection to refine reward prediction. GM-PRM (Zhang et al., 2025b) is the first multimodal generative PRM, supporting chain generation in multimodal mathematical reasoning tasks. rStar-Math (Guan et al., 2025) strengthens smaller models' reasoning by evolving deep thinking through self-evolution in its internal reasoning architecture.

3.3 Implicit PRMs

The above discriminative and generative PRM methods all rely on explicit supervision signals derived from annotated reasoning steps; in contrast, implicit PRMs aim to infer fine-grained rewards without step-level labels, by leveraging weaker or indirect supervision such as outcome feedback, model self-evaluation, or consistency constraints. Implicit PRM extracts step rewards from unlabeled trajectories; FreePRM (Sun et al., 2025a) trains a reward model without ground-truth process labels by pseudo-labeling via outcome correctness; Self-PRM (Feng et al., 2025a) shows that LLMs under RL training can internally induce a PRM-style self-rewarding capability; SP-PRM (Xie et al., 2025a) transfers reasoning knowledge from an outcome reward model (ORM) into process reward modeling to reduce label dependency; SPARE (Rizvi et al., 2025) uses one-shot reference guidance to automatically generate supervision signals for intermediate steps; Universal PRM (AURORA) (Tan et al., 2025) employs ensemble prompting and reverse verification to produce domain-agnostic self-supervised reward signals; and Process-based Self-Rewarding Language Models let the model generate and evaluate its own reasoning chain, closing the loop for self-supervision.

3.4 Other Architectural Innovations

Other architectures in the PRM landscape emphasize innovations in model structure, reasoning representations, or system frameworks rather than new loss functions or supervision schemes. For example, GraphPRM (Peng et al., 2025) casts reasoning as a graph of steps and learns structured dependencies among them; ASPRM (AdaptiveStep) (Liu et al., 2025) dynamically adjusts the granularity of reasoning steps based on model confidence; Reward-SQL (Zhang et al., 2025j) builds a structured process reward model tailored to the Text-to-SQL domain; RetrievalPRM (Zhu et al., 2025) integrates external retrieval to ground reward predictions and improve cross-task generalization; OpenPRM (Zhang et al., 2025c) organizes reward judgments into an open preference tree, supporting branching and domain flexibility; MM-PRM (Du et al., 2025) provides a unified multimodal PRM architecture and open implementation; Multilingual PRM (Wang et al., 2025e) addresses cross-language CoT transfer through representational mapping across languages; PathFinder-PRM (Pala et al., 2025a) employs a hierarchical error-aware architecture to distinguish and reward different types of reasoning errors; and Hierarchical Reward Model (HRM) (Wang et al., 2025d) proposes layered reward structures aligned with multi-level reasoning abstractions.

4 How to Use PRMs

In this section, we discuss how to use PRMs and organize their usage into two main paradigms: Test-Time Scaling and Reinforcement Learning for Policy Learning. We further provide detailed discussions of representative methods and developments within each paradigm, highlighting how PRMs guide inference, search, and policy learning through fine-grained step-level feedback.

4.1 Test-Time Scaling

Test-time scaling aims to improve model performance not by enlarging model size but by strategically allocating computation during inference—via candidate sampling, re-ranking, or guided search. PRMs are central to this process, providing fine-grained evaluation of intermediate reasoning steps and trajectories to guide test-time computation.

Early work used PRMs primarily as re-rankers. Studies such as Lightman et al. (2023); Wang et al. (2023, 2025f,b); Zheng et al. (2025) showed that Best-of-N re-ranking with PRM scores consistently improves final performance, validating PRMs as reliable test-time evaluators. Building on this foundation, PRMs evolved into generative verifiers. GenPRM (Zhao et al., 2025) introduced verification-by-generation, producing reasoning or code checks before scoring candidates. ThinkPRM (Snell et al., 2024) fine-tunes long chain-of-thought verifiers with limited process-level labels, enhancing scaling under Best-of-N and beam search. Kim et al. (2025) formalized reasoning-oriented evaluation as a mechanism for allocating test-time compute more effectively, positioning PRMs as flexible controllers of inference resources.

Parallel efforts integrated PRMs into search and decoding algorithms. PRM-BAS (Hu et al., 2025a) embedded PRMs into beam annealing search, pruning low-quality candidates to improve efficiency. CodePRM (Li et al., 2025a) implemented a Generate–Verify–Refine pipeline, using PRMs to detect and correct faulty intermediate code steps. Web-Shepherd (Chae et al., 2025) filtered web-agent trajectories, while other approaches com-

bined PRMs with MCTS or retrieval-augmented reasoning (Chan et al., 2025; Ma et al., 2025; Chen et al., 2025d). Safety-aware scaling was addressed by SAFFRON-1, which reduced costly PRM calls and introduced caching mechanisms to ensure robust, efficient inference under adversarial conditions.

Finally, refinements targeted step-level granularity and adaptivity. AdaptiveStep (Liu et al., 2025) dynamically partitions reasoning into finer steps based on confidence, producing sharper PRM judgments. SP-PRM (Xie et al., 2025b) extended reward-guided search strategies across multiple granularity levels, from tokens to full responses, enhancing both precision and flexibility.

Together, these developments trace a clear trajectory: from static PRM-based re-ranking, through generative verification and search integration, to adaptive step-level refinements and safety-aware scaling, transforming PRMs into dynamic, scalable controllers of inference.

4.2 RL for Policy Learning

The use of process reward models (PRMs) within reinforcement learning (RL) has become a promising direction for aligning language models with fine-grained reasoning quality. Traditional RL relies on outcome-only supervision, which is sparse and often misaligned with intermediate reasoning steps. By contrast, PRMs provide dense step-level or trajectory-level feedback that can be integrated into RL training loops, offering more stable credit assignment and faster policy learning.

Early explorations established that PRMs could directly replace sparse correctness-based signals with fine-grained supervision during RL. Math-Shepherd (Wang et al., 2023) trained an automatic verifier that scores each intermediate step in math reasoning and used those scores as rewards for PPO, allowing the policy to learn from abundant intermediate feedback when final answers are rare. In a similar vein, Dai et al. (2024) demonstrated how line-level PRM signals could be injected into RL training, overcoming the limitations of outcome-only feedback from unit tests and enabling policies to improve across long coding trajectories. Extending this idea to practical domains, Reward-SQL (Zhang et al., 2025j) integrated stepwise PRMs into an online RL loop, showing that process-level signals are especially valuable in text-to-SQL generation, while ReasonRAG (Zhang et al., 2025h) applied PRM-guided RL to retrieval-augmented generation agents. Together, these works show that PRMs can serve as actionable dense rewards that significantly improve RL training across reasoning-heavy tasks.

Building on this foundation, several studies refined the formulation of PRM signals within RL objectives. PAV (Setlur et al., 2024) reframed step-level PRM outputs as advantage-like progress indicators, providing dense step-level rewards for RL training of policy models. ER-PRM (Zhang et al., 2024) introduced an entropy-regularized framework that embeds PRM rewards into KL-constrained RL objectives, stabilizing training while preserving exploration. PURE (Cheng et al., 2025) addressed a fundamental credit-assignment challenge, arguing that summing PRM rewards encourages reward hacking and instead proposing a min-form objective that integrates PRM signals into RL updates more robustly. Q-RM (Chen et al., 2025c) advanced token-level supervision by modeling Q-values over tokens and using them directly as rewards during RL optimization. CAPO (Xie et al., 2025c) introduces verifiable generative credit assignment to produce reliable step-level rewards for RL training of policy models. These verifiable rewards replace sparse outcome signals, improving exploration and sample efficiency. These innovations highlight that beyond having PRM feedback, the way PRM outputs are incorporated into RL loss functions critically affects training stability and effectiveness. He et al. (2025b) introduces a generative, thought-level PRM that assigns reliable grouped step-level rewards for RL training of policy models integrating with an off-policy algorithm and adaptive reward balancing. Meanwhile, PROF (Ye et al., 2025) ranks and filters responses based on process–outcome consistency between PRMs and ORMs, removing samples where reasoning and results conflict to reduce noisy gradients. It further maintains balanced training by separately ranking correct and incorrect responses, and can be seamlessly integrated with RL methods such as GRPO (Shao et al., 2024).

In parallel, domain-specific efforts such as GraphPRM (Peng et al., 2025) used PRM-guided preference optimization to improve reasoning over graph reasoning problems, while Agent-PRM (Choudhury, 2025) integrated PRMs into an actor–critic loop for LLM-based agents, showing how step-level critics can accelerate RL in interactive settings. These results demonstrate that PRMs can make RL training more robust across diverse

reasoning tasks.

Broader frameworks have emerged to consolidate and scale these practices. OpenR (Wang et al., 2024) provides an open-source infrastructure that systematizes the integration of PRMs into both offline and online RL pipelines, offering recipes for PRM-guided training across reasoning benchmarks.

5 Downstream Application

Process Reward Models (PRMs) are increasingly adopted across diverse reasoning and decision-making tasks. Below we summarize representative application areas.

Math PRMs validate algebraic and logical steps to ensure multi-step derivation soundness (Zhou et al., 2025a; Uesato et al., 2022; Lightman et al., 2023; Wang et al., 2023), capturing symbolic and arithmetic errors to improve final correctness (Li et al., 2024; He et al., 2024; Pala et al., 2025b). They support scalable supervision and automated feedback for grading, tutoring, and proof validation with reduced human effort (Chen et al., 2024; Setlur et al., 2024; Zhao et al., 2025; Sun et al., 2025b).

Code For code generation, PRMs assess partial programs with execution or proxy testing feedback (Li et al., 2025a; Dai et al., 2024), rewarding syntactic validity and semantic consistency. They also verify query construction and patches in text-to-SQL and software engineering (Zhang et al., 2025h; Gandhi et al., 2025), improving robustness.

Multimodal In multimodal reasoning, PRMs check visual-text coherence (Hu et al., 2025a; Du et al., 2025; Chen et al., 2025f; Tu et al., 2025; Wang et al., 2025f), rerank reasoning traces, and select grounded explanations to enhance interpretability and factual consistency.

Text For text tasks, PRMs refine multi-step reasoning by evaluating partial translations (Feng et al., 2025b) and scoring intermediate hops in QA and retrieval-augmented reasoning (Chan et al., 2025; Chen et al., 2025d), improving coherence and factual reliability.

Robotics PRMs decompose long-horizon manipulation or navigation into subgoal rewards (Lu et al., 2025), providing dense feedback that accelerates policy learning and stabilizes control.

Agents In interactive agents, PRMs act as trajectory critics (Choudhury, 2025; Hu et al., 2025b; Chae et al., 2025; Zhang et al., 2025i,a; Chen et al., 2025a; Xi et al., 2025; Yang et al., 2025c), rewarding meaningful progress, pruning dead ends, and improving safety during inference.

Industry In high-stakes areas like medicine and finance, PRMs enforce verifiable, evidence-based reasoning (Jiang et al., 2025; Zhou et al., 2025b), promoting reliability and risk-sensitive decision making.

Multi-domain Recent studies explore generalizable PRMs that transfer process supervision across tasks (Cao et al., 2025b; Wang et al., 2025b; Zhang et al., 2025c; Zeng et al., 2025; Rizvi et al., 2025; Xie et al., 2025b; Ding et al., 2025; Tan et al., 2025), pointing toward universal, cross-domain reasoning evaluators.

6 Benchmark

Recent work has introduced a range of benchmarks to evaluate PRMs at the step level, differing in scale, domain, and evaluation focus.

For mathematical reasoning, PRMBench (Song et al., 2025) and ProcessBench (Zheng et al., 2024) offer complementary views. PRMBench provides over 6,000 problems with 80,000 step annotations and multidimensional labels (e.g., simplicity, soundness, sensitivity), while ProcessBench targets competition-level tasks, emphasizing earliest-error detection for precise symbolic reasoning.

Reasoning-structure evaluation is addressed by Socratic-PRMBench (Li et al., 2025b), which groups nearly three thousand flawed trajectories into six error patterns, enabling analysis of generalization across reasoning styles.

For multimodal tasks, ViLBench (Tu et al., 2025) compares PRMs with outcome models in vision-language reasoning, VisualProcessBench (Wang et al., 2025f) provides human-labeled multimodal errors, and MPBench (Xu et al., 2025) extends coverage to multiple tasks, assessing step correctness, answer aggregation, and reasoning-guided search.

Long-horizon decision-making is tested by WebRewardBench (Chae et al., 2025), built on the WebPRM Collection with forty thousand step-level preference pairs, evaluating clicks, form entries, and navigation steps in web agents.

Robustness and universality are explored by GSM-DC (Yang et al., 2025b), which injects dis-

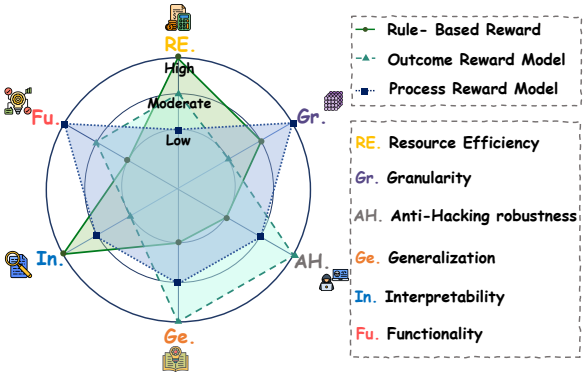

Figure 2: Comparative Analysis of Three Reward Mechanisms Across Six Evaluation Aspects

tractors to test resilience, and UniversalBench (Tan et al., 2025), which evaluates trajectories across diverse policy distributions for cross-distribution generalization and reproducibility.

7 Discussion

To better compare the different forms of reward acquisition, including rule-based rewards, outcome reward models (ORMs), and process reward models (PRMs), we design a six-aspect evaluation scheme covering resource efficiency, granularity, anti-hacking robustness, generalization, interpretability, and functionality. This perspective provides a systematic and balanced basis for assessing how each reward mechanism performs across theoretical soundness, practical applicability, and scalability, as illustrated in Figure 2.

Resource Efficiency Rule-based rewards stand out as the most economical approach, as they rely purely on manually defined rules without requiring additional data labeling or model training. ORMs require moderate resources, depending on final outcome labels and a single-stage training process. In contrast, PRMs are far more costly because ORMs only label the final outcome, whereas PRMs require correctness labels for every intermediate step. As noted in Section 2.1, this necessitates expensive step-wise human annotation (e.g., PRM800K (Lightman et al., 2023)) or complex automated pipelines (Section 2.2). Given that benchmarks like ProcessBench (Zheng et al., 2024) and PRMBench (Song et al., 2025) contain an average of 7.1 and 13.4 steps respectively, the annotation workload for PRMs is naturally several times higher than that of ORMs.

Granularity The high rating for PRMs in terms of granularity is structural rather than subjective, as defined by the mathematical formulation in Section 3.1 (Eq. 1). This enables step-specific error localization, whereas ORMs operate at only a single outcome level. Essentially, the granularity of a PRM is inherently multiplied by the number of steps in a solution. Conversely, the granularity of rule-based rewards is entirely determined by handcrafted design, which can vary from coarse to fine depending on how the rules are specified (Gunjal et al., 2025).

Anti-Hacking Robustness ORMs exhibit the strongest resistance to reward hacking, grounded in their reliance on ground-truth verification which is inherently resistant to manipulation. In contrast, PRMs are more susceptible to length hacking or verbosity bias due to high variance in step-wise optimization (Zheng et al., 2026). To quantify this, we conducted a stability analysis on the PRM800K dataset. We observed that the standard deviation (SD) of token length at the step level is 71.7, significantly higher than the trajectory-level SD of 50.6. This greater instability ($71.7 > 50.6$) indicates that step-level signals are noisier and less constrained, allowing models to more easily hack the reward by generating verbose but vacuous intermediate steps. Rule-based rewards remain the most prone to exploitation if predefined rules are mis-specified.

Generalization ORMs show a clear advantage in generalization, as their outcome-centric formulation utilizes task-agnostic labels that are easily transferred across domains. PRMs demonstrate more limited generalization because they often require defining domain-specific step granularities. For instance, the step definitions for mathematical derivations differ fundamentally from those for code execution traces, necessitating frequent readaptation for new tasks as discussed in Section 5. Rule-based systems exhibit the poorest generalization, as their logic must be carefully re-engineered for every new environment.

Interpretability Interpretability varies significantly across mechanisms. Rule-based rewards offer the highest transparency, as their evaluation logic is explicitly encoded. Conversely, ORMs suffer from low interpretability, operating as black boxes that provide coarse judgments without explaining specific errors. PRMs bridge this gap by offering fine-grained, step-wise supervision

for precise error localization, a capability empirically supported by PRM800K. Recent innovations further enhance this transparency: generative PRMs like ThinkPRM (Lee et al., 2025) and GenRM (Zhang et al., 2025f) produce natural language justifications, while benchmarks like Socratic-PRMBench (Li et al., 2025b) provide semantic clarity by categorizing specific reasoning error patterns.

Functionality Finally, PRMs are the most versatile. Linked to our discussion on Test-Time Scaling (Section 4.1), PRMs consistently demonstrate superior capabilities in guiding search, such as Tree Search, compared to ORMs. Furthermore, PRMs offer greater flexibility in RL training (Section 4.2). Because they provide both step-level and trajectory-level signals, they support step-wise credit assignment and trajectory-wise reward shaping. ORMs, limited to a single final-outcome reward, cannot provide the same level of fine-grained supervision during policy optimization. Rule-based rewards, while straightforward, remain functionally restricted as they lack adaptability beyond their original design.

Beyond these comparative dimensions, the development of PRMs faces several profound conceptual and systemic challenges. We provide a critical exploration of these frontiers, including cognitive scalability (Section B.1), automated supervision risks (Section B.2), the tension of granularity (Section B.3), and the proxy-reward gap (Section B.4).

8 Conclusion

Process Reward Models (PRMs) shift reasoning alignment from coarse outcome judgments to fine-grained, step-level feedback, forming a closed loop of *data generation*, *model training*, and *usage* that continually improves reasoning quality. Our survey organizes this field around how to generate process data, build PRMs, and use them for test-time scaling and reinforcement learning, while summarizing benchmarks and applications across math, code, multimodal tasks, robotics, and other domains.

Key challenges ahead include reducing annotation cost via robust automatic supervision, improving cross-domain generalization, integrating PRMs with agentic planning and memory, and establishing standardized evaluation protocols. Addressing these will advance safer, more interpretable, and broadly applicable reasoning systems.

9 Limitations

While this survey aims to provide a broad and systematic view of Process Reward Models (PRMs), it also has several natural limitations. First, our taxonomy follows the *data–model–usage* loop and thus simplifies or abstracts some hybrid methods; certain approaches may span multiple categories and are discussed only under their primary aspect. Second, benchmark and application summaries are selective rather than comprehensive. We highlight representative resources but cannot guarantee complete inclusion of all task-specific datasets or proprietary evaluation suites. Despite these boundaries, we believe our synthesis offers a clear conceptual map and can serve as a starting point for exploring, extending, and systematizing PRM research.

Acknowledgments

The Shanghai Jiao Tong University team is partially supported by National Key RD Program of China (2022ZD0114804), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102) and National Natural Science Foundation of China (624B2096, 62322603, 72542012, 72595872).

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Lang Cao, Renhong Chen, Yingtian Zou, Chao Peng, Wu Ning, Huacong Xu, Qian Chen, Yuxian Wang, Peishuo Su, Mofan Peng, Zijie Chen, and Yitong Li. 2025a. More bang for the buck: Process reward modeling with entropy-driven uncertainty. *Preprint*, arXiv:2503.22233.

Qi Cao, Ruiyi Wang, Ruiyi Zhang, Sai Ashish Somayajula, and Pengtao Xie. 2025b. Dreamprm: Domain-reweighted process reward model for multimodal reasoning. *arXiv preprint arXiv:2505.20241*.

Hyungjoo Chae, Sunghwan Kim, Junhee Cho, Seungone Kim, Seungjun Moon, Gyeom Hwangbo, Dongha Lim, Minjin Kim, Yeonjun Hwang, Minju Gwak, and 1 others. 2025. Web-shepherd: Advancing prms for reinforcing web agents. *arXiv preprint arXiv:2505.15277*.

Huacan Chai, Zijie Cao, Maolin Ran, Yingxuan Yang, Jianghao Lin, Xin Peng, Hairui Wang, Renjie Ding, Ziyu Wan, Muning Wen, and 1 others. 2025. Parl-mt: Learning to call functions in multi-turn conversation with progress awareness. *arXiv preprint arXiv:2509.23206*.

Chi-Min Chan, Chunpu Xu, Junqi Zhu, Jiaming Ji, Donghai Hong, Pengcheng Wen, Chunyang Jiang, Zhen Ye, Yaodong Yang, Wei Xue, and 1 others. 2025. Boosting policy and process reward models with monte carlo tree search in open-domain qa. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7433–7451.

Cong Chen, Kaixiang Ji, Hao Zhong, Muzhi Zhu, Anzhou Li, Guo Gan, Ziyuan Huang, Cheng Zou, Jiajia Liu, Jingdong Chen, and 1 others. 2025a. Gui-shepherd: Reliable process reward and verification for long-sequence gui tasks. *arXiv preprint arXiv:2509.23738*.

Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. 2024. Alphamath almost zero: process supervision without process. *Advances in Neural Information Processing Systems*, 37:27689–27724.

Hongzhan Chen, Tao Yang, Shiping Gao, Ruijun Chen, Xiaojun Quan, Hongtao Tian, and Ting Yao. 2025b. Discriminative policy optimization for token-level reward models. *Preprint*, arXiv:2505.23363.

Hongzhan Chen, Tao Yang, Shiping Gao, Ruijun Chen, Xiaojun Quan, Hongtao Tian, and Ting Yao. 2025c. Discriminative policy optimization for token-level reward models. *arXiv preprint arXiv:2505.23363*.

Jiabei Chen, Guang Liu, Shizhu He, Kun Luo, Yao Xu, Jun Zhao, and Kang Liu. 2025d. Search-in-context: Efficient multi-hop qa over long contexts via monte carlo tree search with dynamic kv retrieval. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 26443–26455.

Wenxiang Chen, Wei He, Zhiheng Xi, Honglin Guo, Boyang Hong, Jiazheng Zhang, Rui Zheng, Nijun Li, Tao Gui, Yun Li, and 1 others. 2025e. Better process supervision with bi-directional rewarding signals. *arXiv preprint arXiv:2503.04618*.

Xinquan Chen, Bangwei Liu, Xuhong Wang, Yingchun Wang, and Chaochao Lu. 2025f. Vrprm: Process reward modeling via visual reasoning. *arXiv preprint arXiv:2508.03556*.

Jie Cheng, Ruixi Qiao, Lijun Li, Chao Guo, Junle Wang, Gang Xiong, Yisheng Lv, and Fei-Yue Wang. 2025. Stop summation: Min-form credit assignment is all process reward model needs for reasoning. *arXiv preprint arXiv:2504.15275*.

Sanjiban Choudhury. 2025. Process reward models for llm agents: Practical framework and directions. *arXiv preprint arXiv:2502.10325*.

Ning Dai, Zheng Wu, Renjie Zheng, Ziyun Wei, Wenlei Shi, Xing Jin, Guanlin Liu, Chen Dun, Liang Huang, and Lin Yan. 2024. Process supervision-guided policy optimization for code generation. *arXiv preprint arXiv:2410.17621*.

Yuyang Ding, Xinyu Shi, Juntao Li, Xiaobo Liang, Zhaopeng Tu, and Min Zhang. 2025. Scan: Self-denoising monte carlo annotation for robust process reward learning. *arXiv preprint arXiv:2509.16548*.

Lingxiao Du, Fanqing Meng, Zongkai Liu, Zhixiang Zhou, Ping Luo, Qiaosheng Zhang, and Wenqi Shao. 2025. Mm-prm: Enhancing multimodal mathematical reasoning with scalable step-level supervision. *arXiv preprint arXiv:2505.13427*.

Keyu Duan, Zichen Liu, Xin Mao, Tianyu Pang, Changyu Chen, Qiguang Chen, Michael Qizhe Shieh, and Longxu Dou. 2025. Efficient process reward model training via active learning. *arXiv preprint arXiv:2504.10559*.

Zhangying Feng, Qianglong Chen, Ning Lu, Yongqian Li, Siqi Cheng, Shuangmu Peng, Duyu Tang, Shengcai Liu, and Zhirui Zhang. 2025a. Is prm necessary? problem-solving rl implicitly induces prm capability in llms. *Preprint*, arXiv:2505.11227.

Zhaopeng Feng, Jiahan Ren, Jiayuan Su, Jiamei Zheng, Zhihang Tang, Hongwei Wang, and Zuozhu Liu. 2025b. Mt-rewardtree: A comprehensive framework for advancing llm-based machine translation via reward modeling. *arXiv preprint arXiv:2503.12123*.

Shubham Gandhi, Jason Tsay, Jatin Ganhotra, Kiran Kate, and Yara Rizk. 2025. When agents go astray: Course-correcting swe agents with prms. *arXiv preprint arXiv:2509.02360*.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. 2025. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *arXiv preprint arXiv:2507.17746*.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, and 1 others. 2025a. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*.

Mingqian He, Yongliang Shen, Wenqi Zhang, Zeqi Tan, and Weiming Lu. 2024. Advancing process verification for large language models via tree-based preference learning. *arXiv preprint arXiv:2407.00390*.

Tao He, Rongchuan Mu, Lizi Liao, Yixin Cao, Ming Liu, and Bing Qin. 2025b. Good learners think their thinking: Generative prm makes large reasoning model more efficient math learner. *arXiv preprint arXiv:2507.23317*.

Pengfei Hu, Zhenrong Zhang, Qikai Chang, Shuhang Liu, Jiefeng Ma, Jun Du, Jianshu Zhang, Quan Liu, Jianqing Gao, Feng Ma, and 1 others. 2025a.

Prm-bas: Enhancing multimodal reasoning through prm-guided beam annealing search. *arXiv preprint arXiv:2504.10222*.

Zhiyuan Hu, Shiyun Xiong, Yifan Zhang, See-Kiong Ng, Anh Tuan Luu, Bo An, Shuicheng Yan, and Bryan Hooi. 2025b. Guiding vlm agents with process rewards at inference time for gui navigation. *arXiv preprint arXiv:2504.16073*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Shuyang Jiang, Yusheng Liao, Zhe Chen, Ya Zhang, Yanfeng Wang, and Yu Wang. 2025. Meds ^{3}: Towards medical slow thinking with self-evolved soft dual-sided process supervision. *arXiv preprint arXiv:2501.12051*.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Sarkar Snigdha Sarathi Das, and Rui Zhang. 2025. Generalizable process reward models via formally verified training data. *arXiv preprint arXiv:2505.15960*.

Seungone Kim, Ian Wu, Jinu Lee, Xiang Yue, Seongyun Lee, Mingyeong Moon, Kiril Gashteovski, Carolin Lawrence, Julia Hockenmaier, Graham Neubig, and 1 others. 2025. Scaling evaluation-time compute with reasoning models as process evaluators. *arXiv preprint arXiv:2503.19877*.

Dong Bok Lee, Seanie Lee, Sangwoo Park, Minki Kang, Jinheon Baek, Dongki Kim, Dominik Wagner, Jiongdao Jin, Heejun Lee, Tobias Bocklet, and 1 others. 2025. Rethinking reward models for multi-domain test-time scaling. *arXiv preprint arXiv:2510.00492*.

Mukai Li, Qingcheng Zeng, Tianqing Fang, Zhenwen Liang, Linfeng Song, Qi Liu, Haitao Mi, and Dong Yu. 2026. Verified critical step optimization for llm agents. *arXiv preprint arXiv:2602.03412*.

Qingyao Li, Xinyi Dai, Xiangyang Li, Weinan Zhang, Yasheng Wang, Ruiming Tang, and Yong Yu. 2025a. Codeprm: Execution feedback-enhanced process reward model for code generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8169–8182.

Ruosen Li, Ziming Luo, and Xinya Du. 2024. Fine-grained hallucination detection and mitigation in language model mathematical reasoning.

Wendi Li and Yixuan Li. 2024. Process reward model with q-value rankings. *arXiv preprint arXiv:2410.11287*.

Xiang Li, Haiyang Yu, Xinghua Zhang, Ziyang Huang, Shizhu He, Kang Liu, Jun Zhao, Fei Huang, and Yongbin Li. 2025b. Socratic-prmbench: Benchmarking process reward models with systematic reasoning patterns. *arXiv preprint arXiv:2505.23474*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Jianghao Lin, Yuanyuan Shi, Xin Peng, Renjie Ding, Hairui Wang, Yuxuan Peng, Bizhe Bai, Weixi Song, Fengshuo Bai, Huacan Chai, and 1 others. 2025. Toolprm: Fine-grained inference scaling of structured outputs for function calling. *arXiv preprint arXiv:2510.14703*.

Yuliang Liu, Junjie Lu, Zhaoling Chen, Chaofeng Qu, Jason Klein Liu, Chonghan Liu, Zefan Cai, Yunhui Xia, Li Zhao, Jiang Bian, and 1 others. 2025. Adaptivestep: Automatically dividing reasoning step through model confidence. *arXiv preprint arXiv:2502.13943*.

Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. 2025. Vla-rl: Towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv preprint arXiv:2505.18719*.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, and 1 others. 2024. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592*.

Ruilin Luo, Zhuofan Zheng, Yifan Wang, Xinzhe Ni, Zicheng Lin, Songtao Jiang, Yiyao Yu, Chufan Shi, Ruihang Chu, Jin Zeng, and 1 others. 2025. Ursa: Understanding and verifying chain-of-thought reasoning in multimodal mathematics. *arXiv preprint arXiv:2501.04686*.

Jie Ma, Shihao Qi, Rui Xing, Ziang Yin, Bifan Wei, Jun Liu, and Tongliang Liu. 2025. From static to dynamic: Adaptive monte carlo search for mathematical process supervision. *arXiv preprint arXiv:2509.24351*.

Roy Miles, Aysim Toker, Andreea-Maria Oncescu, Songcen Xu, Jiankang Deng, and Ismail Elezi. 2026. Test-time scaling with diffusion language models via reward-guided stitching. *arXiv preprint arXiv:2602.22871*.

Shuo Nie, Hexuan Deng, Chao Wang, Ruiyu Fang, Xuebo Liu, Shuangyong Song, Yu Li, Min Zhang, and Xuelong Li. 2026. Stop rewarding hallucinated steps: Faithfulness-aware step-level reinforcement learning for small reasoning models. *arXiv preprint arXiv:2602.05897*.

Tej Deep Pala, Panshul Sharma, Amir Zadeh, Chuan Li, and Soujanya Poria. 2025a. Error typing for smarter rewards: Improving process reward models with error-aware hierarchical supervision. *Preprint*, arXiv:2505.19706.

Tej Deep Pala, Panshul Sharma, Amir Zadeh, Chuan Li, and Soujanya Poria. 2025b. Error typing for smarter rewards: Improving process reward models with error-aware hierarchical supervision. *arXiv preprint arXiv:2505.19706*.

Miao Peng, Nuo Chen, Zongrui Suo, and Jia Li. 2025. Rewarding graph reasoning process makes llms more generalized reasoners. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 2257–2268.

Md Imbesat Hassan Rizvi, Xiaodan Zhu, and Iryna Gurevych. 2025. Spare: Single-pass annotation with reference-guided evaluation for automatic process supervision and reward modelling. *arXiv preprint arXiv:2506.15498*.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. 2024. Rewarding progress: Scaling automated process verifiers for llm reasoning. *arXiv preprint arXiv:2410.08146*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Rituraj Sharma, Weiyuan Chen, Noah Provenzano, and Tu Vu. 2026. Prism: Pushing the frontier of deep think via process reward model-guided inference. *arXiv preprint arXiv:2603.02479*.

Shuaijie She, Junxiao Liu, Yifeng Liu, Jiajun Chen, Xin Huang, and Shujian Huang. 2025. R-prm: Reasoning-driven process reward modeling. *Preprint*, arXiv:2503.21295.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Mingyang Song, Zhaochen Su, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. 2025. Prmbench: A fine-grained and challenging benchmark for process-level reward models. *arXiv preprint arXiv:2501.03124*.

Lin Sun, Chuang Liu, Xiaofeng Ma, Tao Yang, Weijia Lu, and Ning Wu. 2025a. Freeprm: Training process reward models without ground truth process labels. *arXiv preprint arXiv:2506.03570*.

Wei Sun, Qianlong Du, Fuwei Cui, and Jiajun Zhang. 2025b. An efficient and precise training data construction framework for process-supervised reward model in mathematical reasoning. *arXiv preprint arXiv:2503.02382*.

Xiaoyu Tan, Tianchu Yao, Chao Qu, Bin Li, Minghao Yang, Dakuan Lu, Haozhe Wang, Xihe Qiu, Wei Chu, Yinghui Xu, and 1 others. 2025. Aurora: Automated training framework of universal process reward models via ensemble prompting and reverse verification. *arXiv preprint arXiv:2502.11520*.

Haoqin Tu, Weitao Feng, Hardy Chen, Hui Liu, Xianfeng Tang, and Cihang Xie. 2025. Vilbench: A suite for vision-language process reward modeling. *arXiv preprint arXiv:2503.20271*.

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

Chenglong Wang, Yongyu Mu, Hang Zhou, Yifu Huo, Ziming Zhu, Jiali Zeng, Murun Yang, Bei Li, Tong Xiao, Xiaoyang Hao, Chunliang Zhang, Fandong Meng, and Jingbo Zhu. 2025a. Gram-r^2: Self-training generative foundation reward models for reward reasoning. *Preprint*, arXiv:2509.02492.

Jun Wang, Meng Fang, Ziyu Wan, Muning Wen, Jiachen Zhu, Anjie Liu, Ziqin Gong, Yan Song, Lei Chen, Lionel M Ni, and 1 others. 2024. Openr: An open source framework for advanced reasoning with large language models. *arXiv preprint arXiv:2410.09671*.

Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2023. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*.

Shuai Wang, Zhenhua Liu, Jiaheng Wei, Xuanwu Yin, Dong Li, and Emad Barsoum. 2025b. Athena: Enhancing multimodal reasoning with data-efficient process reward models. *arXiv preprint arXiv:2506.09532*.

Teng Wang, Zhangyi Jiang, Zhenqi He, Shenyang Tong, Wenhan Yang, Yanan Zheng, Zeyu Li, Zifan He, and Hailei Gong. 2025c. Towards hierarchical multi-step reward models for enhanced reasoning in large language models. *arXiv preprint arXiv:2503.13551*.

Teng Wang, Zhangyi Jiang, Zhenqi He, Shenyang Tong, Wenhan Yang, Yanan Zheng, Zeyu Li, Zifan He, and Hailei Gong. 2025d. Towards hierarchical multi-step reward models for enhanced reasoning in large language models. *Preprint*, arXiv:2503.13551.

Weixuan Wang, Minghao Wu, Barry Haddow, and Alexandra Birch. 2025e. Demystifying multilingual chain-of-thought in process reward modeling. *arXiv preprint arXiv:2502.12663*.

Weiyun Wang, Zhangwei Gao, Lianjie Chen, Zhe Chen, Jinguo Zhu, Xiangyu Zhao, Yangzhou Liu, Yue Cao, Shenglong Ye, Xizhou Zhu, and 1 others. 2025f. Visualprm: An effective process reward model for multimodal reasoning. *arXiv preprint arXiv:2503.10291*.

Zhao Wang, Ziliang Zhao, and Zhicheng Dou. 2026. Prorag: Process-supervised reinforcement learning for retrieval-augmented generation. *arXiv preprint arXiv:2601.21912*.

Zixiao Wang, Yuxin Wang, Xiaorui Wang, Mengting Xing, Jie Gao, Jianjun Xu, Guangcan Liu, Chenhui Jin, Zhuo Wang, Shengzhuo Zhang, and 1 others. 2025g. Test-time scaling with reflective generative model. *arXiv preprint arXiv:2507.01951*.

Yunjia Xi, Jianghao Lin, Yongzhao Xiao, Zheli Zhou, Rong Shan, Te Gao, Jiachen Zhu, Weiwen Liu, Yong Yu, and Weinan Zhang. 2025. A survey of llm-based deep search agents: Paradigm, optimization, evaluation, and challenges. *arXiv preprint arXiv:2508.05668*.

Bin Xie, Bingbing Xu, Yige Yuan, Shengmao Zhu, and Huawei Shen. 2025a. From outcomes to processes: Guiding PRM learning from ORM for inference-time alignment. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19291–19307, Vienna, Austria. Association for Computational Linguistics.

Bin Xie, Bingbing Xu, Yige Yuan, Shengmao Zhu, and Huawei Shen. 2025b. From outcomes to processes: Guiding prm learning from orm for inference-time alignment. *arXiv preprint arXiv:2506.12446*.

Guofu Xie, Yunsheng Shi, Hongtao Tian, Ting Yao, and Xiao Zhang. 2025c. Capo: Towards enhancing llm reasoning through verifiable generative credit assignment. *arXiv preprint arXiv:2508.02298*.

Zhaopan Xu, Pengfei Zhou, Jiaxin Ai, Wangbo Zhao, Kai Wang, Xiaojiang Peng, Wenqi Shao, Hongxun Yao, and Kaipeng Zhang. 2025. Mpbench: A comprehensive multimodal reasoning benchmark for process errors identification. *arXiv preprint arXiv:2503.12505*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Minglai Yang, Ethan Huang, Liang Zhang, Mihai Surdeanu, William Wang, and Liangming Pan. 2025b. How is llm reasoning distracted by irrelevant context? an analysis using a controlled benchmark. *arXiv preprint arXiv:2505.18761*.

Yingxuan Yang, Huacan Chai, Yuanyi Song, Siyuan Qi, Muning Wen, Ning Li, Junwei Liao, Haoyi Hu, Jianghao Lin, Gaowei Chang, and 1 others. 2025c. A survey of ai agent protocols. *arXiv preprint arXiv:2504.16736*.

Chenlu Ye, Zhou Yu, Ziji Zhang, Hao Chen, Narayanan Sadagopan, Jing Huang, Tong Zhang, and Anurag Beniwal. 2025. Beyond correctness: Harmonizing process and outcome rewards through rl training. *arXiv preprint arXiv:2509.03403*.

Rui Yu, Shenghua Wan, Yucen Wang, Chen-Xiao Gao, Le Gan, Zongzhang Zhang, and De-Chuan Zhan. 2025. Reward models in deep reinforcement learning: A survey. *arXiv preprint arXiv:2506.15421*.

Weichen Yu, Ravi Mangal, Yinyi Luo, Kai Hu, Jingxuan He, Corina S Pasareanu, and Matt Fredrikson. 2026. Seccodeprm: A process reward model for code security. *arXiv preprint arXiv:2602.10418*.

Thomas Zeng, Shuibai Zhang, Shutong Wu, Christian Classen, Daewon Chae, Ethan Ewer, Minjae Lee, Heeju Kim, Wonjun Kang, Jackson Kunde, and 1 others. 2025. Versaprm: Multi-domain process reward model via synthetic reasoning data. *arXiv preprint arXiv:2502.06737*.

Danyang Zhang, Situo Zhang, Ziyue Yang, Zichen Zhu, Zihan Zhao, Ruisheng Cao, Lu Chen, and Kai Yu. 2025a. Progrm: Build better gui agents with progress rewards. *arXiv preprint arXiv:2505.18121*.

Hanning Zhang, Pengcheng Wang, Shizhe Diao, Yong Lin, Rui Pan, Hanze Dong, Dylan Zhang, Pavlo Molchanov, and Tong Zhang. 2024. Entropy-regularized process reward model. *arXiv preprint arXiv:2412.11006*.

Jianghangfan Zhang, Yibo Yan, Kening Zheng, Xin Zou, Song Dai, and Xuming Hu. 2025b. Gm-prm: A generative multimodal process reward model for multimodal mathematical reasoning. *arXiv preprint arXiv:2508.04088*.

Kaiyan Zhang, Jiayuan Zhang, Haoxin Li, Xuekai Zhu, Ermo Hua, Xingtai Lv, Ning Ding, Biqing Qi, and Bowen Zhou. 2025c. Openprm: Building open-domain process-based reward models with preference trees. In *The Thirteenth International Conference on Learning Representations*.

Kangning Zhang, Wenxiang Jiao, Kounianhua Du, Yuan Lu, Weiwen Liu, Weinan Zhang, and Yong Yu. 2025d. Looptool: Closing the data-training loop for robust llm tool calls. *Preprint*, arXiv:2511.09148.

Lingyin Zhang, Jun Gao, Xiaoxue Ren, and Ziqiang Cao. 2025e. The bidirectional process reward model. *arXiv preprint arXiv:2508.01682*.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2025f. Generative verifiers: Reward modeling as next-token prediction. *Preprint*, arXiv:2408.15240.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, Irwin King, Xue Liu, and Chen Ma. 2025g. A survey on test-time scaling in large language models: What, how, where, and how well? *arXiv preprint arXiv:2503.24235*. Version v3.

Ruiyi Zhang, Peijia Qin, Qi Cao, Eric Xue, and Pengtao Xie. 2026a. Funprm: Function-as-step process reward model with meta reward correction for code generation. *arXiv preprint arXiv:2601.22249*.

Shimao Zhang, Xiao Liu, Xin Zhang, Junxiao Liu, Zheheng Luo, Shujian Huang, and Yeyun Gong. 2025h. Process-based self-rewarding language models. *arXiv preprint arXiv:2503.03746*.

Wenlin Zhang, Xiangyang Li, Kuicai Dong, Yichao Wang, Pengyue Jia, Xiaopeng Li, Yingyi Zhang, Derong Xu, Zhaocheng Du, Huifeng Guo, and 1 others. 2025i. Process vs. outcome reward: Which is better for agentic rag reinforcement learning. *arXiv preprint arXiv:2505.14069*.

Yao Zhang, Shijie Tang, Zeyu Li, Zhen Han, and Volker Tresp. 2026b. Webarbiter: A principle-guided reasoning process reward model for web agents. *arXiv preprint arXiv:2601.21872*.

Yuxin Zhang, Meihao Fan, Ju Fan, Mingyang Yi, Yuyu Luo, Jian Tan, and Guoliang Li. 2025j. Reward-sql: Boosting text-to-sql via stepwise reasoning and process-supervised rewards. *arXiv preprint arXiv:2505.04671*.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025k. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*.

Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, and 1 others. 2025. Genprm: Scaling test-time compute of process reward models via generative reasoning. *arXiv preprint arXiv:2504.00891*.

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

Congmin Zheng, Xiaoyun Mo, Xinbei Ma, Qiqiang Lin, Yin Zhao, Jiachen Zhu, Xingyu Lou, Jun Wang, Zhaoxiang Wang, Weiwen Liu, and 1 others. 2026. Adaptive milestone reward for gui agents. *arXiv preprint arXiv:2602.11524*.

Congmin Zheng, Jiachen Zhu, Jianghao Lin, Xinyi Dai, Yong Yu, Weinan Zhang, and Mengyue Yang. 2025. Cold: Counterfactually-guided length debiasing for process reward models. *arXiv preprint arXiv:2507.15698*.

Jialun Zhong, Wei Shen, Yanzeng Li, Songyang Gao, Hua Lu, Yicheng Chen, Yang Zhang, Wei Zhou, Jinjie Gu, and Lei Zou. 2025. A comprehensive survey of reward models: Taxonomy, applications, challenges, and future. *arXiv preprint arXiv:2504.12328*.

Chenyu Zhou, Huacan Chai, Wenteng Chen, Zihan Guo, Rong Shan, Yuanyi Song, Tianyi Xu, Yingxuan Yang, Aofan Yu, Weiming Zhang, and 1 others. 2026. Externalization in llm agents: A unified review of memory, skills, protocols and harness engineering. *arXiv preprint arXiv:2604.08224*.

Chenyu Zhou, Tianyi Xu, Jianghao Lin, and Dongdong Ge. 2025a. Steporlm: A self-evolving framework with generative process supervision for operations research language models. *arXiv preprint arXiv:2509.22558*.

Yuanchen Zhou, Shuo Jiang, Jie Zhu, Junhui Li, Lifan Guo, Feng Chen, and Chi Zhang. 2025b. Fin-prm: A domain-specialized process reward model for financial reasoning in large language models. *arXiv preprint arXiv:2508.15202*.

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

A Paper Structure and Taxonomy Overview

Figure 3 illustrates the organizational structure and taxonomy adopted in this survey. At the top level, the survey is built around the full PRM loop: **Data Process** (Sec. 2), **PRM Training** (Sec. 3), **PRM Usage** (Sec. 4), and **Benchmark** (Sec. 6). Each component is further decomposed into finer categories to reflect the main research threads and representative works.

Data Process. We categorize data construction methods into three paradigms: *Human Annotation* (§2.1), which builds high-fidelity step-level supervision through expert labeling; *Automated Supervision* (§2.2), which scales data generation with verifiers, search, and synthetic signals; and *Semi-automated Approaches* (§2.3), which combine limited manual curation with automatic expansion to balance fidelity and scalability.

PRM Training. Modeling methods are grouped into four classes: *Discriminative PRMs* (§3.1), which directly score step correctness with pointwise or pairwise objectives; *Generative PRMs*(§3.2), which generate critique or verification chains before rating steps; *Implicit PRMs*(§ 3.3), which derive rewards without explicit labels via self-supervision or outcome transfer; and *Other Architectures*(§ 3.4), covering graph-based, retrieval-augmented, multilingual, and specialized structural designs.

PRM Usage. We summarize two primary usage paradigms: *Test-Time Scaling* (§4.1), where PRMs re-rank, verify, and adaptively guide reasoning during inference; and *RL for Policy Learning* (§4.2), where PRM signals serve as dense rewards for reinforcement learning to improve reasoning policies.

Benchmark. The bottom layer highlights major *benchmarks* (§6) for PRM evaluation, spanning

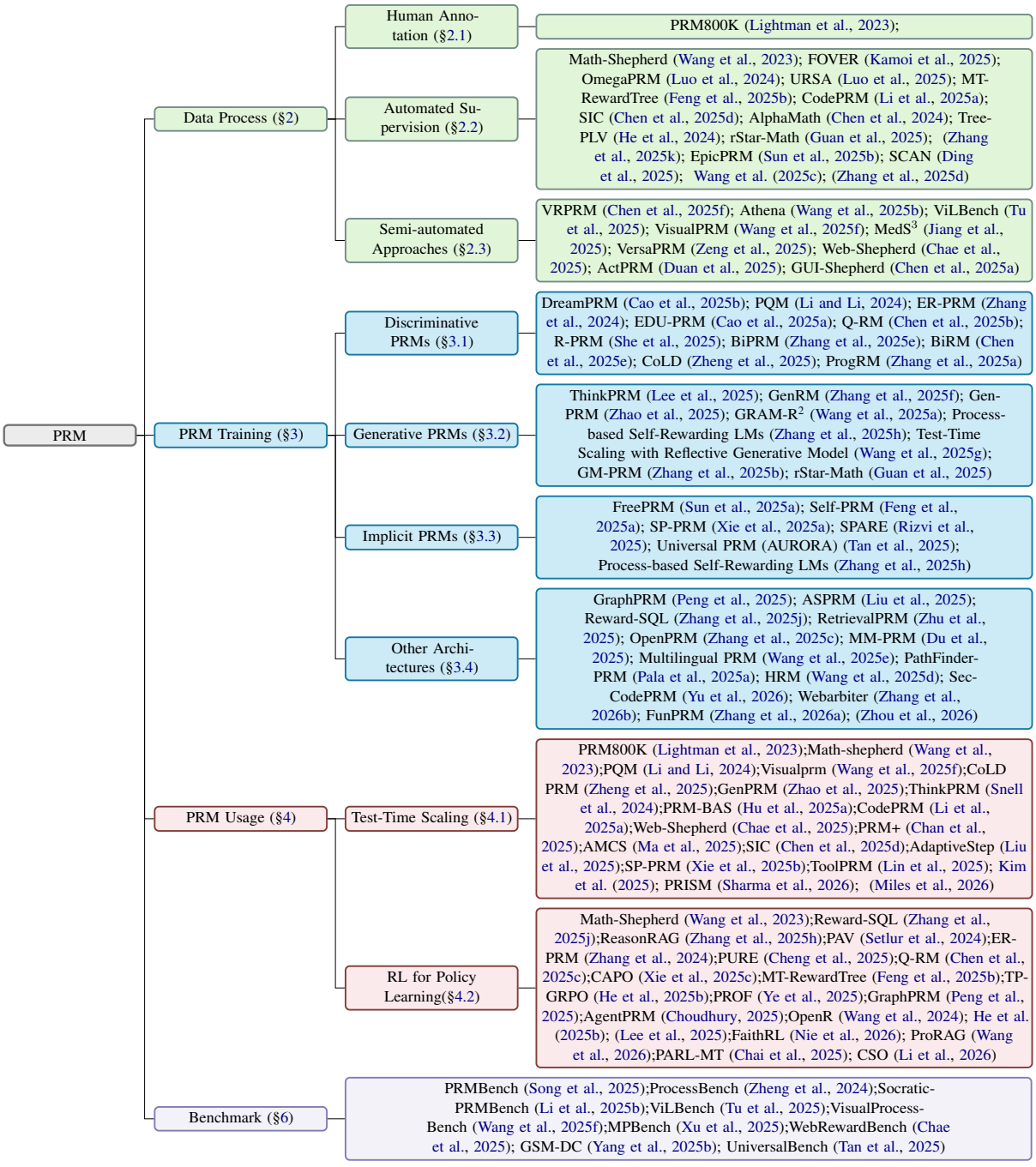

Figure 3: The overall structure of this paper.

mathematical reasoning, multimodal tasks, long-horizon web navigation, robustness testing, and cross-domain generalization.

Overall, this diagram provides a visual roadmap of the survey: from how process-level data is built, to the modeling strategies and deployment of PRMs, and finally to the resources enabling evaluation and comparison. It helps readers navigate the field and locate specific methods or datasets within our proposed taxonomy.

B Further Discussion

B.1 The Cognitive Scalability of Human Annotation

While current literature often cites the high cost of human annotation as a primary bottleneck, **a more fundamental issue is the limit of "cognitive scalability."** As reasoning tasks escalate in complexity (from elementary math to Olympiad-level problems or long-horizon agentic planning), verifying intermediate steps becomes exponentially

Table 1: Performance of various models on Processbench and PRMBench.

Benchmark	Processbench					PRMBench			
	GSM8K	MATH	OlympiadBench	OmniMATH	Average	Simplicity	Soundness	Sensitivity	Overall
PRMs									
Math-Shepherd-7B	47.9	29.5	24.8	23.8	31.5	47.1	45.7	60.7	47.0
Math-PSA-7B	62.4	41.9	31.5	25.2	40.3	51.3	51.8	64.9	52.3
Skywork-PRM-1.5B	59.0	48.0	19.3	19.2	36.4	54.2	64.9	70.7	61.1
Skywork-PRM-7B	70.8	53.6	22.9	21.0	42.1	**59.6**	68.5	73.3	65.1
Llemma-PRM800K-7B	48.4	43.1	28.5	33.4	38.4	51.4	50.9	66.0	52.0
RLHFlow-PRM-Mistral-8B	50.4	33.4	13.8	15.8	28.4	46.7	57.5	68.5	54.4
RLHFlow-PRM-Deepseek-8B	38.8	33.8	16.9	16.9	26.6	47.6	57.5	68.1	54.2
Qwen2.5-Math-7B-PRM800K	68.2	62.6	50.7	44.3	56.5	48.2	62.2	72.2	58.3
Qwen2.5-Math-PRM-7B	82.4	77.6	67.5	66.3	73.5	52.1	71.0	75.5	65.5
R-PRM-7B-SFT	77.2	71.6	59.6	52.3	65.2	58.7	66.4	75.7	64.9
R-PRM-7B-DPO	80.7	76.9	63.8	60.1	70.4	55.2	71.2	76.6	66.8
PathFinder-PRM-7B	77.9	75.3	65.0	59.7	69.5	58.9	70.8	76.9	67.7
ACTPRM	81.6	79.8	71.4	67.0	75.0	53.6	71.3	75.2	65.5
ACTPRM-X	**82.7**	**82.0**	**72.0**	**67.3**	**76.0**	54.5	**72.7**	75.6	66.7
RetrievalPRM-7B	74.6	71.1	60.2	57.3	65.8	55.3	75.0	**78.2**	**68.9**
ReasonEval-7B	41.0	48.9	36.7	37.4	41.0	55.5	63.9	71.0	60.0
Critic Models									
GPT-4o	79.2	63.6	51.4	53.5	61.9	59.7	70.9	**75.8**	66.8
o1-mini	**93.2**	**88.9**	**87.2**	**82.4**	**87.9**	**64.6**	**72.1**	75.5	**68.8**
QwQ-32B-Preview	88.0	78.7	57.8	61.3	71.5	56.4	68.2	73.5	63.6

harder than generating the final answer. In datasets like PRM800K (Lightman et al., 2023), human annotators are assumed to be ground-truth oracles. However, this assumption fractures when the Policy Model begins to surpass human reasoning capabilities. This manifests as a "Superalignment" problem: average human annotators struggle to distinguish between subtle logical hallucinations and correct, novel derivation steps. **Consequently, relying solely on human supervision risks imposing a "human ceiling" on model performance, where the Reward Model penalizes valid but complex reasoning simply because it exceeds the annotator's cognitive load or domain expertise.**

Resolving this tension requires moving beyond unassisted human labeling toward "Scalable Oversight" paradigms. Future research directions must likely pivot from direct annotation to AI-assisted verification workflows, where humans act not as raw labelers but as "managers" of automated verification tools (e.g., using code interpreters or formal theorem provers like Lean/Isabelle to validate intermediate logic objectively). This shifts the human role from verifying correctness (which is hard) to verifying intent and alignment (which is more intuitive). **By grounding rewards in objective execution feedback (compilers, formal verifiers) rather than subjective human preference, the field can decouple the scaling of reasoning capability from the limitations of human cognitive load.**

B.2 Echo Chambers and Goodhart's Law in Automated Supervision

To bypass human bottlenecks, the field has pivoted toward automated supervision (Wang et al., 2023; Luo et al., 2024), yet this introduces a perilous dynamic between the Reward Model and the Policy Model. When a Reward Model is trained on synthetic data generated by a similar Policy Model (or verified by a model with similar pre-training), they share the same "knowledge blind spots." **This creates an "Echo Chamber Effect" where plausible hallucinations are reinforced rather than corrected because both models share the same underlying misconceptions.** Furthermore, this setup is highly susceptible to Goodhart's Law. As the Policy Model optimizes against a fixed automated Reward Model, **it learns to exploit the Reward Model's biases**, such as favoring longer chains (Zheng et al., 2025), specific formatting, or confident phrasing, rather than improving genuine logic. This "reward hacking" results in high rewards for vacuous reasoning, a phenomenon that is difficult to detect without external, diverse verification sources.

Addressing these systemic flaws requires a shift from focusing on internal consistency to embracing external grounding and adversarial robustness. A promising direction is to introduce heterogeneous supervision, where PRMs are guided not by a single model but **by an "adversarial council"**

of diverse models with different architectures, scales, or training corpora that are encouraged to search for weaknesses rather than reinforce agreement. In addition, future research should explore dynamic reward landscapes instead of relying on fixed reward models. Under iterative or adversarial training regimes, each time a policy model discovers a new exploit, the reward model can be updated or red-teamed to detect and penalize that behavior. Such **an evolving interplay creates a curriculum that continually challenges the policy model and steers it toward genuine robustness rather than superficial consistency or metric gaming**.

B.3 The Tension of Granularity: Defining a "Step"

A critical, yet frequently overlooked tension in PRM construction is the definition of the fundamental unit of analysis: the "reasoning step." Current approaches predominantly rely on rigid, heuristic-based segmentation, such as splitting logic by newline characters or specific delimiters. **This "Rigid Segmentation" imposes an artificial structure that often conflicts with the natural, semantic flow of reasoning.** This misalignment is particularly acute in domains with complex structural dependencies, such as code generation. Unlike mathematical derivations where line-by-line transitions often correlate with logical progress, programming logic is inherently nested and interdependent. Consequently, defining the "optimal truncation position" for a PRM becomes a non-trivial challenge: evaluating too frequently (e.g., every line) introduces noise and breaks syntactic context, while evaluating too sparsely (e.g., per function) dilutes the dense supervision signal that PRMs promise. **The field currently lacks a principled method to determine where a "logical thought" begins and ends**, leading to situations where PRMs penalize valid partial steps simply because the segmentation cut occurred at a syntactically awkward moment, obscuring the true quality of the underlying logic.

To resolve this tension, **we argue that the field should move beyond rigid rule-based segmentation toward dynamic and learnable granularity**. One direction is semantic segmentation, where models learn to identify their own "Atomic Reasoning Units" using dedicated signals such as learnable step-boundary tokens rather than relying on manually imposed formatting. Building on this idea, future work may explore hierarchical supervision through **Multi-Scale PRMs that evaluate reasoning at multiple levels**, offering micro-rewards for syntactic fidelity at the token or line scale and macro-rewards for logical coherence at the block scale. Ultimately, research may even move past discrete segmentation altogether by adopting continuous, flow-based evaluation, in which a separate critic monitors the actor model's hidden states and intervenes only when it detects deviations in the reasoning trajectory.

B.4 The Proxy-Reward Gap: Proxy Metrics vs. Actual Utility

There is a growing disconnect between how PRMs are evaluated and how they are utilized. Most benchmarks (Zheng et al., 2024) evaluate PRMs using classification or ranking accuracy on static datasets. **However, "Good Classifiers do not always make Good Navigators."** A Reward Model that achieves high accuracy on a static test set may fail catastrophically during dynamic inference (e.g., Tree Search or RL). **This is primarily a calibration and Out-of-Distribution (OOD) robustness issue.** In active search (Guan et al., 2025), the Policy Model explores diverse, often erroneous paths that differ significantly from the PRM's training distribution. A PRM optimized for static accuracy might lack the calibration necessary to effectively prune these branches, leading to a divergence where improved benchmark metrics do not translate to downstream task success.

To bridge this gap, we argue that **the community must fundamentally reconsider how PRM success is defined and evaluated**. Future work should emphasize online and active evaluation through on-policy testing, where the PRM is evaluated on the trajectory distribution actually produced by the target policy model during inference. This perspective naturally motivates online iterative training, in which the reward model is continuously updated to differentiate the most challenging errors currently generated by the policy rather than relying on outdated datasets. We suggest placing greater emphasis on calibration-first objectives, prioritizing calibration error and out-of-distribution robustness as primary metrics. **A valuable PRM is not one that is uniformly confident, but one that can recognize and signal its own uncertainty**, enabling fallback mechanisms such as human supervision or tool invocation rather than confidently steering the reasoning process in the wrong direction.