Fin-PRM: A Domain-Specialized Process Reward Model for Financial Reasoning in Large Language Models

Jie Zhu[1,2], **Yuanchen Zhou**[2], **Shuo Jiang**[2,3], **Junhui Li**[1*], **Lifan Guo**[2*], **Feng Chen**[2], **Chi Zhang**[2]

[1]School of Computer Science and Technology, Soochow University
[2]Qwen DianJin Team, Alibaba Cloud Computing
[3]Osaka University
zhujie951121@gmail.com, lijunhui@suda.edu.cn,
{jiangshuo.jiang, lifan.lg, betterman.chenf, edward.zhang}@alibaba-inc.com

Abstract

Process Reward Models (PRMs) supervise intermediate reasoning steps in large language models (LLMs), but existing PRMs are mainly trained on general-domain data and struggle with the structured, symbolic, and fact-sensitive nature of financial reasoning. Financial tasks require not only correct final answers but also verifiable intermediate steps grounded in domain knowledge. In this paper, we propose Fin-PRM, a domain-specialized, trajectory-aware PRM for financial reasoning that jointly models step-level correctness and trajectory-level coherence, producing binary supervision signals for both local and global reasoning quality. To support reliable supervision, we construct a high-quality financial reasoning dataset of 3K trajectories, where step- and trajectory-level labels are automatically derived from multi-source reward signals, including Monte Carlo rollouts, LLM-based evaluation, and explicit financial knowledge verification. Fin-PRM defines a unified ranking score that integrates step- and trajectory-level rewards, enabling consistent use across multiple settings. We evaluate Fin-PRM in three scenarios: (1) offline trajectory selection for supervised fine-tuning, (2) reward-guided Best-of-N inference for test-time scaling, and (3) process-aware reward shaping for reinforcement learning. Experiments on financial reasoning benchmarks, including CFLUE and FinQA, show that Fin-PRM consistently outperforms general-purpose PRMs and strong baselines. Our project resources will be available at https://github.com/aliyun/qwen-dianjin.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in complex reasoning tasks, motivating their increasing use in specialized domains such as finance [Yang *et al.*, 2023; Zhu *et al.*, 2025]. However, financial reasoning tasks, including financial statement analysis, investment strategy formulation, and regulatory compliance assessment, require precision, factual correctness, and logical coherence that often exceed the capabilities of general-purpose models. In finance, even minor errors in intermediate reasoning steps can invalidate entire solutions, highlighting the need for supervision mechanisms that evaluate not just final answers but the reasoning process itself.

Process reward models (PRMs) have emerged as a promising approach for supervising intermediate reasoning. PRMs assign rewards to individual steps or full reasoning trajectories, helping models select the best responses among multiple candidates and providing scalar signals for reinforcement learning [Lightman *et al.*, 2024; Zhang *et al.*, 2025b; Setlur *et al.*, 2025; Khalifa *et al.*, 2025; Liu *et al.*, 2025; Zou *et al.*, 2025; Cui *et al.*, 2025]. Prior work has shown PRMs to be effective in general domains, but their domain-agnostic nature limits performance in financial reasoning, where structured, symbolic, and verifiable constraints must be respected. A central challenge is the quality and interpretability of reward signals. While recent work has leveraged LLMs as automated judges [Gu *et al.*, 2025], such signals are often opaque and insufficiently grounded in domain knowledge. In contrast, many financial reasoning steps, such as numerical computations, rule-based deductions, and fact lookups, can be verified. To leverage this structure, we introduce knowledge-aware verification signals that aggregate reward labels in a way that is both interpretable and domain-aware.

Within this framework, we introduce Fin-PRM, a domain-specialized, trajectory-aware PRM for financial reasoning. We train Fin-PRM on a curated dataset of 3,000 step-by-step reasoning trajectories from CFLUE [Zhu *et al.*, 2024], a knowledge-based Chinese financial benchmark, generated with a strong reasoning model [Guo *et al.*, 2025] and annotated with verifiable reward signals. Fin-PRM provides reward supervision at both step and trajectory levels, supporting three applications: offline trajectory selection for supervised fine-tuning (SFT), reward-guided Best-of-N inference for test-time scaling, and process-level reward shaping for reinforcement learning.

*Corresponding Author.

In summary, our primary contributions include:

- **A High-Quality Financial Reasoning Dataset**: We construct and curate 3,000 step-by-step reasoning traces with verifiable reward annotations, providing a high-quality dataset for financial reasoning supervision.
- **A Novel Dual-Level Training Framework**: We propose a training paradigm that integrates step-level and trajectory-level rewards, enabling PRMs to learn from multi-dimensional feedback while preserving interpretability.
- **Comprehensive Experimental Validation**: We demonstrate the effectiveness of Fin-PRM across three scenarios, offline trajectory selection for supervised fine-tuning, reward-guided Best-of-N inference at test time, and process-level reward shaping for reinforcement learning, showing significant improvements in downstream financial reasoning performance.

2 Related Work

2.1 Process Reward Models

Process Reward Models (PRMs) have emerged as a crucial framework for providing step-level supervision and interpretable reward signals in complex reasoning tasks. State-of-the-art PRMs, exemplified by MathShepherd [Wang *et al.*, 2024], Skywork-PRM [He *et al.*, 2024], and Qwen2.5-Math-PRM [Zhang *et al.*, 2025b], employ human-annotated supervision with synthetic reward generation to deliver evaluation capabilities across diverse reasoning domains including mathematical problem solving, scientific analysis, and programming. Recent exploratory works such as ReasonFlux-PRM [Zou *et al.*, 2025] combines both step-level and template-guided trace-level reward signals, Open-PRM [Zhang *et al.*, 2025a] leverages authoritative ORMs to reverse-engineer process-level supervision signals. In application, PRMs successfully integrated into Best-of-N sampling [Liu *et al.*, 2025], offline data selection [Xie *et al.*, 2023], and online reinforcement learning for model optimization [Bai *et al.*, 2022]. However, effective PRM evaluation should derive its reasoning assessment capabilities from concrete thinking trajectories rather than merely final solution correctness, and real-world vertical domain applications of PRMs impose critical requirements for deep domain knowledge mastery. Guided by these thoughts, we design a domain-socialized framework that integrates trajectory-aware evaluation with expert knowledge validation, enabling more reliable process-level assessment for finical domain.

2.2 Data Synthesis for Reasoning Tasks

High-quality data has proven fundamental to developing effective reasoning models [Gunasekar *et al.*, 2023]. Early approaches focused on expanded existing datasets through rule-based transformations and template-driven generation [Wei and Zou, 2019]. These methods improving data coverage, but lacked the sophistication required for complex reasoning task. LLMs has enabled more advanced synthesis paradigms with distillation-based approaches leveraging powerful teacher models to generate high-quality reasoning traces for training efficient student models [Mukherjee *et al.*, 2023]. Notable contributions include WizardMath [Luo *et al.*, 2025] synthesizes reasoning data through instruction evolution, MetaMath [Yu *et al.*, 2024] generates diverse problems with step-by-step solutions, and OpenThoughts [Guha *et al.*, 2025] establishes systematic methodological foundations for reasoning data synthesis through comprehensive ablation studies, providing empirical evidence for data composition principles. Considered that domain-specific applications demand specialized knowledge while maintaining reasoning quality standards, we adopt CFLUE [Zhu *et al.*, 2024] as data source and Deepseek-R1 as teacher model to obtain reasoning trace. Recent advanced approaches employ LLM-as-a-judge for automated reward labeling, which always challenged by inherent limitations including black-box evaluation processes and insufficient reproducibility [Zheng *et al.*, 2023]. Building upon these observations, we enhance our reward annotation methodology by integrating expert-based knowledge verification mechanisms, significantly improves the observability and trustworthiness of reward signals, thereby ensuring more reliable evaluation capabilities for complex financial reasoning tasks.

3 Financial Reasoning Dataset Construction

The effectiveness of a process reward model (PRM) hinges on the quality of its training data [Ye *et al.*, 2025]. In financial reasoning, supervision must capture expert-level intermediate steps since small reasoning errors can invalidate an entire solution [Lightman *et al.*, 2024]. Figure 1 (upper) illustrates the Fin-PRM data construction pipeline. We next describe how we build a high-quality chain-of-thought dataset to support verifiable and interpretable reward modeling.

3.1 Synthesizing Reasoning Trajectories

We adopt CFLUE [Zhu *et al.*, 2024], a knowledge-based Chinese financial benchmark, as our primary data source. Its Knowledge Assessment subset consists of multiple-choice questions spanning a diverse range of financial reasoning tasks and is accompanied by expert-written analyses. These analyses provide not only reliable ground-truth answers but also rich domain knowledge, making the dataset well suited for supervising and evaluating financial reasoning processes.

Following the systematic data synthesis framework of OpenThoughts [Guha *et al.*, 2025], we employ a strong reasoning model, Deepseek-R1 [Guo *et al.*, 2025], to generate structured reasoning trajectories. Given an input question x from the CFLUE training split, the model produces a pair (s, a), where:

- $s = (s_1, s_2, \ldots, s_T)$ denotes a *reasoning trace*, consisting of a sequence of intermediate reasoning steps.
- a denotes the final *answer* derived from the reasoning trace.

Each triplet (x, s, a) forms a candidate reasoning trajectory. We first filter candidate triplets based on the length of the reasoning trace s to remove overly short or degenerate trajectories. We then use Qwen3-8B to perform reasoning-on and reasoning-off evaluations for each candidate sample. A triplet is retained only if the model answers correctly with

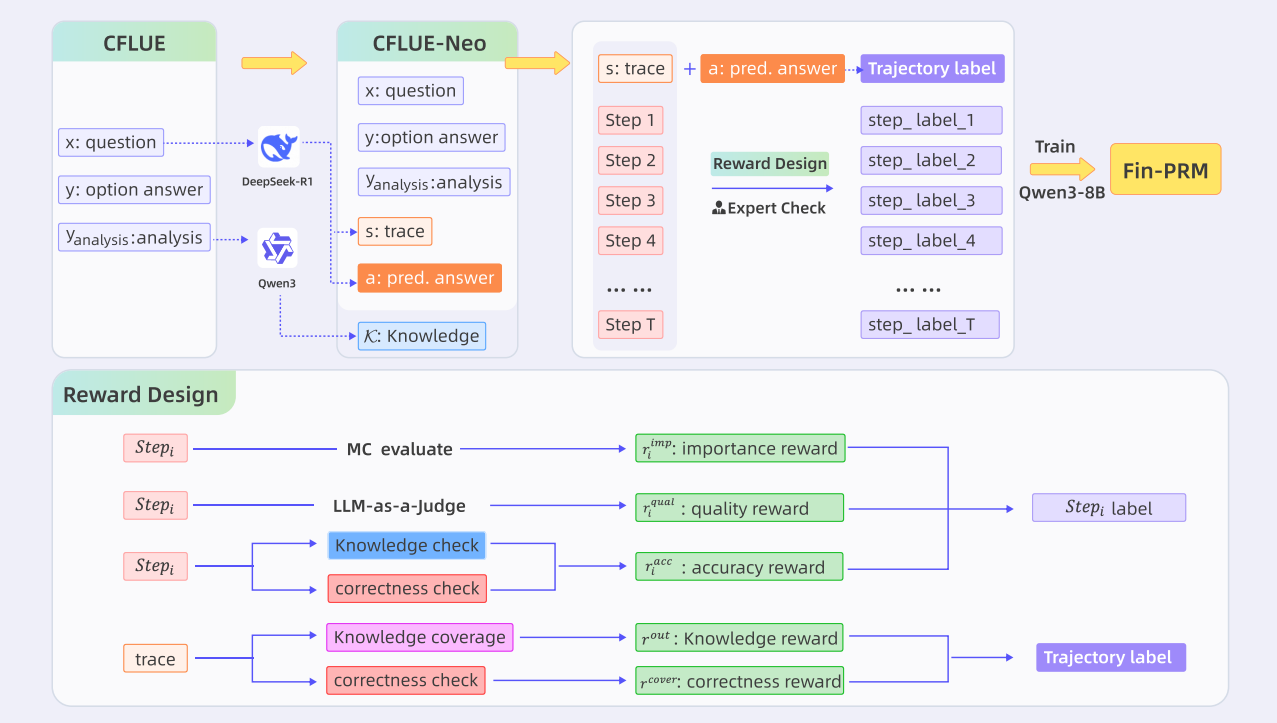

Figure 1: Illustration of the data construction process (upper) and the step-level and trajectory-level labeling procedures used for Fin-PRM training (lower).

reasoning enabled but fails without reasoning. This filtering ensures that retained samples intrinsically depend on explicit reasoning rather than surface-level pattern matching.

After this preprocessing, similar to [Zou *et al.*, 2025], we obtain a curated dataset, CFLUE-Neo, consisting of 3,000 reasoning triplets. These triplets are then annotated with step-level and trajectory-level reward signals to train Fin-PRM.

3.2 Financial Knowledge Extraction

Financial reasoning is intensely knowledge-driven. To reduce reward-hacking and ensure interpretable supervision [Khalifa *et al.*, 2025], we construct an explicit financial knowledge base $\mathcal{K}$ extracted from the expert analyses provided in CFLUE.

We employ Qwen3-235b-a22b to extract key financial concepts and their definitions from expert-written explanations. For example, from an analysis discussing company valuation, the model may extract:

- **Term:** Price-to-Book Ratio
- **Explanation:** A financial ratio used to compare a company's current market price to its book value.

The resulting knowledge base $\mathcal{K}$ serves as an external, trusted reference for subsequent knowledge verification during reward annotation, enabling reward signals that are factually grounded rather than purely judgment-based.

Based on CFLUE, our final dataset $\mathcal{D}$ consists of tuples $(x, s, a, y, y_{\text{analysis}})$, where y denotes the gold-standard answer and y_{analysis} is the corresponding expert analysis. Since the teacher model may occasionally produce incorrect final answers, we treat a as a silver-standard solution. In contrast, expert-provided answers, analyses, and the extracted knowledge base $\mathcal{K}$ are used as authoritative sources for verification and reward supervision.

4 Fin-PRM: Domain-Specialized Process Reward Model

As illustrated in the lower part of Figure 1, we define binary supervision signals at two granularities: step-level labels that capture local reasoning correctness (Section 4.1) and trajectory-level labels that reflect global reasoning validity (Section 4.2). Based on these labels, we joint train Fin-PRM to predict step-wise and trajectory-wise rewards within a unified framework (Section 4.3).

4.1 Step-Level Labeling

Given an input question x and a reasoning trajectory $s = (s_1, s_2, \ldots, s_T)$, we assign a binary label to each reasoning step s_t, denoted as $L_t^{\text{step}} \in \{0, 1\}$. A positive label indicates that the step constitutes a valid and useful intermediate reasoning action, while a negative label denotes an incorrect, misleading, or unsupported step.

To determine these labels, we first compute continuous *step-level reward scores* from multiple complementary perspectives, and then aggregate and binarize them to obtain final step-level labels.

Step-Level Reward Scores

To capture the multifaceted nature of high-quality financial reasoning, we decompose the step-level reward for s_t into

three components: an importance score r_t^{imp}, a qualitative score r_t^{qual}, and an accuracy score r_t^{acc}.

Importance Score r_t^{imp}. The importance score measures the potential of a reasoning step to lead toward a correct solution. For each step s_t, we prompt Qwen2.5-7B-Math to generate N Monte Carlo rollouts (with $N = 8$ in our experiments), continuing the reasoning process from s_t until a final answer is produced. The importance score is defined as the proportion of rollouts that yield a correct final answer:

$$r_t^{\text{imp}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{I}\big(\xi(\mathrm{R}\mu, i(s_t \mid x, s{<}\, t, y))\big), \tag{1}$$

where $\mathrm{R}_{\mu,i}$ denotes the i-th rollout, $\xi(\cdot)$ is an answer-checking function, and $\mathbf{I}(\cdot)$ is the indicator function which returns 1 if the final answer of the rollout is correct and 0 otherwise. This score provides a soft estimate of whether the current step lies on a promising reasoning path.

Qualitative Score r_t^{qual}. The qualitative score evaluates the semantic and logical quality of a reasoning step. We employ an LLM (Qwen3-235B-A22B) as an automated judge to assess each step s_t with respect to semantic coherence, logical soundness, and alignment with the problem objective. The model produces a scalar score in $[0, 1]$:

$$r_t^{\text{qual}} = R_\theta(s_t \mid x, s_{<t}, a). \tag{2}$$

Accuracy Score r_t^{acc}. The accuracy score provides supervision in verifiable ground truth and domain knowledge. It consists of two complementary components:

Procedural Correctness ($r_t^{\text{step\_corr}}$). This component assesses whether s_t represents a logically valid and relevant step toward the gold-standard answer y. We prompt an LLM (Qwen3-235B-A22B) to produce a binary judgment indicating procedural correctness (1 for correct, 0 for incorrect).

Factual Accuracy ($r_t^{\text{know\_acc}}$). This component evaluates whether factual claims and financial concepts in s_t are consistent with the extracted knowledge base $\mathcal{K}$ and expert analysis y_{analysis}. Similarly, we prompt Qwen3-235B-A22B to produce a binary judgment indicating procedural correctness (1 for correct, 0 for incorrect).

The final accuracy score is computed as:

$$r_t^{\text{acc}} = 0.5\big(r_t^{\text{step\_corr}}(s_t, y) + \omega_k \cdot r_t^{\text{know\_acc}}(s_t, \mathcal{K}_x)\big), \tag{3}$$

where ω_k controls the relative importance of factual grounding. We set $\omega_k = 1.0$ in all experiments.

Step-level Label Construction

To derive a single supervisory signal for each step, we aggregate the three reward scores using a dynamic weighting scheme based on the softmax function:

$$r_t^{\text{step}} = \sum_{k \in \text{imp,qual,acc}} \text{softmax}\big(r_t^{\text{imp}}, r_t^{\text{qual}}, r_t^{\text{acc}}\big)_k \cdot r_t^k. \tag{4}$$

This aggregation emphasizes the most informative signal among the three scores, making the supervision more robust than a fixed-weight average.

Finally, the aggregated score r_t^{step} is binarized using a threshold of 0.5 to obtain the step-level label:

$$L_t^{\text{step}} = \mathbf{I}\left(r_t^{\text{step}} > 0.5\right). \tag{5}$$

4.2 Trajectory-Level Labeling

While step-level supervision encourages local correctness, a reasoning trajectory composed of individually plausible steps may still lead to an incorrect final answer. Moreover, PRMs trained solely on step-level signals are vulnerable to reward hacking, where models optimize intermediate rewards without producing valid solutions. To address these issues, we introduce trajectory-level supervision that evaluates global correctness and knowledge completeness.

Trajectory-level Reward Modeling

We define two complementary reward components for a reasoning trajectory (s, a): an outcome-based correctness score r^{out} and a knowledge coverage score r^{cover}.

Outcome Correctness Score r^{out}. The outcome correctness score assesses whether the final answer a is correct. Since tasks in our dataset typically require selecting a discrete option (e.g., *A*, *B*, *ACD*), we directly compare the model's predicted answer a with the gold-standard answer y. This yields a strict binary signal: $r^{\text{out}} \in \{0, 1\}$.

Knowledge Coverage Score r^{cover}. Beyond correctness, high-quality financial reasoning should adequately leverage the relevant domain knowledge. The knowledge coverage score measures how thoroughly the reasoning trace and final answer utilize the required financial concepts. Formally, we define:

$$r^{\text{cover}} = \frac{|\phi_{\text{ext}}(s \oplus a) \cap \mathcal{K}_x|}{|\mathcal{K}_x|}, \tag{6}$$

where $\mathcal{K}_x \subseteq \mathcal{K}$ denotes the subset of financial knowledge deemed relevant to the input question x, and $\phi_{\text{ext}}(\cdot)$ extracts financial concepts mentioned in the generated reasoning and answer. The operator $\oplus$ denotes concatenation. This score encourages comprehensive reasoning grounded in the necessary domain knowledge, even when the final answer is correct.

Trajectory-level Label Construction

We combine the two trajectory-level reward components into a single scalar score:

$$r^{\text{traj}}(s, a) = r^{\text{out}}(a) + \eta \cdot r^{\text{cover}}(s, a), \tag{7}$$

where η controls the relative contribution of knowledge coverage. In our experiments, we set $\eta = 1.5$, ensuring that trajectories with correct answers but insufficient knowledge grounding are penalized.

The final trajectory-level label $L^{\text{traj}} \in \{0, 1\}$ is obtained by thresholding r^{traj} at 1.25:

$$L^{\text{traj}} = \mathbf{I}\left(r^{\text{traj}} > 1.25\right). \tag{8}$$

4.3 Joint Training of Fin-PRM

Given an input question x, a reasoning trace $s = (s_1, \ldots, s_T)$, and a final answer a, Fin-PRM is parameterized as a reward model R_ϕ that predicts binary reward signals at both the step and trajectory levels. In implementation, we concatenate the input question x, the reasoning steps s, and the final answer a into a single sequence. A special placeholder token is inserted after each reasoning step s_t and after

the final answer a. The concatenated sequence is fed into a pretrained LLM (Qwen3-8B), and the hidden states corresponding to these placeholder tokens are extracted and passed to lightweight classification heads to produce step-level and trajectory-level predictions.

Step-Level Prediction. At the step level, Fin-PRM estimates the correctness and utility of an individual reasoning step s_t, conditioned on the full context:

$$R_\phi^{\text{step}}(s_t \mid x, s_{<t}), \tag{9}$$

where $s_{<t}$ denotes the preceding reasoning history. This prediction reflects properties such as logical validity, numerical correctness, and relevance to the final objective.

Trajectory-Level Prediction. At the trajectory level, Fin-PRM evaluates the overall coherence, strategic soundness, and knowledge completeness of the entire reasoning process:

$$R_\phi^{\text{traj}}(s \mid x, a). \tag{10}$$

This score captures whether the reasoning trajectory follows an appropriate global strategy for solving the financial task.

Training Objective. We jointly train Fin-PRM using binary cross-entropy (BCE) loss for both step-level and trajectory-level supervision. The overall training objective is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{step}} + \lambda \cdot \mathcal{L}_{\text{traj}}, \tag{11}$$

where λ balances the contributions of step-level and trajectory-level losses.

The **step-level loss** is computed as the average BCE loss over all steps in a reasoning trace:

$$\mathcal{L}_{\text{step}} = \frac{1}{T}\sum_{t=1}^{T} \mathcal{L}_{\text{BCE}}\big(R_\phi^{\text{step}}(s_t \mid x, s_{<t}), L_t^{\text{step}}\big), \tag{12}$$

where $L_t^{\text{step}} \in \{0, 1\}$ denotes the ground-truth step-level label defined in Section 4.1.

The **trajectory-level loss** compares the predicted trajectory score with the corresponding trajectory-level label:

$$\mathcal{L}_{\text{traj}} = \mathcal{L}_{\text{BCE}}\big(R_\phi^{\text{traj}}(s \mid x, a), L^{\text{traj}}\big), \tag{13}$$

where $L_{\text{traj}} \in \{0, 1\}$ is constructed as described in Section 4.2.

The binary cross-entropy loss is defined as:

$$\mathcal{L}_{\text{BCE}}(z, L) = -\big[L \log \sigma(z) + (1 - L)\log(1 - \sigma(z))\big], \tag{14}$$

where $\sigma(\cdot)$ denotes the sigmoid function that maps model logits to probabilities.

5 Applications of Fin-PRM

We instantiate Fin-PRM on top of Qwen3-8B and evaluate its effectiveness across three representative application settings that reflect common usages of process reward models. In each setting, Fin-PRM is compared against relevant baselines to assess its impact on downstream financial reasoning performance.

- **Offline Trajectory Selection for Supervised Fine-Tuning.** Fin-PRM is used as an offline trajectory selector to filter and curate high-quality reasoning traces, enabling more efficient and effective supervised fine-tuning (SFT).

Data Selection	Accuracy (%)
None	45.3
Random Selection	43.8
Math-PRM-7B	56.5
Math-PRM-72B	57.1
Fin-PRM (Ours)	**58.2**

Table 1: Accuracy of Qwen2.5-7B-Instruct on the CFLUE test set using different data selection strategies for supervised fine-tuning. Best result is shown in **bold**.

- **Reward-Guided Best-of-N Inference.** Fin-PRM is applied at inference time to rank and select the best response from multiple candidates in a Best-of-N (BoN) decoding setting.
- **Process-Level Reward Shaping for Reinforcement Learning.** Fin-PRM serves as a dense, process-level reward function to guide policy optimization through reinforcement learning.

5.1 Offline Trajectory Selection for Supervised Fine-Tuning

PRMs can be used as offline filters to identify high-quality reasoning trajectories from large pools of synthetic data. In the financial domain, where incorrect intermediate steps can severely degrade learning, effective data selection is especially critical. In this experiment, we evaluate whether Fin-PRM can better curate supervised fine-tuning (SFT) data than general-purpose PRMs.

Settings. We first use Qwen3-8B to generate multiple distinct reasoning trajectories for each question in the CFLUE training set,[1] enabling diversity in intermediate reasoning paths. Each trajectory (x, s, a) is then scored by Fin-PRM using a combined step- and trajectory-level reward:

$$\hat{R} = \frac{1}{T}\sum_{t=1}^{T} \hat{R}_{\text{step}}(s_t \mid x, s_{<t}, a) + \zeta \cdot \hat{R}_{\text{traj}}(s, a \mid x), \tag{15}$$

where $\hat{R}_{\text{step}}$ and $\hat{R}_{\text{traj}}$ are the predicted step-level and trajectory-level scores produced by Fin-PRM. The hyperparameter ζ controls the trade-off between fine-grained step correctness and overall reasoning coherence; we set $\zeta = 1.0$ in all experiments.

Based on $\hat{R}$, we select the top 1000 highest-scoring trajectories as the SFT dataset and fine-tune Qwen2.5-7B-Instruct on this curated set. We compare Fin-PRM-based selection against several baselines, including random selection and selection using Math-PRM-7B and Math-PRM-72B.

Results. Table 1 reports the performance on the CFLUE test set. Random data selection degrades performance from 45.3% to 43.8%, demonstrating the risk of training on noisy or low-quality reasoning traces. In contrast, all PRM-based

[1]To avoid coupling reward modeling with the trajectory generator, and to test generalization beyond the teacher model used for PRM training, we do not use DeepSeek-R1 at this stage.

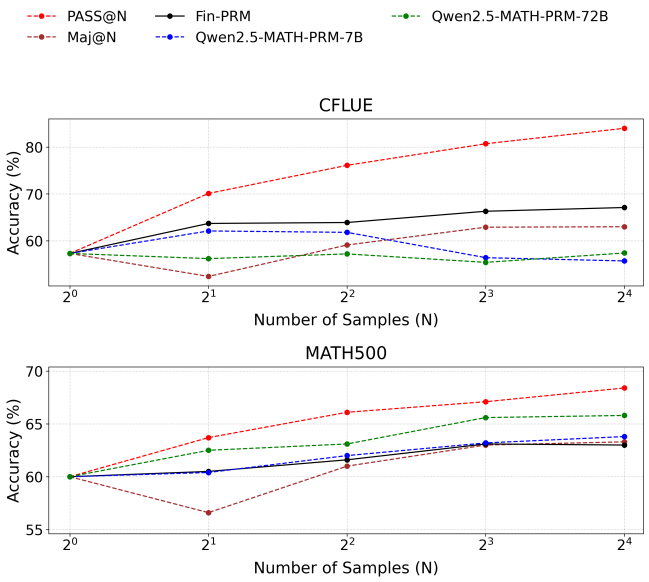

Figure 2: Best-of-N selection accuracy on the CFLUE (upper) and Math500 (lower) datasets for different values of N.

selection strategies significantly outperform the base model, confirming the effectiveness of reward-guided data curation.

Among all methods, Fin-PRM achieves the best performance, reaching an accuracy of 58.2%. This corresponds to a 12.9% improvement over the base model and surpasses strong general-domain PRMs. These results highlight the importance of domain-specialized process reward modeling for selecting high-quality financial reasoning data.

5.2 Reward-Guided Best-of-N Inference

A second important application of Fin-PRM is reward-guided test-time scaling via Best-of-N (BoN) selection. In this setting, a policy model generates multiple candidate reasoning trajectories, and a reward model is used to select the highest-quality response. Compared with simple heuristics such as majority voting, BoN selection enables more fine-grained evaluation of reasoning quality by explicitly scoring both step-level correctness and overall trajectory quality.

Settings. We evaluate BoN selection using the ranking score $\hat{R}$ defined in Eq. 15. Qwen2.5-7B-Instruct is used as the generator model to sample N candidate reasoning trajectories per question. We consider $N \in \{2, 4, 8, 16\}$ and apply Fin-PRM to select the highest-scoring candidate. As baselines, we include majority voting and a strong general-domain PRMs, Qwen2.5-Math-PRM-7B and Qwen2.5-Math-PRM-72B.

We first evaluate in-domain performance on a 1,000-sample subset of the CFLUE test set, followed by an out-of-domain evaluation on the Math500 benchmark to assess generalization.

Results on CFLUE. Figure 2 upper presents the BoN results on the CFLUE dataset. Fin-PRM consistently achieves larger accuracy gains as N increases, substantially outperforming majority voting across all settings. At $N = 16$, Fin-PRM exceeds majority voting by more than 5.1%, demonstrating the benefit of reward-guided selection over purely frequency-based aggregation.

In contrast, Qwen2.5-Math-PRM-7B shows diminishing returns as N grows and eventually underperforms majority voting. This behavior highlights the limitations of general-domain mathematical reward models when applied to financial reasoning, and underscores the importance of domain-specific trajectory evaluation.

Results on Math500. To evaluate robustness and generalization, we conduct the same BoN experiments on the Math500 benchmark. Results are shown in the below of Figure 2. Fin-PRM demonstrates stable and competitive performance, outperforming majority voting for small and moderate values of N (e.g., $N \leq 8$), while remaining comparable at larger N. Its performance closely tracks that of Qwen2.5-Math-PRM-7B at the same model scale, demonstrating that, despite being specialized for financial reasoning, Fin-PRM retains a solid generalization ability and does not overfit narrowly to the financial domain.

5.3 Process-Level Reward Shaping for Reinforcement Learning

Fin-PRM can also be used as a reward function to guide reinforcement learning, providing fine-grained, step-aware supervision beyond offline data selection or test-time ranking.

Settings. We integrate Fin-PRM into the Group Relative Policy Optimization (GRPO) framework [Shao *et al.*, 2024], using Qwen2.5-7B-Instruct as the policy model. The evaluation is conducted on two financial reasoning benchmarks: *CFLUE* and *FinQA*. We compare three reward sources:

- **Outcome-only reward**: using the final-answer correctness signal r^{out} by comparing the predicted option answer a with the gold-standard answer y;
- **General-domain PRM**: Qwen2.5-Math-PRM-7B;
- **Fin-PRM (ours)**: the composite reward $\hat{R}$ in Eq. 15 combining step-level and trajectory-level guidance.

Methodology. To enable step-aware supervision, we define a composite RL reward for a reasoning trajectory (s, a) as:

$$r_{\text{rl}} = (1 - \delta) \cdot r^{\text{out}} + \delta \cdot \hat{R}, \tag{16}$$

where δ balances the contribution of process-level reward. For a group of N candidate trajectories, the advantage is computed as:

$$A_{\text{rl}} = \frac{r_{\text{rl}} - \text{mean}(\{r_{\text{rl}}\}_{j=1}^{N})}{\text{std}(\{r_{\text{rl}}\}_{j=1}^{N})}. \tag{17}$$

The GRPO objective is then updated to optimize the policy with respect to A_{rl}, including the clipping and KL-penalty terms:

$$\begin{aligned} \mathcal{J}_{\text{GRPO}}(\theta) = \mathrm{E}_{x,\{s_i\}\sim\pi_{\theta_{\text{old}}}} \Bigg[& \frac{1}{N}\sum_{i=1}^{N} \frac{1}{T_i}\sum_{t=1}^{T_i} \Bigg(\\ & \min\Bigg\{ \frac{\pi_\theta(s_{i,t}|x,s_{i,<t})}{\pi_{\theta_{\text{old}}}(s_{i,t}|x,s_{i,<t})} A_{\text{rl},i}, \\ & \text{clip}\Big(\frac{\pi_\theta(s_{i,t}|x,s_{i,<t})}{\pi_{\theta_{\text{old}}}(s_{i,t}|x,s_{i,<t})}, 1-\epsilon, 1+\epsilon\Big) A_{\text{rl},i} \Bigg\} \\ & - \beta_{\text{KL}} \mathcal{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \Bigg) \Bigg]. \end{aligned} \tag{18}$$

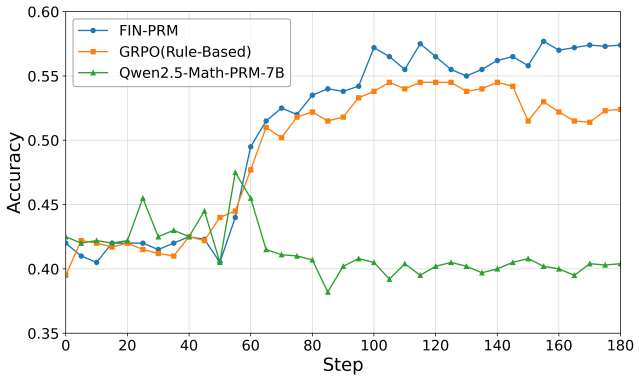

Figure 3: Learning curves of GRPO-based policy optimization using different reward signals. All policies are initialized from Qwen2.5-7B-Instruct, and performance is reported as mean accuracy on the CFLUE test set over multiple runs.

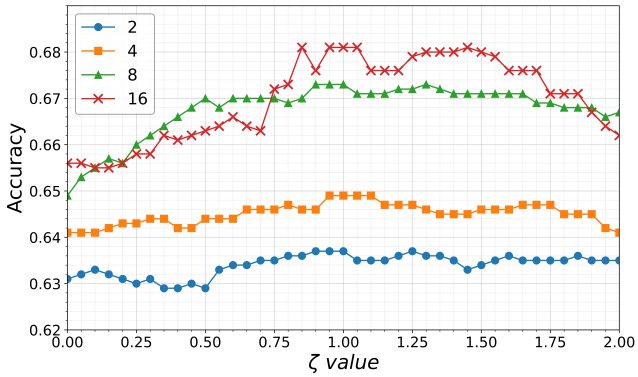

Figure 4: Ablation study of the ranking score weight ζ. The figure reports Best-of-N accuracy on the CFLUE test set as a function of ζ for different values of N.

Results. Figure 3 shows learning curves on the CFLUE test set for policies optimized with different reward sources. In the early training stage (steps 0–60), all three strategies exhibit similar performance, suggesting comparable initial learning dynamics. In the later phase (steps 60–100), clear differences emerge: policies trained with Fin-PRM and the rule-based reward improve rapidly, while Qwen2.5-Math-PRM-7B shows shartly decrease. Throughout training, Fin-PRM consistently achieves higher accuracy and reaches the best final performance around step 100, indicating that domain-specialized, process-aware rewards provide more effective guidance than outcome-only heuristics or general-domain PRMs for financial reasoning.

Using Fin-PRM as the reward signal yields the strongest policy across evaluations, reaching an accuracy of 70.5% on *CFLUE* and 62.8% on *FinQA*. Compared to optimization with an outcome-only reward, this corresponds to an absolute improvement of 3.3%, and it also surpasses the performance obtained with the general-domain PRM baseline. These results indicate that domain-specific, process-aware supervision provided by Fin-PRM offers more effective learning signals for reinforcement learning in financial reasoning tasks.

6 Ablation Study

As defined in Eq. 15, the hyperparameter ζ controls the relative contribution of the aggregated step-level reward ($\hat{R}_{\text{step}}$) and the trajectory-level reward ($\hat{R}_{\text{traj}}$) when computing the final ranking score for candidate solutions. This ranking score is a core component shared across all three applications of Fin-PRM, including offline data selection, test-time Best-of-N (BoN) selection, and reinforcement learning.

In this section, we use BoN selection as a representative and controlled setting to systematically study the effect of ζ, as it isolates the ranking behavior of Fin-PRM without introducing additional confounding factors from model retraining.

Settings. We conduct BoN selection on a 1,000-sample subset of the CFLUE test set, using the same fine-tuned Qwen2.5-7B-Instruct model as the response generator. For $N \in \{2, 4, 8, 16\}$, we vary ζ in the range $[0.0, 2.0]$ and report the resulting selection accuracy.

Results and Analysis. The results are shown in Figure 4. Overall, performance is sensitive to the choice of ζ, particularly for larger values of N (i.e., $N = 8$ and $N = 16$). Several consistent trends emerge:

- When $\zeta = 0$, the ranking relies solely on step-level rewards. While this setting yields reasonable performance, it is consistently suboptimal, indicating that local step correctness alone is insufficient for identifying the best reasoning trajectory.
- Performance peaks around $\zeta \approx 1.0$, where step-level and trajectory-level rewards are weighted approximately equally. This suggests that effective evaluation requires a balance between fine-grained reasoning quality and global coherence.
- As ζ increases further, accuracy gradually degrades. Overemphasizing trajectory-level scores while down-weighting step-level signals can lead to the selection of trajectories that appear globally plausible but contain critical local errors.

These results provide empirical support for our dual-granularity reward design, demonstrating that balanced integration of step-level and trajectory-level supervision is essential for effective ranking across all downstream applications of Fin-PRM.

7 Conclusion

In this work, we introduce Fin-PRM, a domain-specialized, knowledge-aware process reward model that improves financial reasoning in LLMs via step- and trajectory-level supervision. Trained on 3,000 curated trajectories with verifiable reward annotations, Fin-PRM supports offline data selection, reward-guided inference, and reinforcement learning. Experiments on CFLUE and FinQA show that models trained or guided by Fin-PRM achieve significant gains, highlighting the effectiveness of domain-specific, process-aware reward modeling.

References

[Bai *et al.*, 2022] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. *Computing Research Repository*, 2022.

[Cui *et al.*, 2025] Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards. *Computing Research Repository*, 2025.

[Gu *et al.*, 2025] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. A survey on llm-as-a-judge. *Computing Research Repository*, 2025.

[Guha *et al.*, 2025] Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, et al. Openthoughts: Data recipes for reasoning models. *Computing Research Repository*, 2025.

[Gunasekar *et al.*, 2023] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need. *Computing Research Repository*, 2023.

[Guo *et al.*, 2025] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

[He *et al.*, 2024] Jujie He, Tianwen Wei, Rui Yan, Jiacai Liu, Chaojie Wang, Yimeng Gan, Shiwen Tu, Chris Yuhao Liu, Liang Zeng, Xiaokun Wang, Boyang Wang, Yongcong Li, Fuxiang Zhang, Jiacheng Xu, Bo An, Yang Liu, and Yahui Zhou. Skywork-o1 open series. *Computing Research Repository*, 2024.

[Khalifa *et al.*, 2025] Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. Process reward models that think. *Computing Research Repository*, 2025.

[Lightman *et al.*, 2024] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *Proceedings of ICLR*, 2024.

[Liu *et al.*, 2025] Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen Zhou. Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling. *Computing Research Repository*, 2025.

[Luo *et al.*, 2025] Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. In *Proceedings of ICLR*, 2025.

[Mukherjee *et al.*, 2023] Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4. *Computing Research Repository*, 2023.

[Setlur *et al.*, 2025] Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. In *Proceedings of ICLR*, 2025.

[Shao *et al.*, 2024] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Computing Research Repository*, 2024.

[Wang *et al.*, 2024] Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of ACL*, pages 9426–9439, 2024.

[Wei and Zou, 2019] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of EMNLP*, pages 6382–6388, 2019.

[Xie *et al.*, 2023] Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. In *Proceedings of NeurIPS*, pages 69798–69818, 2023.

[Yang *et al.*, 2023] Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. Fingpt: Open-source financial large language models. *FinLLM Symposium at IJCAI*, 2023.

[Ye *et al.*, 2025] Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning. In *Proceedings of COLM*, 2025.

[Yu *et al.*, 2024] Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. In *Proceedings of ICLR*, 2024.

[Zhang *et al.*, 2025a] Kaiyan Zhang, Jiayuan Zhang, Haoxin Li, Xuekai Zhu, Ermo Hua, Xingtai Lv, Ning Ding, Biqing Qi, and Bowen Zhou. Openprm: Building open-domain process-based reward models with preference trees. In *Proceedings of ICLR*, 2025.

[Zhang *et al.*, 2025b] Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. In *Findings of ACL*, pages 10495–10516, 2025.

[Zheng *et al.*, 2023] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of NeurIPS*, pages 46595–46623, 2023.

[Zhu *et al.*, 2024] Jie Zhu, Junhui Li, Yalong Wen, and Lifan Guo. Benchmarking large language models on CFLUE - a Chinese financial language understanding evaluation dataset. In *Findings of ACL*, pages 5673–5693, 2024.

[Zhu *et al.*, 2025] Jie Zhu, Qian Chen, Huaixia Dou, Junhui Li, Lifan Guo, Feng Chen, and Chi Zhang. Dianjin-r1: Evaluating and enhancing financial reasoning in large language models. *Computing Research Repository*, 2025.

[Zou *et al.*, 2025] Jiaru Zou, Ling Yang, Jingwen Gu, Jiahao Qiu, Ke Shen, Jingrui He, and Mengdi Wang. Reasonflux-prm: Trajectory-aware prms for long chain-of-thought reasoning in llms. In *Proceedings of NeurIPS*, pages 26764–26802, 2025.