

# PRInTS: Reward Modeling for Long-Horizon Information Seeking

Jaewoo Lee<sup>1</sup> Archiki Prasad<sup>1</sup> Justin Chih-Yao Chen<sup>1</sup>

Zaid Khan<sup>1</sup> Elias Stengel-Eskin<sup>2</sup> Mohit Bansal<sup>1</sup>

University of North Carolina at Chapel Hill<sup>1</sup>, University of Texas at Austin<sup>2</sup>

jwoolee@cs.unc.edu archiki@cs.unc.edu

## Abstract

Information-seeking is a core capability for AI agents, requiring them to gather and reason over tool-generated information across long trajectories. However, such multi-step information-seeking tasks remain challenging for agents backed by language models. While process reward models (PRMs) can guide agents by ranking candidate steps at test-time, existing PRMs – designed for short reasoning with binary judgment – cannot capture richer dimensions of information-seeking steps, such as tool interactions and reasoning over tool outputs, nor handle the rapidly growing context in long-horizon tasks. To address these limitations, we introduce PRInTS, a generative PRM trained with dual capabilities: (1) dense scoring based on the PRM’s reasoning across multiple dimensions of step quality (e.g., interpretation of tool outputs, tool call informativeness) and (2) trajectory summarization that compresses the growing context while preserving essential information for step evaluation. Extensive evaluations across FRAMES, GAIA (levels 1-3), and WebWalkerQA (easy-hard) benchmarks on multiple models reveal that best-of- $n$  sampling with PRInTS enhances information-seeking in open-source models as well as specialized agents, matching or surpassing frontier models with a much smaller backbone agent and outperforming other strong reward modeling baselines.<sup>1</sup>

## 1 Introduction

A long-standing goal in artificial intelligence has been to develop agents that can answer novel queries by intelligently seeking information (Bachman et al., 2016; Yuan et al., 2020), thereby enabling them to tackle challenging tasks in mathematics (Liu et al., 2025a,b), software engineering (Yang et al., 2025b; Pan et al., 2024), and research (Li et al., 2025b; Wu et al., 2025a). Large

Language Models (LLMs) have shown promise as agents for such tasks when equipped with frameworks like ReAct (Yao et al., 2023), which interleaves LLM reasoning with external tool interactions. However, long-horizon information-seeking tasks, which require agents to gather and synthesize information across multiple steps (Su et al., 2025; Shao et al., 2025), remain challenging, even for recent LLMs with tool-use training, performing far below human-level (Mialon et al., 2024; Wei et al., 2025). While finetuning LLMs as information-seeking agents has shown promise (Li et al., 2025b; Wu et al., 2025a), it is limited to specific model families and is highly computationally demanding (Gao et al., 2025). An alternative way to boost a variety of agents is to build reward models (e.g., as done for math reasoning and instruction following (Wang et al., 2024a; Zou et al., 2025)). These models approximate the expected reward of a step or sequence of steps, enabling test-time scaling by ranking and selecting higher-quality actions or trajectories to successfully tackle long-horizon tasks. Specifically, Process Reward Models (PRMs) (Zou et al., 2025; Choudhury, 2025) offer a promising model-agnostic way of improving performance, scoring the quality of each of an agent’s steps.

While past work has developed PRMs for tasks such as mathematics and logical reasoning, these methods are insufficient for long-horizon information-seeking tasks for two critical reasons. (1) **Tool-Reasoning Evaluation Granularity:** existing PRMs evaluate short reasoning units in isolation, typically one- to two-sentence logical or mathematical inferences (Xiong et al., 2025; Zhao et al., 2025), providing binary judgments based on logical/math validity. In contrast, long-horizon information-seeking requires jointly evaluating a complete trajectory step, which encompasses a reasoning step combined with tool interactions (e.g., web search, web browsing, code execution). Moreover, step quality depends on multiple factors (e.g.,

<sup>1</sup>Code available at <https://github.com/G-JWLee/PRInTS>

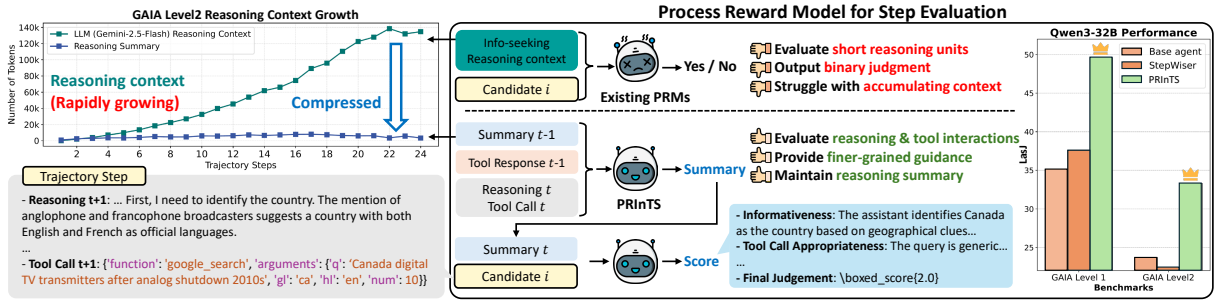


Figure 1: Comparison between existing PRMs and PRInTS. **Top:** Existing PRMs are limited for long-horizon information-seeking as they evaluate a short reasoning unit (e.g., one-to-two-sentence inferences) with coarse feedback, which cannot capture multi-faceted quality factors from tool interactions. They also struggle with rapidly accumulating reasoning context (left). **Bottom:** In contrast, PRInTS evaluates a complete trajectory step (reasoning + tool interactions), considers multiple trajectory step quality dimensions to produce dense scores for finer-grained guidance at each step, and maintains compact trajectory summaries that keep key information for the evaluation.

interpretation of tool outputs, tool call informativeness, next-action plan) that coarse feedback cannot capture, requiring more granular guidance to effectively steer agents toward good trajectories. (2) **Context Accumulation:** existing PRMs cannot manage the ever-growing reasoning context arising over multiple trajectory steps. As illustrated in Figure 1 (Top-left), the information-seeking trajectory – interleaving reasoning steps, tool calls, and tool call outputs – grows rapidly as tool responses at each step introduce lengthy content. Since models struggle to process such long, accumulated contexts (Tang et al., 2025; Yen et al., 2025; Kuratov et al., 2024), often resulting in noisy evaluations, it necessitates compressing trajectories into compact forms rather than processing the full history.

Our work aims to fill these gaps by introducing **Process Reward** via **Information gain** scoring and **Trajectory Summarization** (PRInTS), a novel generative PRM for long-horizon information-seeking tasks. PRInTS is a unified model jointly trained with two key abilities to address both the need for fine-grained guidance and the challenge of context accumulation. These two abilities are learned jointly within the same PRM. First, PRInTS acts as a scorer that evaluates candidate next trajectory steps by generating Chain-of-Thought (Wei et al., 2022) analyses across multiple quality dimensions and outputting dense scores derived from this generative reasoning, as illustrated in (Figure 1 (Bottom)). Crucially, we frame step evaluation as information gain estimation that quantifies how much each trajectory step increases the probability of reaching the correct answer. This formulation enables training via reinforcement learning with information gain estimation and preference prediction objectives, providing richer reward signals that

account for the multi-faceted quality of trajectory steps. At test-time, PRInTS evaluates  $n$  candidate next steps, selecting the step expected to yield the greatest information gain. Second, PRInTS simultaneously functions as a summarizer, recursively generating and updating a compact trajectory summary at each step. PRInTS compresses the query, previous summary, latest tool response, and current step into an updated summary that captures essential findings and plans up to the current timestep (Figure 1 (Bottom)). This keeps input length bounded, as shown in Figure 1 (Top-left), while preserving information for subsequent evaluation.

To equip PRInTS with these dual capabilities, we first design preference and summary data that can produce supervision signals needed to train the scoring and summarization components. Specifically, our annotation pipeline (Figure 3) uses Monte Carlo rollouts (Wang et al., 2024a; Setlur et al., 2025) to estimate information gain scores and construct preference trajectory step pairs, and generates compact trajectory summaries for each step. Next, we use this annotated data to train PRInTS to score steps via reinforcement learning with two complementary rewards: (1) a **Score Reward** that teaches the model to analyze the trajectory step quality and estimate the step’s information gain score, and (2) a **Comparison Reward** that teaches the model to assign higher scores to preferred trajectory steps by learning from pairwise preferences. These rewards enable the model to capture the multi-faceted quality of the trajectory step and perform dense step-level evaluation. We jointly train PRInTS via supervised fine-tuning for summarization that recursively updates the trajectory summary based on the previous summary and the most recent reasoning context at each step, directly ad-

addressing the context accumulation challenge while preserving key information needed for step-level evaluation. Together, this pipeline enables PRInTS to serve as a unified PRM capable of both managing long, noisy trajectories and providing fine-grained test-time guidance.

We validate our approach across three distinct LLMs used as information-seeking agents: Qwen3-32B (Yang et al., 2025a), Tongyi DeepResearch-30B-A3B (Li et al., 2025a) – a specialized information-seeking agent – and Gemini-2.5-Flash (Gemini Team, 2025), evaluated on three long-horizon information-seeking benchmarks: FRAMES, GAIA, and WebWalkerQA. The experimental results show that PRInTS, a 4B PRM, consistently provides test-time gains across diverse agents – Qwen3-32B by 9.3%, DeepResearch-30B-A3B by 3.9%, and Gemini-2.5-Flash by 4.0% absolute average accuracy – without fine-tuning the underlying models. Unlike existing PRMs, which obtain diminished and inconsistent gains as agents become stronger, PRInTS continues to deliver substantial improvements. Notably, on GAIA (levels 1-3), PRInTS raises DeepResearch-30B-A3B from 61.9% to 64.4% in our implementation, enabling the 30B agent augmented with the 4B PRM to outperform Deepseek-V3.1-671B (63.1%) — which is 20 times larger — while also closing the gap to a strong proprietary system like OpenAI DeepResearch (67.4%). Furthermore, our ablation studies reveal that providing compressed summaries outperforms using raw trajectories as input context, showing that context management is essential for accurate step-level evaluation in long-horizon tasks. Overall, our approach enhances information-seeking abilities of pretrained open-source models as well as specialized agents, showing strong generalizability.

## 2 Related Work

**Large Language Models (LLM) as Agents.** LLMs have been increasingly adopted as agents through frameworks such as ReAct (Yao et al., 2023), which interleaves reasoning and tool use to solve complex tasks (Deng et al., 2025; Wu et al., 2025c). To facilitate effective information-seeking behaviors, recent studies, such as WebSailor (Li et al., 2025b), WebShaper (Tao et al., 2025), and DeepResearch (Li et al., 2025a), improve the intrinsic quality of information-seeking trajectories by training models on synthetic data

to reduce search space uncertainty. However, such methods share key limitations: they require (1) substantial supervision (MiroMind AI Team, 2025; Li et al., 2025a) and (2) access to model weights, which pose challenges for generalization whenever the underlying model changes. We empirically show that PRInTS enhances information-seeking capabilities of agents via test-time guidance with strong generalizability, offering an orthogonal yet mutually beneficial direction to agent fine-tuning.

**Reward Models for Reasoning.** Outcome Reward Models (ORMs) predict the correctness of complete reasoning trajectories (Kim et al., 2024; Pan et al., 2024) but cannot provide finer-grained, step-wise guidance over partial trajectories. Process Reward Models (PRMs) address this limitation by evaluating individual steps (Ton et al., 2024; He et al., 2025; Chen et al., 2025). Recent advancements cast PRMs as generative judges (Wang et al., 2024b; Whitehouse et al., 2025; He et al., 2025) that generate justifications for step scores and have achieved strong performance in mathematics (Zhao et al., 2025; Xiong et al., 2025; Wang et al., 2024a), finance (Zhou et al., 2025), and agentic tasks (Chae et al., 2025; Choudhury, 2025). In contrast to these existing PRM approaches that rank the validity of short reasoning snippets or struggle with managing growing contexts, PRInTS is equipped with jointly evaluating reasoning with tool interactions, planning for subsequent actions across multiple dimensions of “information gain” in tandem with a compact trajectory summarization mechanism, inspired by reasoning context compression approaches (Wu et al., 2025c; Kang et al., 2025; Ye et al., 2025).

## 3 Process Reward via Information Gain Score and Trajectory Summarization

We start by introducing the framework that quantifies and annotates the quality of each trajectory step – a reasoning step combined with a tool call – followed by reinforcement learning that uses these annotations to train PRInTS as a scorer (Section 3.2). Next, we describe our approach for generating compact summaries of long interaction trajectories, and explain how these summaries are used to train the same PRInTS as a summarizer (Section 3.3). The overall design of PRInTS is illustrated in Figure 2.

### 3.1 Preliminaries

To tackle long-horizon information-seeking problems, we adopt the agentic ReAct (Yao et al., 2023)

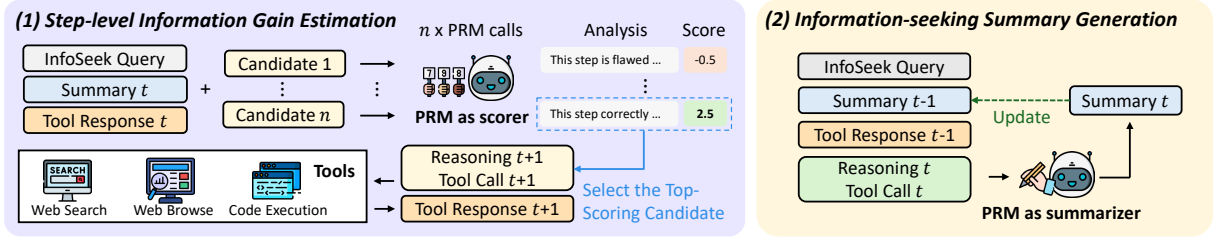


Figure 2: **Overview of PRInTS.** **Left:** PRInTS functions as a scorer, evaluating agent’s multiple candidate next trajectory steps based on the summarized context and current tool response. It generates an analysis and a dense score for each candidate, selecting the top-scoring one to guide the agent’s information-seeking. **Right:** PRInTS acts as a summarizer, recursively updating a compact information-seeking trajectory summary to preserve key information for subsequent score evaluation.

paradigm, where a Large Language Model (LLM) acts as an agent that interleaves reasoning with tool-based action toward the goal of answering query  $q$  (Tao et al., 2025; Li et al., 2025a). At each timestep  $t$ , the agent may generate intermediate reasoning  $s_t$  based on the current context and then predict the subsequent action  $a_t$ , i.e., calls to external tools, such as web search, web browsing, and code execution, to acquire new information. The resulting tool response  $o_t$  is observed and added to the context, which informs the agent’s reasoning at timestep  $t + 1$ . Figure 2 (Left) visually shows this tool interaction process:  $s_t, a_t, o_t$  correspond to reasoning, tool call, tool response at timestep  $t$ . This process repeats until timestep  $T$ , when the agent submits its answer  $o_T$ , which is successful if  $o_T$  matches the ground-truth answer  $a^*$ . The accumulated reasoning context up to timestep  $t$ , termed the information-seeking trajectory, is defined as:

$$H_t = (s_1, a_1, o_1, s_2, a_2, o_2, \dots, s_t, a_t, o_t) \quad (1)$$

Specifically, the agent  $\pi$  generates the next reasoning step and tool call conditioned on the query and information-seeking trajectory, i.e.,  $s_t, a_t \sim \pi(\cdot | q, H_{t-1})$ . Then the tool call is executed to get the tool response  $o_t$ . This interleaving of reasoning and action has shown success in long-horizon information-seeking tasks (Li et al., 2025b; Gao et al., 2025; Li et al., 2025a), and thus we adopt this setting. However, applying PRMs here faces two key challenges: trajectory steps contain substantially richer content than traditional steps, requiring multi-dimensional evaluation beyond simple correctness, and rapidly growing context  $H_t$  introduces noise that complicates evaluation. Thus, we introduce our data annotation and train pipeline, which equips PRInTS with two core capabilities: (1) dense step-level scoring for fine-grained guidance, and (2) trajectory summarization for effective step-level evaluation under context accumulation.

### 3.2 Step-level Information Gain Estimation

**Information Gain Score.** To train a Process Reward Model (PRM) for long-horizon information seeking, we measure how much each reasoning step and tool call contributes towards reaching the correct answer. To this end, we define information gain of the current step as the change in expected likelihood of arriving at the correct answer before and after taking the current step (Rao and III, 2018; Prasad et al., 2023; Wang et al., 2024a). This local evaluation quantifies the marginal improvement in task success contributed by the current step. Specifically, for a reasoning step and tool call  $(s_t, a_t)$  preceded by information-seeking trajectory prefix,  $H_{t-1}$ , we use Monte Carlo estimation (Wang et al., 2024a; Xiong et al., 2025; Setlur et al., 2025) by executing  $M$  rollouts until their final answers are produced and compute the mean accuracy:

$$m_t = \frac{\sum_{j=1}^M \mathbb{1}(o_{T_j}^{(j)} = a^*)}{M}, \quad (2)$$

where  $o_{T_j}^{(j)} \sim \pi(\cdot | q, H_{t-1}, s_t, a_t)$  is the final answer from rollout  $j$ , which terminates at timestep  $T_j$ . The information gain score  $g_t$  is then computed as:

$$g_t = (m_t - m_{t-1}) \times M/2, \quad (3)$$

which quantifies how much  $(s_t, a_t)$  contributes to the successful completion of the task. We scale by  $M/2$  to map the scores into the interval  $[-M/2, M/2]$  with discrete increments of 0.5. This normalization provides a more intuitive understanding of the relative quality differences across steps. A positive  $g_t$  indicates that the current step  $(s_t, a_t)$  increases the probability of reaching the correct answer – for example, through logically coherent reasoning or a tool call that resolves uncertainties – whereas a  $g_t$  lower than zero indicates that the current step reduces the probability, e.g., by making unverified assumptions or invoking an irrelevant tool call.

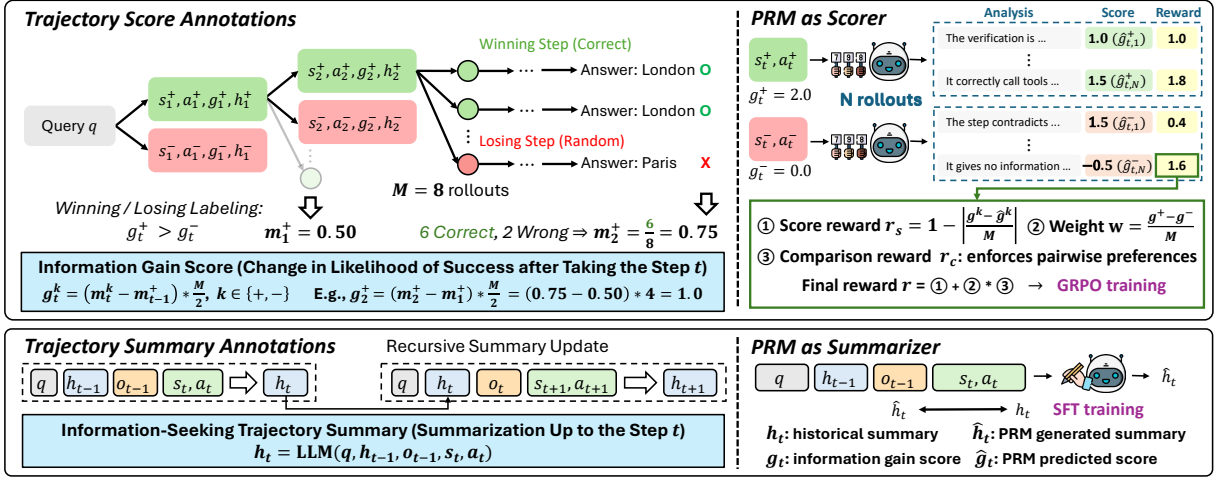


Figure 3: **PRInTS: data annotation and training pipeline.** **Top:** For each trajectory step, we estimate the information gain score via Monte Carlo rollouts as the change in mean answer accuracy before and after the step. Then we construct winning-losing step pairs based on these scores (left). Preference pair examples are shown in Figure 8. Then we train PRInTS as a scorer via GRPO on these pairs (right). The final reward combines a score reward for accurate prediction, a comparison reward for pairwise preference learning, and an adaptive weight to mitigate noisy annotations. **Bottom:** Each step is annotated with a compact, recursively updated trajectory summary capturing essential findings and plans up to the step (left). The same PRM is jointly trained as a summarizer via SFT on this summary data (right).

**Trajectory Score Annotations.** While prior approaches annotate individual trajectory in isolation (Xiong et al., 2025; Wang et al., 2024a) or use imitation learning on step-level scores (Wang et al., 2024a), Whitehouse et al. (2025) shows that pairwise preference learning builds robust judges. We extend this by automatically constructing preference pairs grounded in information gain scores, then training PRInTS with complementary objectives – a score reward for information gain estimation and a comparison reward for preference prediction – as illustrated in Figure 3 (Top).

During the  $M$  rollouts for  $g_t$ , the LLM generates a set of  $M$  unique next trajectory steps  $\{s_{t+1}^{(j)}, a_{t+1}^{(j)}\}_{j=1}^M$  and the corresponding final answer predictions  $\{o_{T_j}^{(j)} \sim \pi(\cdot | q, H_t, s_{t+1}^{(j)}, a_{t+1}^{(j)})\}_{j=1}^M$ . As shown in Figure 3 (Top-left), to construct a candidate preference pair, we first select the step that yields a successful trajectory, assuming this step has the highest potential to be effective among the  $M$  rollouts. A second trajectory step is then randomly sampled from the remaining steps, serving as a contrasting, less effective (i.e., incorrect or inefficient) alternative.

Next, we annotate the information gain scores of this candidate preference pair by treating each trajectory step as a new starting point and running  $M$  rollouts to estimate their respective mean accuracies and information gain scores. After annotation, we reassign preference labels based on these scores:

the step with the higher score becomes the winning sample  $(s_{t+1}^+, a_{t+1}^+)$ , while the other becomes the losing sample  $(s_{t+1}^-, a_{t+1}^-)$ . The winning step then serves as the starting point for generating the next pair at step  $t + 2$ . This contrastive labeling ensures that PRInTS learns relative preferences between trajectory steps grounded in empirical improvements.

**Training the PRM as a Scorer.** The core function of PRInTS is to assess trajectory step quality and assign higher scores to steps expected to yield greater information gain. To this end, we train PRInTS to evaluate step quality by predicting information gain scores. Given a query  $q$ , trajectory summary  $h_{t-1}$  (introduce in Section 3.3 below), latest tool response  $o_{t-1}$ , and trajectory step  $(s_t, a_t)$ , PRInTS generates a Chain-of-Thought analysis and outputs a scalar score  $\hat{g}_t = f_I(q, h_{t-1}, o_{t-1}, s_t, a_t)$ , where  $f_I$  denotes PRInTS that works as an information gain scorer function. We train this scoring capability using GRPO (Shao et al., 2024) with the following rewards: (1) a score reward ( $r_s$ ) that targets minimizing the discrepancy between the predicted score ( $\hat{g}_t$ ) and the ground-truth score ( $g_t$ ), and (2) a comparison reward ( $r_c$ ) that enforces pairwise preferences derived from annotated pairs:

$$r_s^k = 1 - \frac{|g^k - \hat{g}^k|}{M}, r_c^k = \sum_{j=1}^N \frac{y^k \cdot \text{sgn}(\hat{g}^k - \hat{g}_j^k)}{N}, \quad (4)$$

where  $\text{sgn}(x) = \begin{cases} +1, & x \geq 0 \\ -1, & x < 0 \end{cases}$ ,  $k \in \{+, -\}$  indicates the winning or losing sample,  $y^+ = 1$  and  $y^- = -1$  for comparison direction,  $\bar{k}$  denotes its counterpart,  $N$  is the number of rollouts, and  $\hat{g}_j^k$  is the predicted score of  $j$ -th rollout from the counterpart. For simplicity, we omit rollout- and step-level indices, which do not affect the underlying formulation of the scores. The comparison reward ensures PRInTS learns to distinguish better from worse reasoning paths, while the score reward provides fine-grained feedback on estimation accuracy. Finally, we combine the two rewards into a single scalar per rollout with an adaptive weight:

$$r^k = r_s^k + w * r_c^k, \quad w = \frac{g^+ - g^-}{M}, \quad (5)$$

where  $w$  is the comparison reward weight set for each pair. This adaptive weighting scheme addresses noise in the automatically annotated pairs. Pairs with large score margins ( $g^+ - g^-$ ) are more reliably ranked and receive higher weights, while pairs with small margins receive lower weights, as they may reflect annotation noise rather than true preference (Prasad et al., 2024). The combined reward thus encourages PRInTS to estimate absolute scores accurately and to learn robust preferences in information-seeking trajectories. The illustration of reward computation is shown in Figure 7.

### 3.3 Info-seeking Summary Generation

**Trajectory Summary Annotations.** Another core challenge in building a PRM for information-seeking agents is the rapidly growing context (Figure 1 Top-left) of lengthy and noisy reasoning and tool interactions. This context explosion hinders PRMs from efficient processing and results in noise and distraction in quality evaluation. To address this, we extract a concise summary of the information-seeking trajectory of each trajectory step  $(s_t, a_t)$ . This summary,  $h_t$ , captures the essential findings and plan development up to timestep  $t$ . As illustrated in Figure 3 (Bottom-left), each summary is recursively updated and generated by an LLM, incorporating the previous summary  $h_{t-1}$ , latest tool response  $o_{t-1}$ , and current trajectory step  $(s_t, a_t)$  (i.e.,  $h_t = \text{LLM}(q, h_{t-1}, o_{t-1}, s_t, a_t)$ ). This recursive formulation ensures that  $h_t$  maintains a compressed form of the entire trajectory  $H_t$ .

**Training the PRM as a Summarizer.** To enable efficient processing of the context during the score

estimation, PRInTS learns to generate a concise summary  $\hat{h}_t = f_S(q, h_{t-1}, o_{t-1}, s_t, a_t)$  that retains essential information, where  $f_S$  is PRInTS that works as a summarization function. We use supervised fine-tuning (SFT) on the annotated summaries  $h_t$ , effectively teaching the model to compress the context by imitating the annotations.

## 4 Experiments

**Models.** To evaluate the efficacy and generalizability of PRInTS, we use three distinct LLMs: Qwen3-32B (Yang et al., 2025a), an open-source model with strong reasoning capability; Gemini-2.5-Flash (Gemini Team, 2025), a closed-source frontier model; and Tongyi DeepResearch-30B-A3B (Li et al., 2025a), a recently developed agent specifically optimized for long-horizon information-seeking tasks. We instantiate ReAct-based agents using these LLMs.

**Benchmarks and Metrics.** Following recent work on long-horizon information-seeking (Gao et al., 2025; Li et al., 2025c,a), we assess PRM-guided reasoning on three benchmarks: FRAMES (Krishna et al., 2025), GAIA (Mialon et al., 2024), and WebWalkerQA (Wu et al., 2025b). We evaluate both Qwen3-32B and DeepResearch agent across all three benchmarks, while Gemini-2.5-Flash is evaluated on GAIA. Further explanations of these benchmarks are provided in Appendix A. Following past work (Gao et al., 2025; Li et al., 2025c,a), we adopt the LLM-as-Judge (LasJ) paradigm to measure benchmark performance, which is a standard approach in long-horizon information-seeking research. We use GPT-5 to judge the correctness of final answers. All results are reported using Avg@3, defined as the mean accuracy over three independent runs.

**Baselines.** We compare PRInTS against three categories: **(1) Base agent:** the unguided LLM, serving as a reference point for test-time improvement. **(2) Intrinsic reasoning heuristics:** widely used reasoning quality heuristics, including confidence (Ghasemabadi et al., 2025), relevance (Wan et al., 2025), and verbal-progress. **(3) Existing PRMs:** GenPRM-7B (Zhao et al., 2025), WebShepherd-8B (Chae et al., 2025), StepWiser (Xiong et al., 2025). We reimplement StepWiser, which outputs binary judgment, using the same annotations and backbone model as PRInTS to provide a controlled comparison (c.f. Appendix A). This lets

Method	FRAMES	GAIA			WebWalkerQA			Avg.
		Level 1	Level 2	Level 3	Easy	Medium	Hard	
Base agent	49.3	35.1	23.7	11.1	30.1	26.9	30.3	29.5
Confidence	55.7	36.8	24.4	<u>16.7</u>	31.7	31.3	32.9	<u>32.8</u>
Relevance	<u>56.3</u>	34.2	20.5	8.3	<u>33.3</u>	29.5	32.5	30.7
Verbal-progress	45.0	35.9	21.2	13.9	27.6	30.2	34.2	29.7
GenPRM-7B	50.0	32.5	<u>25.7</u>	<u>16.7</u>	<u>33.3</u>	<u>32.8</u>	<u>34.6</u>	32.2
Web-Shepherd-8B	49.0	<u>38.5</u>	23.7	5.5	28.5	31.8	33.3	30.0
StepWiser	51.3	37.6	22.4	8.3	31.7	31.8	33.8	31.0
<b>PRInTS</b>	<b>58.7</b>	<b>49.6</b>	<b>33.3</b>	<b>19.4</b>	<b>39.8</b>	<b>33.3</b>	<b>37.3</b>	<b>38.8</b>

Table 1: Comparison of step quality evaluation methods on Qwen3-32B across information-seeking benchmarks. We adopt the LLM-as-Judge (LasJ) metric and report Avg@3. The best and the second best results are in **bold** and underline, respectively. PRInTS delivers consistent gains all benchmarks, whereas the second-best baseline varies.

Method	FRAMES	GAIA			WebWalkerQA			Avg.
		Level 1	Level 2	Level 3	Easy	Medium	Hard	
Base agent	79.3	68.4	61.6	<u>41.7</u>	61.8	59.5	68.0	62.9
Confidence	61.3	60.7	47.5	25.0	63.4	62.0	64.9	55.0
Relevance	81.3	<b>70.1</b>	63.5	33.3	<u>66.7</u>	62.8	66.7	63.5
Verbal-progress	<b>82.3</b>	<u>69.2</u>	60.9	<u>41.7</u>	63.4	<u>63.8</u>	68.4	<u>64.2</u>
GenPRM-7B	79.0	<b>70.1</b>	<u>64.1</u>	38.9	60.2	<u>63.8</u>	<u>68.9</u>	63.6
Web-Shepherd-8B	79.7	<u>69.2</u>	61.5	36.1	62.6	62.3	67.1	62.6
StepWiser	81.0	<b>70.1</b>	60.9	36.1	65.0	61.8	64.9	62.8
<b>PRInTS</b>	<u>81.7</u>	<u>69.2</u>	<b>65.4</b>	<b>44.5</b>	<b>70.7</b>	<b>65.9</b>	<b>70.1</b>	<b>66.8</b>

Table 2: Comparison of step quality evaluation methods on Tongyi DeepResearch-30B-A3B across information-seeking benchmarks. We adopt the LLM-as-Judge (LasJ) metric and report Avg@3. The best and the second best results are in **bold** and underline, respectively. The results show that PRInTS enhances the performance of strong information-seeking agents.

us isolate the contributions of dense comparative scoring and compact trajectory summarization.

**Implementation Details.** We use Qwen3-32B to annotate information gain score and summary for each step. We initialize PRInTS with Qwen3-4B, training with one epoch of SFT for the summarization objective, followed by a period of GRPO training for the scoring objective. This SFT-GRPO cycle is repeated for a total of X iterations. At test-time, the LLM policy generates  $n = 4$  next steps which are evaluated via the baselines and PRInTS, and the highest scoring candidate is selected. Further details can be found in [Appendix A](#).

#### 4.1 Results and Discussion

**PRInTS substantially outperforms existing PRMs on foundation models.** [Table 1](#) shows PRInTS consistently achieves substantial gains over the Qwen3-32B base agent across all long-horizon information-seeking benchmarks. Step-wiser yields only a 1.5% absolute average accuracy gain compared to 9.3% with PRInTS despite us-

ing the same data, highlighting the limitations of the coarse supervision binary correctness provides. Yet, expressiveness alone is insufficient: even baselines with richer outputs – Verbal-progress (scalar) and Web-Shepherd (multi-item checklists) – add only marginal gains of 0.2% and 0.5%, respectively. In contrast, PRInTS is trained to derive multi-factor analyses before outputting a dense comparative score grounded in both information-gain estimation and pairwise preference learning, letting it identify subtle yet key quality differences and select the most informative steps.

**PRInTS improves highly performant information-seeking agents.** In [Table 2](#), we evaluate PRInTS on DeepResearch-30B-A3B, a specialized agent optimized for long-horizon information-seeking tasks. The results show that adding PRInTS to this strong information-seeking agent consistently achieves performance gains across benchmarks, surpassing the base agent by 3.9% absolute average accuracy, while no other baselines come close to achieving notable improvements. No-

Method	GAIA			Avg.
	Level 1	Level 2	Level 3	
Base agent	58.1	42.3	<u>19.5</u>	40.0
Relevance	58.1	44.9	<u>19.5</u>	40.8
Verbal-progress	<u>60.7</u>	44.2	16.7	40.5
GenPRM-7B	56.4	44.9	11.1	37.5
Web-Shepherd-8B	59.8	<b>46.2</b>	16.7	40.9
StepWiser	<u>60.7</u>	44.3	<u>19.5</u>	<u>41.5</u>
<b>PRInTS</b>	<b>61.5</b>	<u>45.5</u>	<b>25.0</b>	<b>44.0</b>

Table 3: **PRInTS shows strong generalization to the frontier LLM (Gemini).** We adopt the LLM-as-Judge (LasJ) metric and report Avg@3 comparing with other step quality evaluation methods on Gemini-2.5-Flash.

tably, PRInTS improves on the challenging subsets, such as GAIA Level 3 and WebWalkerQA Hard. Moreover, on GAIA, PRInTS lifts DeepResearch-30B-A3B from 61.9% to 64.4% average accuracy, enabling the 30B agent augmented with the 4B PRM to reach competitive performance with OpenAI DeepResearch (67.4%) and surpass DeepSeek-V3.1-671B (63.1%) (Li et al., 2025a).<sup>2</sup> This test-time scaling approach is substantially more efficient than scaling the base model; it achieves frontier-level performance while requiring significantly less memory and computational overhead than deploying the 20-times-larger 671B model. Crucially, while specialized information-seeking agents (Wu et al., 2025a; Li et al., 2025b,c) require resource-intensive fine-tuning with massive datasets (10k-100k+ samples) and costly online RL (tool interactions and multi-step rollouts) (Gao et al., 2025), PRInTS achieves these gains using a 4B model that either does not require a large dataset (2k+ pair samples) or tool interactions and long-horizon rollouts during training. This demonstrates that step-level guidance provided by PRInTS is a cost-effective strategy to push the performance limits of even highly optimized information-seeking agents.

**PRInTS also generalizes to frontier LLMs.** To further demonstrate the versatility of our approach, we use the closed-source Gemini-2.5-Flash as the LLM agent, as shown in Table 3. PRInTS provides 4.0% absolute average accuracy gain, whereas the second-best method improves performance by only 1.5%. On the most challenging subset, GAIA Level 3, PRInTS yields the largest improvement among all baselines (+5.5%), showing its strength on long-

<sup>2</sup>See Appendix A for frontier model results reporting details.

Input Context	FRAMES	GAIA		Avg.
		Level 1	Level 2	
$H_{-1}$ :	56.3	44.5	25.7	42.2
$H_{-2}$ :	<b>61.0</b>	44.5	26.9	44.1
$H_{-4}$ :	57.0	37.6	25.0	39.9
$H_t$	55.7	38.5	24.4	39.5
$h_t$ (Ours)	58.7	<b>49.6</b>	<b>33.3</b>	<b>47.2</b>

Table 4: **Effectiveness of context compression.** Comparison of input context representations for PRM on Qwen3-32B across information-seeking tasks.  $H_{-1}$ ,  $H_{-2}$ , and  $H_{-4}$ : provide the most recent one, two, and four trajectory steps from the full trajectory  $H_t$ , while  $h_t$  uses the trajectory summary from PRInTS. Our approach ( $h_t$ ) shows better scoring ability by retaining essential information for step evaluation.

horizon reasoning tasks. Overall, our results indicate that PRInTS provides effective test-time guidance, improving the information-seeking behavior of both open-source LLMs, closed-source LLMs, and information-seeking agents, which shows strong versatility and generalizability without modifying or retraining underlying LLMs.

## 4.2 Analysis and Ablations

**Summarization ability contributes to better scoring ability.** To validate the effectiveness of compressed representations for accurate step-level scoring, we compare our summarization approach against several alternatives: providing the most recent one, two, or four steps as input context ( $H_{-1}$ ,  $H_{-2}$ ,  $H_{-4}$ ), and the full trajectory ( $H_t$ ). We evaluate on Qwen3-32B across FRAMES and GAIA. Since performance on GAIA Level 3 is low (see Table 1), we use Levels 1 and 2. Results in Table 4 show that our summarization approach achieves the best or second-best performance across benchmarks, outperforming the full raw-history baseline by 7.7% absolute average accuracy.

Notably, extending raw history does not improve performance.  $H_{-2}$ : outperforms both  $H_{-1}$ : (insufficient context),  $H_{-4}$ :, and  $H_t$  (excessive, noisy context). This confirms that long histories introduce noise and irrelevant information that distract step-level evaluation. In contrast, our summarization compresses entire trajectories into compact representations that preserve key information while filtering noise, enabling accurate scoring even as trajectories grow arbitrarily long.

**Complementary rewards improve step-level evaluation.** We analyze the contribution of each reward component in Equation (5). Following the setup from the previous ablation, we evaluate on

Reward Design	FRAMES	GAIA		Avg.
		Level 1	Level 2	
$r = r_s$ (score-only)	57.0	43.6	32.0	44.2
$r = r_c$ (comparison-only)	58.7	41.9	28.8	43.1
$r = r_s + r_c$ (combination)	<b>60.3</b>	47.0	31.4	46.2
$r = r_s + w \cdot r_c$ (Ours)	58.7	<b>49.6</b>	<b>33.3</b>	<b>47.2</b>

Table 5: **Impact of reward components.** Experiments with reward components on PRM performance across information-seeking tasks evaluated with Qwen3-32B. Combining the score reward with comparison reward ( $r_s + r_c$ ) leads to better step evaluation, with further improvement when mitigating noise in preference pairs through adaptive weight ( $r_s + w \cdot r_c$ ).

Qwen3-32B across FRAMES, GAIA Levels 1, and 2. As shown in Table 5, combining the score and comparison rewards ( $r_s + r_c$ ) yields substantially better performance than using either component alone, leading to 2.0% and 3.1% absolute average accuracy gains compared to using the score reward ( $r_s$ ) and comparison reward ( $r_c$ ), respectively. This indicates that information-gain estimation and preference prediction capture complementary aspects of the quality of a trajectory step, underscoring the benefit of our pairwise annotation strategy over prior work that labels individual steps in isolation (Xiong et al., 2025; Wang et al., 2024a). Furthermore, incorporating the adaptive weight ( $r_s + w \cdot r_c$ ) yields 1.0% additional absolute average accuracy gain over the naive combination. This is because the adaptive weight mitigates noise in preference pairs. Pairs with small information gain score differences are inherently noisier, as they may reflect annotation variance rather than true quality differences. Thus, these pairs receive lower weights, while pairs with clear margins are weighted higher, leading to more stable learning. Overall, adaptive weighting provides a simple and cost-efficient way of leveraging existing annotated preference pairs.

**PRInTS scales effectively with test-time compute.** In order to evaluate how PRInTS benefits from additional test-time compute, we conduct best-of- $n$  scaling experiments with varying numbers of candidate steps ( $n \in \{1, 2, 4, 8, 16\}$ ) on GAIA Level 2 using Qwen3-32B, as shown in Figure 4. PRInTS exhibits strong scaling behavior, achieving 2.5%, 3.8%, 8.9% absolute accuracy gains at  $n = 2, 4, 8$ , respectively, demonstrating that PRInTS reliably identifies higher-quality steps within large candidate sets. However, performance declines at  $n = 16$ . We attribute this to the ex-

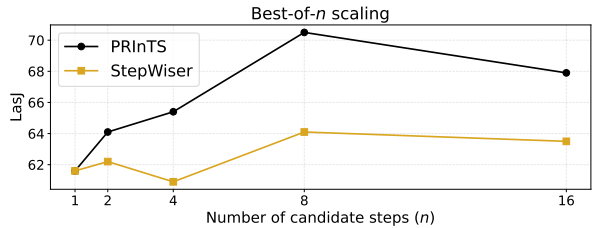


Figure 4: **Scaling test-time compute.** Best-of- $n$  test-time scaling results on GAIA Level 2 using Tongyi DeepResearch-30B-A3B. PRInTS benefits from additional test-time compute by identifying higher-quality steps from  $n$  candidates.

panded candidate pool. A larger  $n$  increases the likelihood of generating steps that superficially appear to resolve uncertainty or offer high information gain. Hence, it becomes increasingly prone to selecting these seemingly informative exploratory steps over candidates that correctly output the final answer. Consequently, the agent abandons shorter, successful trajectories and continues exploring until it reaches the maximum step budget, failing to output an answer despite having generated one earlier. In contrast, StepWiser provides only marginal and inconsistent improvements under scaling. This difference in scaling efficiency further validates that PRInTS’s design of information-gain estimation and preference prediction captures subtle quality differences between steps, enabling fine-grained guidance for long-horizon information-seeking.

## 5 Conclusion

In this paper, we introduce PRInTS, a generative PRM for long-horizon information seeking. PRInTS unifies information gain scoring with recursive trajectory summarization, enabling fine-grained step-level evaluation under rapidly accumulating agent context. To equip PRInTS with these dual abilities, we construct preference step pairs with information gain scores and summaries and jointly train PRInTS by alternating supervised fine-tuning for summarization and reinforcement learning for scoring. We test PRInTS on three strong and distinct agents and demonstrate that PRInTS consistently enhances their information-seeking abilities, showcasing its high versatility. PRInTS also improves frontier information-seeking agents, showing that test-time guidance is a powerful complement to agent fine-tuning and robust to changes in underlying models.

## Limitations

While our pipeline automates the annotation of information gain scores and summaries, the underlying QA samples we collected are focused on English queries. Expanding coverage to multilingual contexts remains a promising direction to further enhance PRInTS’s versatility. Another compelling avenue for future research is to employ PRInTS as a dense reward signal for training information-seeking agents via reinforcement learning. We leave this exploration for future work due to its significant computational demands, focusing instead on demonstrating test-time improvements.

## Ethical Considerations

In this work, we suggest PRInTS that can improve the information-seeking ability of agent systems by providing finer-grained evaluation of intermediate reasoning and tool interactions, potentially reducing erroneous or inefficient information-seeking behavior. By improving agents’ ability to gather and synthesize information accurately, our work could benefit applications such as scientific research and knowledge discovery. We do not foresee any ethical implications beyond those applicable to agentic research systems more generally.

## Acknowledgments

This work was supported by NSF-AI Engage Institute DRL-2112635, NSF-CAREER Award 1846185, DARPA ECOLE Program No. HR00112390060, Capital One Research Award, Apple PhD Fellowship, NDSEG PhD Fellowship. The views contained in this article are those of the authors and not of the funding agency.

## References

UK AI Security Institute. 2024. [Inspect AI: Framework for Large Language Model Evaluations](#).

Philip Bachman, Alessandro Sordani, and Adam Trischler. 2016. [Towards information-seeking agents](#). *arXiv preprint arXiv:1612.02605*.

Hyungjoo Chae, Sunghwan Kim, Junhee Cho, Seungone Kim, Seungjun Moon, Gyeom Hwangbo, Dongha Lim, Minjin Kim, Yeonjun Hwang, Minju Gwak, Dongwook Choi, Minseok Kang, Gwanhoon Im, ByeongUng Cho, Hyojun Kim, Jun Hee Han, Taeyoon Kwon, Minju Kim, Beong-woo Kwak, Dongjiin Kang, and Jinyoung Yeo. 2025. [Web-shepherd: Advancing prms for reinforcing web agents](#). *arXiv preprint arXiv:2505.15277*.

Justin Chen, Archiki Prasad, Swarnadeep Saha, Elias Stengel-Eskin, and Mohit Bansal. 2025. [MAGCoRe: Multi-agent, iterative, coarse-to-fine refinement for reasoning](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Sanjiban Choudhury. 2025. [Process reward models for LLM agents: Practical framework and directions](#). *arXiv preprint arXiv:2502.10325*.

Yong Deng, Guoqing Wang, Zhenzhe Ying, Xiaofeng Wu, Jinzhen Lin, Wenwen Xiong, Yuqin Dai, Shuo Yang, Zhanwei Zhang, Qiwen Wang, Yang Qin, Yuan Wang, Quanxing Zha, Sunhao Dai, and Changhua Meng. 2025. [Atom-searcher: Enhancing agentic deep research via fine-grained atomic thought reward](#). *arXiv preprint arXiv:2508.12800*.

Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. 2025. [Deep think with confidence](#). *arXiv preprint arXiv:2508.15260*.

Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. 2025. [Beyond ten turns: Unlocking long-horizon agentic search with large-scale asynchronous RL](#). *arXiv preprint arXiv:2508.07976*.

Gemini Team. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). *arXiv preprint arXiv:2507.06261*.

Amirhosein Ghasemabadi, Keith G. Mills, Baochun Li, and Di Niu. 2025. [Guided by gut: Efficient test-time scaling with reinforced intrinsic confidence](#). *arXiv preprint arXiv:2505.20325*.

Tao He, Rongchuan Mu, Lizi Liao, Yixin Cao, Ming Liu, and Bing Qin. 2025. [Good learners think their thinking: Generative PRM makes large reasoning model more efficient math learner](#). *arXiv preprint arXiv:2507.23317*, abs/2507.23317.

Minki Kang, Wei-Ning Chen, Dongge Han, Huseyin A. Inan, Lukas Wutschitz, Yanzhi Chen, Robert Sim, and Saravan Rajmohan. 2025. [Acon: Optimizing context compression for long-horizon llm agents](#). *arXiv preprint arXiv:2510.00615*.

Byoungjip Kim, Youngsoo Jang, Lajanugen Logeswaran, Geon-Hyeong Kim, Yu Jin Kim, Honglak Lee, and Moontae Lee. 2024. [Prospector: Improving LLM agents with self-asking and trajectory ranking](#). In *Findings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananeey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. 2025. [Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation](#). In *Proceedings of the Association for Computational Linguistics (ACL)*.

- Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Y. Sorokin, and Mikhail Burtsev. 2024. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, Kuan Li, and Yong Jiang et al. 2025a. [Tongyi deepresearch technical report](#). *arXiv preprint arXiv:2510.24701*.
- Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, Weizhou Shen, Junkai Zhang, Dingchu Zhang, Xixi Wu, Yong Jiang, Ming Yan, Pengjun Xie, Fei Huang, and Jingren Zhou. 2025b. Websailor: Navigating superhuman reasoning for web agent. *arXiv preprint arXiv:2507.02592*.
- Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. 2025c. [Webthinker: Empowering large reasoning models with deep research capability](#). *arXiv preprint arXiv:2504.21776*.
- Fan Liu, Zherui Yang, Cancheng Liu, Tianrui Song, Xiaofeng Gao, and Hao Liu. 2025a. [Mm-agent: LLM as agents for real-world mathematical modeling problem](#). *arXiv preprint arXiv:2505.14148*.
- Xianyang Liu, Yilin Liu, Shuai Wang, Hao Cheng, Andrew Estornell, Yuzhi Zhao, and Jiaheng Wei. 2025b. [Agenticmath: Enhancing LLM reasoning via agentic-based math data generation](#). *arXiv preprint arXiv:2510.19361*.
- Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2024. GAIA: a benchmark for general AI assistants. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- MiroMind AI Team. 2025. Mirothinker: An open-source agentic model series trained for deep research and complex, long-horizon problem solving. <https://github.com/MiroMindAI/MiroThinker>.
- MiroMind Data Team. 2025. [Miroverse v0.1: A reproducible, full-trajectory, ever-growing deep research dataset](#).
- Jiayi Pan, Xingyao Wang, Graham Neubig, Navdeep Jaitly, Heng Ji, Alane Suhr, and Yizhe Zhang. 2024. [Training software engineering agents and verifiers with swe-gym](#). *arXiv preprint arXiv:2412.21139*.
- Mihir Prabhudesai, Lili Chen, Alex Ippoliti, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. 2025. [Maximizing confidence alone improves reasoning](#). *arXiv preprint arXiv:2505.22660*.
- Archiki Prasad, Swarnadeep Saha, Xiang Zhou, and Mohit Bansal. 2023. Reveal: Evaluating reasoning chains via correctness and informativeness. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Archiki Prasad, Weizhe Yuan, Richard Yuanzhe Pang, Jing Xu, Maryam Fazel-Zarandi, Mohit Bansal, Sainbayar Sukhbaatar, Jason Weston, and Jane A. Yu. 2024. [Self-consistency preference optimization](#). *arXiv preprint arXiv:2411.04109*.
- Sudha Rao and Hal Daumé III. 2018. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In *Proceedings of the Association for Computational Linguistics (ACL)*.
- Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. 2025. Rewarding progress: Scaling automated process verifiers for LLM reasoning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muennighoff, Xi Victoria Lin, Daniela Rus, Bryan Kian Hsiang Low, Sewon Min, Wen-tau Yih, Pang Wei Koh, and Luke Zettlemoyer. 2025. [Reasonir: Training retrievers for reasoning tasks](#). *arXiv preprint arXiv:2504.20595*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *arXiv preprint arXiv:2402.03300*.
- Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han-yu Wang, Haisu Liu, Quan Shi, Zachary S. Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Sercan Ö. Arik, Danqi Chen, and Tao Yu. 2025. BRIGHT: A realistic and challenging benchmark for reasoning-intensive retrieval. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Zecheng Tang, Baibei Ji, Quantong Qiu, Haitian Wang, Xiaobo Liang, Juntao Li, and Min Zhang. 2025. [Longrm: Revealing and unlocking the context boundary of reward modeling](#). *arXiv preprint arXiv:2510.06915*.
- Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu Wang, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. 2025. [Webshaper: Agentic data synthesizing via information-seeking formalization](#). *arXiv preprint arXiv:2507.15061*.
- Jean-Francois Ton, Muhammad Faaiz Taufiq, and Yang Liu. 2024. [Understanding chain-of-thought in llms through information theory](#). *arXiv preprint arXiv:2411.11984*.

- Guangya Wan, Yuqi Wu, Jie Chen, and Sheng Li. 2025. Reasoning aware self-consistency: Leveraging reasoning paths for efficient LLM sampling. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*.
- Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024a. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the Association for Computational Linguistics (ACL)*.
- Tianlu Wang, Iliia Kulikov, Olga Golovneva, Ping Yu, Weizhe Yuan, Jane Dwivedi-Yu, Richard Yuanzhe Pang, Maryam Fazel-Zarandi, Jason Weston, and Xian Li. 2024b. [Self-taught evaluators](#). *arXiv preprint arXiv:2408.02666*.
- Xi Wang, James McInerney, Lequn Wang, and Nathan Kallus. 2025. [Entropy after  \$\langle /Think \rangle\$  for reasoning model early exiting](#). *arXiv preprint arXiv:2509.26522*.
- Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. [Browsecomp: A simple yet challenging benchmark for browsing agents](#). *arXiv preprint arXiv:2504.12516*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Chenxi Whitehouse, Tianlu Wang, Ping Yu, Xian Li, Jason Weston, Iliia Kulikov, and Swarnadeep Saha. 2025. [J1: incentivizing thinking in llm-as-a-judge via reinforcement learning](#). *arXiv preprint arXiv:2505.10320*.
- Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. 2025a. [Webdancer: Towards autonomous information seeking agency](#). *arXiv preprint arXiv:2505.22648*.
- Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. 2025b. Webwalker: Benchmarking llms in web traversal. In *Proceedings of the Association for Computational Linguistics (ACL)*.
- Junde Wu, Jiayuan Zhu, Yuyuan Liu, Min Xu, and Yueming Jin. 2025c. Agentic reasoning: A streamlined framework for enhancing LLM reasoning with agentic tools. In *Proceedings of the Association for Computational Linguistics (ACL)*.
- Wei Xiong, Wenting Zhao, Weizhe Yuan, Olga Golovneva, Tong Zhang, Jason Weston, and Sainbayar Sukhbaatar. 2025. [Stepwiser: Stepwise generative judges for wiser reasoning](#). *arXiv preprint arXiv:2508.19229*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and et al. 2025a. [Qwen3 technical report](#). *arXiv preprint arXiv:2505.09388*.
- John Yang, Kilian Leret, Carlos E. Jimenez, Alexander Wettig, Kabir Khandpur, Yanzhe Zhang, Binyuan Hui, Ofir Press, Ludwig Schmidt, and Diyi Yang. 2025b. [Swe-smith: Scaling data for software engineering agents](#). *arXiv preprint arXiv:2504.21798*.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Rui Ye, Zhongwang Zhang, Kuan Li, Huifeng Yin, Zhengwei Tao, Yida Zhao, Liangcai Su, Liwen Zhang, Zile Qiao, Xinyu Wang, Pengjun Xie, Fei Huang, Siheng Chen, Jingren Zhou, and Yong Jiang. 2025. [Agentfold: Long-horizon web agents with proactive context management](#). *arXiv preprint arXiv:2510.24699*.
- Howard Yen, Ashwin Paranjape, Mengzhou Xia, Thejas Venkatesh, Jack Hessel, Danqi Chen, and Yuhao Zhang. 2025. [Lost in the maze: Overcoming context limitations in long-horizon agentic search](#). *arXiv preprint arXiv:2510.18939*.
- Xingdi Yuan, Jie Fu, Marc-Alexandre Côté, Yi Tay, Chris Pal, and Adam Trischler. 2020. Interactive machine comprehension with information seeking agents. In *Proceedings of the Association for Computational Linguistics (ACL)*.
- Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, and Bowen Zhou. 2025. [Genprm: Scaling test-time compute of process reward models via generative reasoning](#). *arXiv preprint arXiv:2504.00891*.
- Yuanchen Zhou, Shuo Jiang, Jie Zhu, Junhui Li, Lifan Guo, Feng Chen, and Chi Zhang. 2025. [Fin-prm: A domain-specialized process reward model for financial reasoning in large language models](#). *arXiv preprint arXiv:2508.15202*.
- Jiaru Zou, Ling Yang, Jingwen Gu, Jiahao Qiu, Ke Shen, Jingrui He, and Mengdi Wang. 2025. [Reasonflux-prm: Trajectory-aware prms for long chain-of-thought reasoning in llms](#). *arXiv preprint arXiv:2506.18896*.

## A Details of Experimental Setups

**Tool Use.** We conduct all experiments within the Inspect-Eval evaluation framework (AI Security Institute, 2024), which fully supports the ReAct paradigm (Yao et al., 2023) for multi-turn reasoning and tool interactions necessary for complex information-seeking tasks. The framework provides access to a comprehensive set of tools:

- **Web Search:** We utilize the Serper search API to retrieve up-to-date web content for search queries.
- **Web Browsing:** The framework includes built-in browser automation tools, supporting essential web interaction functions such as browsing with URLs, clicking, scrolling down/up, typing, etc.
- **Code Execution:** The framework also supports built-in Python and Bash code execution environments.

In this evaluation framework, agents use these tools to interact with external sources to search and synthesize information to solve given information-seeking questions.

**Evaluation Benchmarks.** We provide in-depth explanations of the long-horizon information-seeking benchmarks used in our experiments. (1) GAIA (Mialon et al., 2024) evaluates the ability to act as a general AI assistant on complex retrieval and reasoning tasks spanning three difficulty levels. Following prior work (Li et al., 2025c; Gao et al., 2025; Kang et al., 2025), we use 103 questions from the text-only validation subset. (2) WebWalkerQA (Wu et al., 2025b) focuses on web-based reasoning, requiring agents to traverse webpages to locate target information across three difficulty levels. We evaluate on 247 English questions. (3) FRAMES (Krishna et al., 2025) provides factual and reasoning-intensive queries to assess both retrieval and reasoning capabilities. We use a subset that consists of 300 samples that are randomly selected from the original dataset.

**Baselines.** In this section, we provide a more detailed explanation of the baselines.

- **GenPRM-7B** (Zhao et al., 2025) is a generative PRM originally designed for mathematical reasoning. It produces Chain-of-Thought rationales with a binary verdict (yes / no) indicating

whether the current step is correct. We follow their prompt format and ask GenPRM to verify the correctness of a trajectory step and explain why the step is judged correct or incorrect.

- **Web-Shepherd-8B** (Chae et al., 2025) generates a task-specific checklist that decomposes a task into key subgoals and evaluates agentic trajectories based on it. Specifically, it assigns coarse feedback labels (Yes / No / In progress) for each checklist item for evaluation. We also follow their prompt formats to generate a checklist for a given information-seeking task and evaluate each trajectory step relative to that checklist.
- **StepWiser** (Xiong et al., 2025) trains a generative PRM using GRPO with binary rewards, where each step is labeled as effective or ineffective. For implementation, we follow the *Relative Effective Reward Thresholding* in the paper to re-annotate our training dataset: a step receives a positive label if the ratio between the current and previous mean accuracies exceeds the threshold (0.7), and a negative label otherwise. Using this binary supervision, we train Qwen3-4B for 4 epochs to build StepWiser PRM.
- **Confidence** (Ghasemabadi et al., 2025; Fu et al., 2025; Prabhudesai et al., 2025; Wang et al., 2025) estimates reasoning quality based on the model’s certainty. Following the confidence definition in recent work (Ghasemabadi et al., 2025; Fu et al., 2025), we calculate confidence by taking the negative average log-probability of the top-10 tokens across all generated token positions in the reasoning step. Higher scores indicate lower uncertainty.
- **Relevance** (Wan et al., 2025) measures the coherence between the current step and the preceding context. Specifically, it uses the Jaccard similarity between the current step and the accumulated past steps. A higher similarity indicates better contextual coherence.
- **Verbal progress** is a zero-shot baseline that assesses progress toward the final answer by prompting Qwen3-4B to estimate how close the current reasoning state is to completing the task. The model is asked to output a scalar that ranges from 1 to 5 based on the textual content of the current step and its information-seeking trajectory. A higher score indicates that the current reasoning state is close to the final answer.

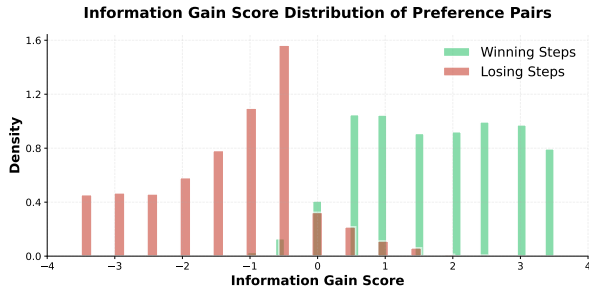


Figure 5: Distribution of annotated information gain scores.

**Train Configurations.** We train PRInTS using Qwen3-4B with an alternating SFT-GRPO schedule over four cycles, where each cycle consists of one SFT epoch for summarization followed by one GRPO epoch for scoring, to jointly acquire both abilities. A single SFT epoch requires approximately 0.5 hours and a single GRPO epoch requires 3.5 hours, totaling 16 hours on a single node with 8x NVIDIA RTX 6000 Ada GPUs (48GB). For the SFT stage, we use a batch size of 128 and a learning rate of  $1e-6$ . For the GRPO stage, we execute  $N = 4$  rollouts and use a batch size of 128 and a learning rate of  $1e-6$ . Such alternating optimization allows the model to continuously refine its summarization ability while simultaneously improving its scoring accuracy on reasoning quality. This iterative schedule ensures both modules evolve synergistically, stabilizing training and preventing either skill from degrading over time.

**Data Annotations.** We construct our annotated data from 4,344 information-seeking questions, comprising 720 questions used to train information-seeking agents from the Alibaba group (Wu et al., 2025a; Li et al., 2025b; Tao et al., 2025), and 3,624 questions from the Miroverse-v0.1 dataset (MiroMind Data Team, 2025), a large-scale agent dataset covering multi-hop QA, web navigation, and scientific reasoning tasks. For the score annotation process, we execute  $M = 8$  rollouts per trajectory step to estimate the mean accuracy and information gain score. We discard steps that are either too easy ( $m_t = 1$ ) or too hard ( $m_t = 0$ ), resulting in 2,294 preference pairs used for training.

As shown in Figure 5, our annotation pipeline produces well-balanced information gain score distributions across the full score range. This balanced distribution ensures that PRInTS learns to estimate diverse trajectory step quality, from harmful steps that reduce success probability to highly effective steps that substantially advance toward the correct

Method	FRAMES	GAIA		Avg.
		Level 1	Level 2	
Qwen3-32B	54.7	44.5	29.5	42.9
PRInTS	<b>58.7</b>	<b>49.6</b>	<b>33.3</b>	<b>47.2</b>

Table 6: **Ablation study of summarizer.** The Qwen3-32B approach utilizes Qwen3-32B as a summarizer and employs the PRM trained solely as a scorer. PRInTS that simultaneously acts as a summarizer and scorer shows better performance, showing that two abilities are complementary.

answer. Moreover, the distributions of winning and losing steps exhibit clear separation. This clear separation validates the effectiveness of our preference pair construction and using this as training signals for PRInTS to output dense and comparative scores.

**Frontier Model Performance.** For the performance of frontier models on the GAIA benchmark, we follow the reported results (OpenAI DeepResearch: 67.4% and DeepSeek-V3.1-671B: 63.1%) in the DeepResearch-30B-A3B paper (Li et al., 2025a) and use these reported numbers as reference points when evaluating improvements brought by integrating PRInTS into DeepResearch-30B-A3B.

## B Additional Experiments

**PRInTS as Summarizer.** To validate our design of jointly training scoring and summarization within a single model, we compare PRInTS against a variant that uses separate models for each capability. Specifically, we train a PRM using only GRPO for scoring without SFT for summarization, and pair it with Qwen3-32B – the same model we use for summary annotation – to generate summaries at test-time. As shown in 6, PRInTS, which is jointly trained as both scorer and summarizer through our alternating SFT-GRPO schedule, outperforms this separated design. This demonstrates that the two abilities are complementary and that our alternating training schedule enables seamless integration of these abilities. We hypothesize that this benefit arises from positive transfer between the two objectives. As both abilities operate on the same input (i.e., query, preceding summary, latest tool response, current trajectory step), learning to distill essential information during SFT directly aids GRPO optimization by highlighting the most relevant factors for quality evaluation.

**Dataset Scaling** To further validate that PRInTS is a cost-efficient way of improving information-seeking behaviors of agents without fine-tuning

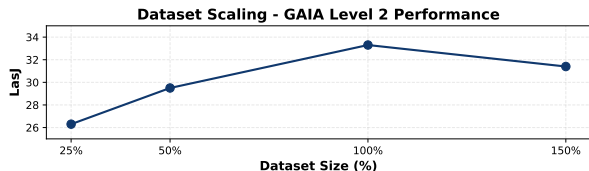


Figure 6: **Dataset scaling.** Experiments on the impact of dataset scaling on GAIA Level 2 using Qwen3-32B. Training PRInTS shows strong sample efficiency, achieving performance gain using only 50% (~1k samples) of our annotation data.

them, we conduct an ablation study on the training dataset scaling. As shown in Figure 6, using only 50% (i.e., ~ 1k samples) of our annotation data still achieves 29.5 accuracy in GAIA Level 2, surpassing the base agent approach by training a relatively lightweight model (4B). The scaling curve saturates beyond 100%, indicating that our ~2k preference pairs represent a data-efficient point. This demonstrates strong sample efficiency compared to fine-tuning agents, which typically require 10k-100k samples and expensive long-horizon rollouts using relatively larger models, requiring substantial computational resources.

**Stability of Information Gain Estimation.** As information gain in our pipeline is estimated via Monte Carlo rollouts, we conduct an empirical study to quantify the stochastic noise of this annotation process. Using a subset of 400 query samples from our source dataset, we measure the mean accuracy (Equation (2)) for each trajectory step. Treating a larger rollout budget ( $M = 32$ ) as a pseudo-ground truth reference, we bootstrap smaller subsets ( $M \in \{4, 8, 16\}$ ) with  $k = 1000$  resamples to quantify estimation variance. We observe that the variance of the estimated mean accuracy decreases significantly as the rollout count increases: 0.025 at  $M = 4$ , 0.011 at  $M = 8$  and 0.004 at  $M = 16$ . Similarly, the Mean Absolute Error (MAE) relative to the  $M = 32$  reference drops from 0.097 to 0.037. To evaluate the downstream impact on preference learning, we calculate the preference flip rate — how often rollout noise alters the preferred step relative to the pseudo-ground truth. This rate is notably low, decreasing from 3.62% ( $M = 4$ ) to 1.81% ( $M = 16$ ). These results indicate that the quality differences between winning and losing steps typically exceed Monte Carlo noise, confirming that our supervision signal reflects robust quality distinctions rather than sampling variance.

**Impact of Annotation Policy Model.** We further analyze the effect of the rollout policy model’s capability on the annotation pipeline by comparing Qwen3-32B and Qwen3-14B on the same 400-query subset ( $M = 8$ ). Both models yield consistent average information gain scores (Equation (3)) for winning and losing samples (1.87 and -1.21 for Qwen3-32B; 1.73 and -1.16 for Qwen3-14B), demonstrating that our pipeline generates stable preference signals across model scales. However, because our annotation process terminates when a step reaches 100% or 0% success, a sufficiently capable model is required to maximize the learning signal. The 32B model maintains a non-trivial success probability over more steps, thereby avoiding premature termination. Consequently, Qwen3-32B generates a higher yield of preference pairs (240 vs. 169) and exhibits greater score diversity, where Qwen3-32B results in standard deviations of 1.27 and 1.30 for winning and losing samples, compared to 1.20 and 1.03 for the 14B model. This better alignment between policy capability and query difficulty justifies our selection of Qwen3-32B as the optimal annotator.

**Qualitative Analysis of Failure Cases.** To better understand PRInTS, we inspected information-seeking trajectories from challenging subsets (i.e., GAIA Levels 2 and 3) generated by the Qwen3-32B agent and PRInTS. Our analysis reveals that PRInTS occasionally exhibits a failure mode of over-conservatism. Qualitatively, the model frequently assigns negative informativeness scores to final answers if the agent arrives at them through speculation or with unresolved uncertainties, rather than relying on explicit, verifiable evidence.

While this conservative behavior is generally beneficial for preventing the agent from concluding with unverified answers, it can inadvertently force the agent to discard a correct heuristic shortcut. Consequently, the agent is compelled to continue searching to clear any remaining uncertainties, which prolongs the trajectory, thereby increasing the risk of context overflow or step-limit time-outs. For instance, in a representative GAIA Level 3 trajectory, the agent correctly guesses the target answer "mice" but is penalized by PRInTS for lacking definitive textual evidence. As a result, the model favors continuing finding concrete evidence over terminating the trajectory. This shows a fundamental trade-off in our approach between strict factual grounding and trajectory efficiency.

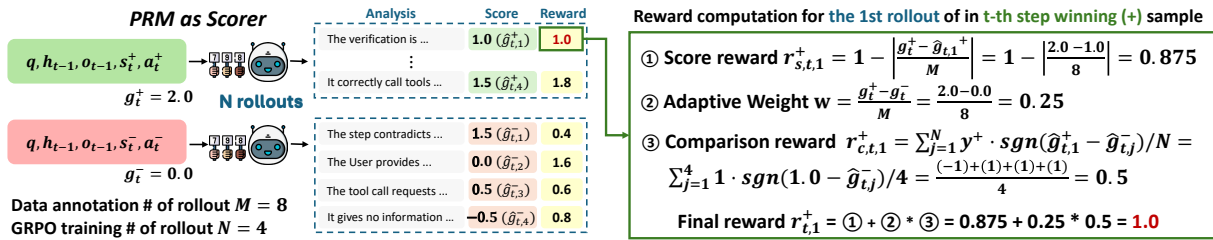


Figure 7: Example of reward computation for the first rollout of the t-th step winning sample during GRPO training.

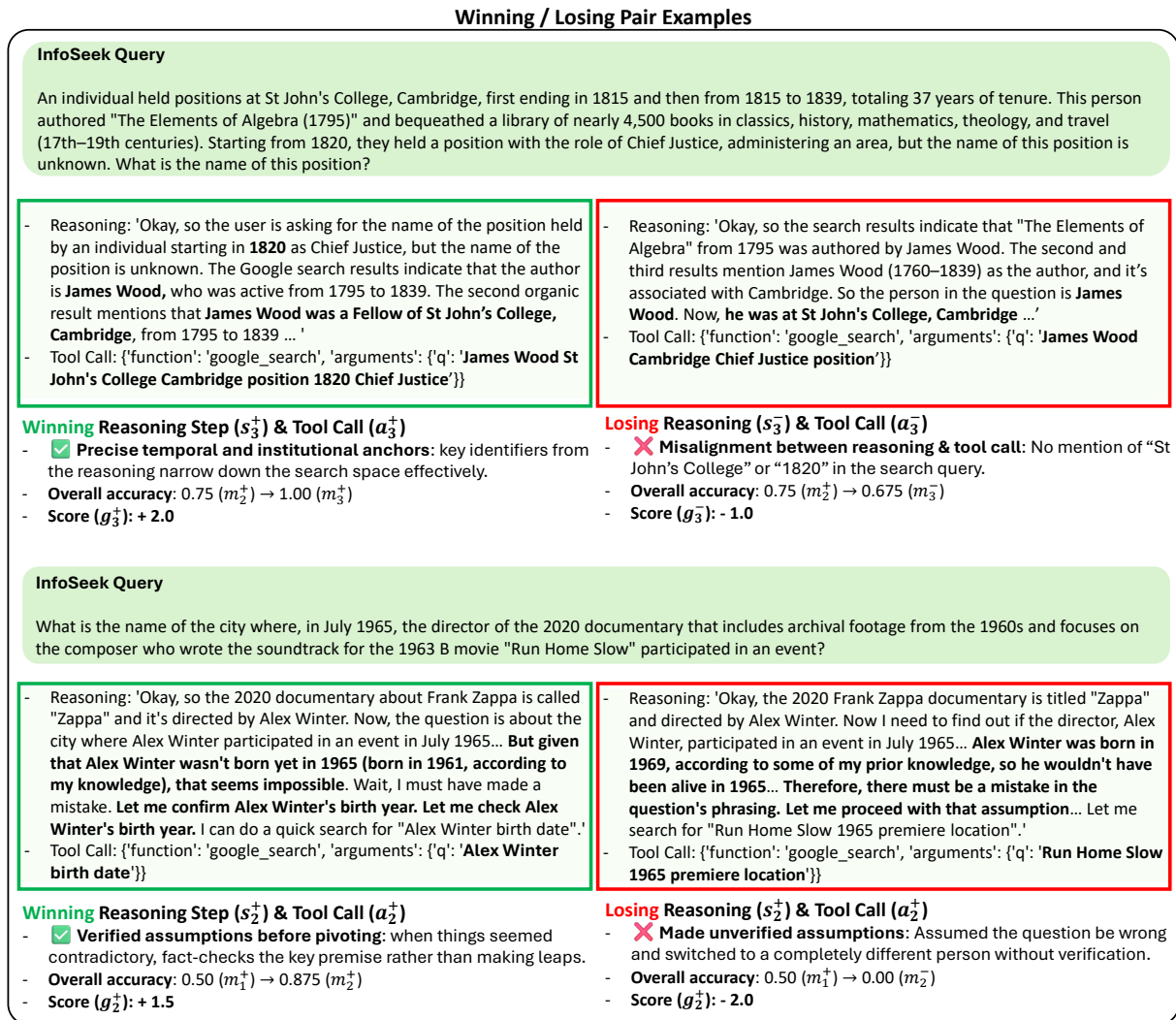


Figure 8: Examples of preference pairs constructed by our annotation pipeline.

#### Prompt Template for PRM as Scorer

##### Instruction:

You are given a user information seeking problem. Your task is to act as an impartial judge and evaluate how well the "Current Reasoning Step" contributes to solving the user's problem based on the problem description and historical reasoning context. It is OK that the current step does not contain any tool call response.

##### REASONING EVALUATION RULES:

- As you evaluate, develop and refine your assessment criteria based on the specific requirements of this problem type and reasoning context. Think carefully about how to assess the quality of the current reasoning step. Your thinking should include your evaluation criteria, explaining how the step aligns with or deviates from your expectations.
- Finally, assign the reasoning step a score from -4 to 4, using either an integer or a decimal with up to 0.1 precision. A higher score should indicate a higher-quality response.

##### [Input]:

# Information Seeking Problem  
{problem}

# Historical Reasoning Trace Summary  
{historical\_summary}

# Previous Tool Response  
{previous\_tool\_response}

# Current Reasoning Step  
{current\_reasoning}

##### [Output format]:

1. Criteria Development: [Identify the key evaluation criteria relevant for evaluating this reasoning step. Consider factors such as: logical validity and coherence of the step, tool call appropriateness and argument quality (whether too general or too narrow), consistency with user problem, historical reasoning trace summary, and previous tool response, informative/progress toward final answer, confidence and uncertainty expression, etc. Briefly explain why your selected criteria are critical for this particular evaluation.]
2. Analysis: [Always provide a step-by-step analysis here. First, briefly state the goal of the current reasoning step. Second, systematically evaluate the step against each of your identified criteria above. For each criterion, assess how well the step performs and explain your reasoning. If errors or deficiencies are found, clearly explain what is wrong and why. If the step performs well on a criterion, explain why it succeeds.]
3. Final Judgment: [Provide the final judgment within `\boxed{score}`. Examples: `\boxed{-3.0}` or `\boxed{3.5}`.]

Figure 9: Input prompt for PRInTS when the model is trained with GRPO for scoring ability and acts as a scorer at test-time.

#### Prompt Template for PRM as Summarizer

##### Instruction:

You are a reasoning trace summarizer for multi-step information seeking problems. Your task is to incrementally build a concise summary of an information-seeking process. Your summary should capture the process's state of knowledge, uncertainty, hypothesis, and next actions.

##### Input Sources:

- # Information Seeking Problem - the original user question.
- # Historical Reasoning Trace Summary - the accumulated summary built from all previous reasoning steps and tool responses.
- # Previous Tool Response - the tool response from the immediately preceding step (not yet incorporated into Historical Summary).
- # Current Reasoning Step - the reasoning and tool interaction from the current step (not the complete reasoning trace).

##### SUMMARIZATION RULES:

- Keep essential information from the Previous Tool Response and Current Reasoning Step needed for the next action.
- Incorporate what the current process believes, suspects, verifies, or is planning further verification.
- For the Current Reasoning Step's action, summarize tool name and key parameters.
- Preserve the Historical Reasoning Trace Summary unless explicitly contradicted or superseded. Justify any removals.
- Do NOT infer or invent missing information. It is normal for reasoning to be incomplete.
- Output the COMPLETE updated summary.

##### [Input]:

# Information Seeking Problem  
{problem}

# Historical Reasoning Trace Summary  
{historical\_summary}

# Previous Tool Response  
{previous\_tool\_response}

# Current Reasoning Step  
{current\_reasoning}

##### [Output format]:

1. Analysis: [In 3-5 sentences, explain what key information from Previous Tool Response and Current Reasoning Step is being added, what (if anything) from Historical Summary is being removed or updated and why, and why the resulting summary is sufficient for next steps.]
2. Updated Summary: [Provide the complete summary within `\boxed_summary{{}}` containing:
  - **Confirmed Knowledge**: Verified facts.
  - **Uncertainty**: What remains unknown.
  - **Previous Hypotheses**: Abandoned hypotheses (if relevant).
  - **Previous Action**: Previous tool calls with key parameters in the Historical Reasoning Trace Summary.
  - **Current Hypothesis**: Current working hypothesis in the Current Reasoning Step and Historical Reasoning Trace Summary.
  - **Current Action**: Most recent tool call with key parameters in the Current Reasoning Step.]

Figure 10: Input prompt for PRInTS when the model is trained with SFT for summarization ability and acts as a summarizer at test-time.

## Step-level Score Evaluation

### InfoSeek Query

Who composed the song that was performed by a rooster and a hamster in separate animated videos at separate tempos with different lyrics? Answer using the format First name Last name.

#### Best Agent Trajectory Step

- Reasoning: 'Okay, let's try to figure out this question. First, I need to recall if there's any well-known song that fits this description. The mention of a rooster and a hamster makes me think of children's songs or maybe something from a cartoon. **But I'm not immediately recalling a specific song.** Maybe I should break it down. **The key elements here are: composed a song, performed by a rooster and a hamster, separate animated videos, different tempos and lyrics...** Since I'm stuck, maybe I should use the available tools to search for the answer. The user has provided a Google search function. Let me try that. I'll use the google\_search function with the query: "song performed by rooster and hamster animated videos different tempos composer". Let's see what comes up...'
- Tool Call: {'function': 'google\_search', 'arguments': {'q': 'song performed by rooster and hamster animated videos different tempos composer'}}



#### PRInTS score evaluation

...

1. **Criteria Development**: ...

2. **Analysis**:

- **Logical Validity**: The assistant correctly identifies the problem's key elements (rooster/hamster, animated videos, different tempos/lyrics) and considers plausible angles (children's songs, educational videos, viral content). This demonstrates a structured approach.
- **Tool Call Appropriateness**: The Google search query is specific and tailored to the problem, making it a strong tool call. The query includes all critical details (e.g., "rooster and hamster," "different tempos," "composer").
- **Progress Toward Answer**: The step is progressively moving toward the answer by combining analysis with targeted research. The assistant avoids vague assumptions and instead focuses on actionable steps.
- **Confidence in Uncertainty**: The assistant acknowledges uncertainty (e.g., "I'm not immediately recalling a specific song") and uses the tool to resolve it, which is appropriate.

3. **Final Judgment**: The assistant's reasoning is well-structured, logically valid, and appropriately uses a tool call to address the problem. The step balances analytical thinking with practical action.  
**\boxed\_score{2.0}**

#### Worst Agent Trajectory Step

- Reasoning: 'Okay, let's tackle this question step by step. ...I recall that "The Yellow Song" by The Yellow Album Leaf is a song where a rooster and a hamster perform the song in different animated versions. The rooster's version is faster, and the hamster's is slower. The lyrics are also slightly different between the two. The composer of this song is likely the same as the artist, The Yellow Album Leaf. But wait, The Yellow Album Leaf is a project by Ben Schneider. **Let me confirm that. A quick search in my mind: yes, Ben Schneider is the creator and composer behind The Yellow Album Leaf.** The song in question is part of their "Yellow Album" series, which includes various characters and their own versions of songs. So, the answer should be Ben Schneider. Let me make sure there's no other composer associated with that specific song. Since The Yellow Album Leaf is his project, and he composes the songs, it's safe to attribute the composition to him. Therefore, the first name is Ben and the last name is Schneider
- Tool Call: {'function': 'submit', 'arguments': {'answer': 'Ben Schneider'}}



#### PRInTS score evaluation

...

1. **Criteria Development**: ...

2. **Analysis**:

- **Logical Validity**: The assistant assumes the song is "The Yellow Song" by "The Yellow Album Leaf" without confirming its existence or relevance to the described scenario. This leap in logic is unsupported by evidence.
- **Evidence-Based Conclusion**: The assistant relies on unverified memory (e.g., "Ben Schneider is the creator of The Yellow Album Leaf") without citing any sources or references. This undermines the reliability of the answer.
- **Relevance to Problem**: The reasoning focuses on a specific project but fails to address the core question directly. The connection between the described scenario and the song/composer is unclear.
- **Confidence in Answer**: The assistant presents the answer with certainty, despite the lack of evidence. This overconfidence is problematic in an information-seeking context.

3. **Final Judgment**: The reasoning step is speculative and lacks evidence, making it unsuitable for solving the problem. The conclusion is not justified by the provided information.  
**\boxed\_score{-2.0}**

Figure 11: PRInTS step-level evaluation examples on a GAIA query. Among four candidate steps, we show the highest-scoring (top) and lowest-scoring (bottom) steps. The high-quality step acknowledges uncertainty and initiates an appropriate tool call to gather missing information, while the low-quality step makes unverified assumptions and confidently produces an unsupported answer without evidence.