Large Language Models Have Intrinsic Meta-Cognition, but Need a Good Lens

Ziyang Ma[1*], Qingyue Yuan[2*], Zhenglin Wang[1], Deyu Zhou[1†]
[1] School of Computer Science and Engineering, Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, China
[2] Department of Neurosurgery, Shanghai Tenth People's Hospital, School of Clinical Medicine of Nanjing Medical University, China
{mazy, zhenglin, d.zhou}@seu.edu.cn yuanqy007@stu.njmu.edu.cn

Abstract

Previous research has primarily focused on the cognitive error detection capabilities of Large Language Models (LLMs), often prompting them to analyze mistakes in reasoning chains. However, few studies have examined the meta-cognitive abilities of LLMs (e.g., their self-awareness of step errors), which are crucial for their reliability. While studies on LLM self-evaluation present some measures, such as perplexity, which can reflect the answer correctness and be viewed as the lens of meta-cognition, they lack step-level analysis and adaptation. This paper studies the evaluation of LLM meta-cognition using the current lenses and how to improve these lenses. Specifically, we propose AutoMeco, an Automated Meta-cognition Evaluation framework for benchmarking the existing lenses. Furthermore, a training-free Markovian Intrinsic Reward Adjustment strategy, MIRA, is proposed to boost current meta-cognition lenses. Experimental results on three mathematical reasoning datasets and three LLMs show the reasonableness of AutoMeco by comparing it with Best-of-N verification. Moreover, the meta-cognition ability of LLMs can be better evaluated using MIRA.[1]

1 Introduction

The reasoning ability of Large Language Models (LLMs) has improved tremendously with the emergence of Large Reasoning Models (LRMs) such as OpenAI-o1 (Jaech et al., 2024) and DeepSeek-R1 (Guo et al., 2025). While evaluation on the cognitive capability of LLMs, such as reasoning outcome accuracy (Lightman et al., 2023; Zeng et al., 2025), presents the strength of LLMs, meta-cognition of these models that points to their consciousness of behavioral correctness is also important, especially for their reliability (Zhou et al., 2024; Griot et al., 2025; Yan et al., 2025).

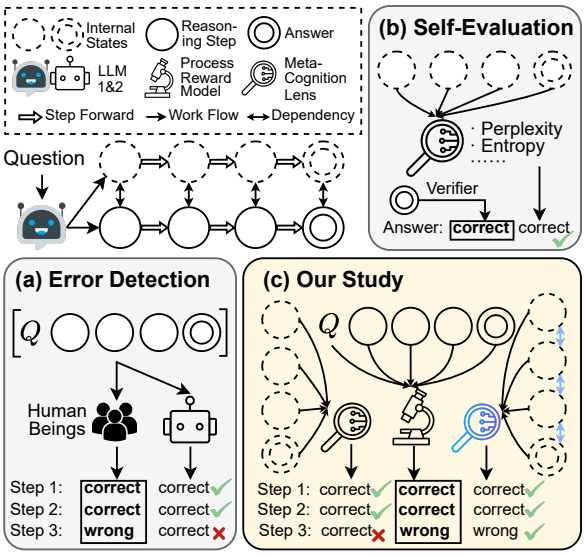


Figure 1: In reasoning tasks, error detection (a) focuses on LLMs' cognitive ability to analyze errors in reasoning steps. Self-evaluation (b) utilizes measures such as entropy as lenses to reflect self-awareness of answer rightness. Our work (c) studies the evaluation and improvement of the current lenses in reflecting LLM meta-cognition. **Bold** "correct" and "wrong" within boxes are ground truths of the answer or step correctness.

In cognitive science, meta-cognition is the cognition beyond cognition, with subjective confidence of cognitive behaviors being the main indicator (Matthews et al., 2018; Shea and Frith, 2019). Feeling of Rightness (Thompson et al., 2011) and Feeling of Error (FoE) (Gangemi et al., 2015) are two principal types of meta-cognition in reasoning tasks of human beings. As shown in Figure 1(a), previous research studies whether LLMs can detect error steps in responses generated by other models (e.g., Zeng et al., 2024; Tyen et al., 2024), focusing on the cognitive error detection ability of LLMs. Besides, research on self-evaluation in Figure 1(b) proves that internal states of LLMs can

*These authors contributed equally to this work.
†Corresponding Author.
[1]The code can be accessed via https://github.com/Yann-Ma/AutoMeco.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 3461–3478
November 4-9, 2025

reflect the answer correctness through a trained linear classifier (Zhang et al., 2025a) or training-free measures such as perplexity (Wang et al., 2025a), which can be viewed as the meta-cognition lenses. However, few of them concentrate on *whether LLMs intrinsically have meta-cognition such as FoE during the reasoning process*, which underlies the trustworthiness and self-improvement feasibility of LLMs. Therefore, this paper studies the following two problems of LLM meta-cognition: (i) *To what extent can LLM meta-cognition be observed through their internal states?* (ii) *How to observe LLM meta-cognition based on their internal states without external sources more accurately?*

Evaluating and enhancing meta-cognition lenses for LLMs confronts two challenges rooted in the dataset and methodology. (1) **Data incompleteness** regarding the internal states of LLMs. To the best of our knowledge, existing error detection benchmarks contain no internal states of LLMs such as hidden states and probabilities (Zeng et al., 2025, 2024; Xia et al., 2025). This absence makes existing benchmarks unusable for the evaluation of meta-cognition lenses. (2) **Insufficient granularity** of existing lenses. While existing approaches mainly assess answer correctness (Si et al., 2022; Huang et al., 2023; Wang et al., 2025a), they ignore the sequential dependencies between steps and probably fail to provide stepwise signals.

To address these challenges, as depicted in Figure 1(c): (1) We propose an **Auto**mated **Me**ta-**co**gnition Evaluation framework (**AutoMeco**) that realizes human-annotation-free benchmarking of LLM meta-cognition lenses. Our framework utilizes the Process Reward Model (PRM) as an annotator of step correctness. Furthermore, with the automatically annotated labels, the framework tests the lenses, such as perplexity, towards their step-rightness classification ability. (2) We propose a training-free **M**arkovian **I**ntrinsic **R**eward **A**djustment strategy (**MIRA**) that modifies the step rightness scores of the lenses based on Markov Decision Process (MDP) modeling and Q-value estimation. We model the step-level scoring as a Markov decision process, with dependencies between the reasoning steps. In the MDP, our adjustment strategy utilizes Q-value estimation to transmit the influences in reverse from the end.

The contributions of this paper are as follows:

- We propose a human-annotation-free Automated Meta-cognition Evaluation framework, AutoMeco, benchmarking the meta-cognition lenses towards step rightness prediction.
- We propose a training-free Markovian Intrinsic Reward Adjustment strategy, MIRA, which enhances the LLM meta-cognition lenses by introducing stepwise dependencies.
- We conduct experiments on mathematical reasoning datasets with different difficulties (GSM8K, MATH500, and MinervaMATH), presenting the reasonableness of AutoMeco and the effectiveness of MIRA.

2 Related Work

Reasoning Step Error Detection Benchmarks PRM800K (Lightman et al., 2023) classifies the intermediate reasoning step as positive, negative, or neutral, and underscores the importance of the reasoning step supervision to solve MATH problems. This urges the improvement of Process Reward Models, which are trained on datasets with annotations on step rightness, such as PRM800K, to output the step correctness probability.

Afterwards, MR-GSM8K (Zeng et al., 2025) and MR-Math (Xia et al., 2025) extend the step error detection beyond the classification task by manually annotating the reason behind the first error step based on subsets of GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al.) datasets, respectively. They propose the cruciality of the error reason interpretation ability of LLMs. Furthermore, BIG-Bench Mistake (BBM) (Tyen et al., 2024) and MR-Ben (Zeng et al., 2024) extend the task to other domains instead of math problems only. BBM contains 2,186 instances from symbolic reasoning tasks (Tyen et al., 2024). MR-Ben consists of 5,975 instances covering natural sciences, coding, and logic (Zeng et al., 2024). However, these benchmarks focus on the error detection ability of LLMs, which is a cognitive task. Instead, this paper aims to conduct meta-cognition evaluation on LLMs, delving into their intrinsic awareness of making mistakes during reasoning.

Process Reward Models In the multi-step mathematical reasoning task, existing research mainly defines process reward models as classifiers that provide step correctness probability as fine-grained process supervision (Uesato et al., 2022; Lightman et al., 2023; Wang et al., 2024; Zhang et al., 2025b; Shao et al., 2024). Lightman et al. (2023) constructs PRM800k, a large-scale manually annotated

process reward dataset, to train the PRM. To mitigate the dependence on costly human annotation, researchers propose automatic process supervision by annotating the step correctness using Monte Carlo Tree Search and training the process reward model on the automatically labeled datasets (Wang et al., 2024; Luo et al., 2024; Li et al., 2025). Besides, Li and Li (2025) proposes the Process Q-Value Model (PQM) that considers the sequential dependencies between steps during the training of PRMs. Benefiting from the development of PRM, this paper utilizes it as a judge to annotate step correctness. Meanwhile, motivated by PQM, we propose a training-free reward adjustment strategy to enhance meta-cognition lenses.

LLM Self-Evaluation Self-evaluation in LLMs covers uncertainty estimation of LLMs. We focus on training-free measures of self-evaluation in this paper. Traditional methods include token probability (Jiang et al., 2021), perplexity (Si et al., 2022), and entropy (Huang et al., 2023). Furthermore, Wang et al. (2025a) proposes Chain-of-Embedding (CoE), which models the layer-by-layer changes of hidden states to reflect the answer correctness of LLMs and outperforms existing methods. However, existing self-evaluation measures mainly predict the answer correctness, and most methods ignore the connections among sentences or steps. Therefore, this paper concentrates on step correctness, regarding self-evaluation measures as the lenses of LLM meta-cognition, and endows their calculated intrinsic rewards with sequential dependencies.

Besides, sampling consistency is also proven effective in uncertainty quantification. Manakul et al. (2023) calculates the similarity of multiple responses towards one question as the uncertainty score. Tonolini et al. (2024) considers the response consistency towards multiple semantically equivalent queries. However, we focus on methods based on LLM internal states in a single sample to study the intrinsic meta-cognition of LLMs.

3 Methodology

In this section, we first formally define the LLM meta-cognition observation task in §3.1. Next, we present the Automated Meta-cognition Evaluation framework in §3.2, which evaluates LLM meta-cognition lenses with the Process Reward Model as a judge. Finally, we introduce the Markovian Intrinsic Reward Adjustment strategy in §3.3 to enhance existing meta-cognition lenses.

3.1 Task Definition

The LLM meta-cognition observation task is defined as scoring the correctness of reasoning steps based on the internal states of LLMs. Given a question Q, an LLM generates a sequence containing a chain of reasoning steps $\{R_i\}_{i=1}^{N}$ and an answer A, where N is the number of reasoning steps. The LLM internal states of the reasoning step R_i include hidden states $\boldsymbol{H}_i$, logits $\boldsymbol{Z}_i$, and probabilities $\boldsymbol{P}_i$ of the output tokens. The hidden states represent all-layer hidden states of each generated token, as shown in Equation 1.

$$\boldsymbol{H}_i = \{\boldsymbol{h}_t\}_{t=1}^{T_i}, \quad \boldsymbol{h}_t = [\boldsymbol{h}_t^0, ..., \boldsymbol{h}_t^L], \tag{1}$$

where T_i and L denote the sequence length of R_i and the number of LLM hidden layers, and $\boldsymbol{h}_t^0$ represents the embedding layer output of the t-th token. With the hidden states, the logits are the unnormalized output scores generated by the final hidden layer of the LLM in Equation 2.

$$\boldsymbol{Z}_i = \{\boldsymbol{z}_t\}_{t=1}^{T_i}, \quad \boldsymbol{z}_t = \boldsymbol{h}_t^L \cdot \boldsymbol{W}_{\text{vocab}}^{\top}, \tag{2}$$

where $\boldsymbol{W}_{\text{vocab}} \in \mathbb{R}^{V \times d}$ is the vocabulary projection matrix, V is the vocabulary size, and d is the hidden dimension of the final layer. The probabilities of the output tokens are obtained by applying the softmax function to the logits:

$$\boldsymbol{P}_i = \{\boldsymbol{p}_t\}_{t=1}^{T_i}, \quad \boldsymbol{p}_t = \text{softmax}(\boldsymbol{z}_t),$$

where the probability vector $\boldsymbol{p}_t \in \mathbb{R}^V$ represents a normalized distribution over the vocabulary, guiding the final token prediction.

We formulate a self-evaluation method as a function $\mathcal{F}$ of the internal states, which calculates the confidence score s_i of the reasoning steps R_i.

$$s_i = \mathcal{F}(\boldsymbol{H}_i, \boldsymbol{Z}_i, \boldsymbol{P}_i) \tag{3}$$

3.2 AutoMated Meta-Cognition Evaluation

Our framework AutoMeco operates through four coordinated phases, as formalized in Algorithm 1. The process initiates with structured response generation, followed by intrinsic rewarding with meta-cognition lenses, automated step correctness annotation using PRM, and metrics calculation.

Structured Response Generation Given an input question Q, the language model generates a response containing N logically discrete reasoning steps $\{R_i\}_{i=1}^{N}$. Each step R_i consists of consecutive tokens $[r_1, r_2, ..., r_{T_i}]$ representing a coherent

Algorithm 1: Automated Meta-cognition Evaluation (AutoMeco)

Input: Dataset $\mathcal{D}$, Language model $\mathcal{M}$, Threshold θ

Output: Evaluation metrics

for each question $Q \in \mathcal{D}$ **do**
  /* Structured Response Generation */
  $\boldsymbol{R} \leftarrow \mathcal{M}(Q)$
  $\{R_i\}_{i=1}^{N} \leftarrow$ Segment $\boldsymbol{R}$ via boundary detection
  /* Stepwise State Aggregation */
  for each step $R_i \in \{R_1, ..., R_N\}$ **do**
    Extract hidden states $\boldsymbol{H}_i = \{\{\boldsymbol{h}_t^k\}_{k=1}^{L}\}_{t=1}^{T_i}$
    Collect logits $\boldsymbol{Z}_i$ and probabilities $\boldsymbol{P}_i$
    /* Intrinsic Rewarding */
    Compute confidence scores: $s_i^{\text{pred}} = \mathcal{F}(\boldsymbol{H}_i, \boldsymbol{Z}_i, \boldsymbol{P}_i)$
  /* Step Correctness Annotation */
  $\{s_i^{\text{true}}\}_{i=1}^{N} \leftarrow \mathcal{PRM}(Q, R_{1:N})$
  for $i = 1$ **to** N **do**
    if $s_i^{\text{true}} < \theta$ **then**
      $y_i^{\text{true}} \leftarrow 0$
    else
      $y_i^{\text{true}} \leftarrow 1$
  /* Metric Calculation */
  metrics $\leftarrow$ AUROC, AUPR, FPR95
  return metrics

reasoning unit. We employ boundary detection based on transitional phrases (e.g., "Step 1:", "Step 2:", and "Answer:") to segment the token sequence into interpretable steps.

Stepwise State Aggregation For each identified step R_i, we aggregate the internal states across its constituent tokens. The hidden states of all the layers and every token are recorded:

$$\boldsymbol{H}_i = \{\{\boldsymbol{h}_t^k\}_{k=1}^{L}\}_{t=1}^{T_i},$$

where $\boldsymbol{h}_t^k$ denotes the k-th layer's hidden state at token position t in R_i. The step-level logits $\boldsymbol{z}_t$ and probabilities $\boldsymbol{p}_t$ are extracted from every token position within R_i, resulting in $\boldsymbol{Z}_i$ and $\boldsymbol{P}_i$.

Intrinsic Rewarding The intrinsic reward s_i^{pred} for each step is computed according to the specific meta-cognition lens $\mathcal{F}$:

$$s_i^{\text{pred}} = \mathcal{F}(\boldsymbol{H}_i, \boldsymbol{Z}_i, \boldsymbol{P}_i)$$

Automated Step Correctness Annotation The PRM receives the original question Q and generated steps $\{R_i\}_{i=1}^{N}$ as input, calculating quality scores $\{s_i^{\text{true}}\}_{i=1}^{N} \in [0,1]^N$, as depicted in Equation 4. These quality scores further produce binary labels based on a threshold θ.

$$\{s_i^{\text{true}}\}_{i=1}^{N} = \mathcal{PRM}(Q, R_{1:N}) \tag{4}$$

$$y_i^{\text{true}} = \begin{cases} 0, \text{ if } s_i^{\text{true}} < \theta, \\ 1, \text{ else,} \end{cases} \quad \forall\, i \in \{1, ..., N\} \tag{5}$$

Metric Formalization Following Wang et al. (2025a), we choose area under the precision-recall curve (AUPR) (Manning and Schutze, 1999), area under the receiver operating characteristic curve (AUROC) (Boyd et al., 2013), and the false positive rate at 95% true positive rate (FPR95) (Wang et al., 2025a) to evaluate the alignment between intrinsic confidence scores and process quality labels. These metrics provide complementary views: AUROC measures global ranking consistency, FPR95 quantifies false alarm rates under high recall constraints, and AUPR evaluates precision-recall tradeoffs.

3.3 Markovian Intrinsic Reward Adjustment

The stepwise Markovian Intrinsic Reward Adjustment method models the reasoning as a Markov decision process and tunes the self-contained confidence scores by considering the sequential relations among the reasoning steps. We formalize the stepwise adjustment method in Algorithm 2 and introduce it as follows.

Given a reasoning trajectory τ defined as Equation 6 and 7, our training-free adjustment strategy formalizes the reasoning process as a deterministic MDP to adjust the step-level rewards by considering the interdependencies between steps.

$$\tau = \{(\mathcal{S}_1, R_1), ..., (\mathcal{S}_N, R_N), (\mathcal{S}_{N+1}, A)\} \tag{6}$$

$$\mathcal{S}_i = \begin{cases} Q, \text{ if } i = 1, \\ \text{concat}(Q, R_1, ..., R_{i-1}), \text{ else.} \end{cases} \tag{7}$$

State Transition Modeling The concatenation operation propagates state representations as shown in Equation 8, where $\mathcal{S}_i$ denotes the i-th reasoning state and R_{i+1} represents the corresponding reasoning action (i.e., the next reasoning step).

$$\mathcal{S}_{i+1} = \text{concat}(\mathcal{S}_i, R_i), \ \forall\, i \in \{1, ..., N\} \tag{8}$$

Algorithm 2: Markovian Intrinsic Reward Adjustment (MIRA)

Input: Reasoning trajectory $\tau = \{(\mathcal{S}_1, R_1), ..., (\mathcal{S}_N, R_N), (\mathcal{S}_{N+1}, A)\}$, confidence scores $\{s_i^{\text{pred}}\}_{i=1}^N$, discount factor γ

Output: Adjusted scores $\{\hat{s}_i^{\text{pred}}\}_{i=1}^N$

/* State Transition Modeling */

for $i \leftarrow 1$ **to** N **do**

  $\mathcal{S}_{i+1} \leftarrow \text{concat}(\mathcal{S}_i, R_i)$

/* Q-Value Backward Propagation */

Initialize $V(\mathcal{S}_{N+1}) \leftarrow 0$

for $i \leftarrow N$ **to** 1 **do**

  $\mathcal{Q}(\mathcal{S}_i, R_i) \leftarrow s_i^{\text{pred}} + \gamma \cdot V(\mathcal{S}_{i+1})$

  $V(\mathcal{S}_i) \leftarrow \max_{R_i} \mathcal{Q}(\mathcal{S}_i, R_i)$

/* Score Normalization */

for $i \leftarrow 1$ **to** N **do**

  $\hat{s}_i^{\text{pred}} \leftarrow \frac{\exp(\mathcal{Q}(\mathcal{S}_i, R_i))}{\sum_{j=1}^N \exp(\mathcal{Q}(\mathcal{S}_j, R_j))}$

return $\{\hat{s}_i^{\text{pred}}\}_{i=1}^N$

Q-Value Estimation and Backward Propagation The expected future reward $\mathcal{Q}(\mathcal{S}_i, R_i)$ integrates immediate confidence and discounted future value in Equation 9, with value function $V(\mathcal{S}_{i+1}) = \max_{R_{i+1}} \mathcal{Q}(\mathcal{S}_{i+1}, R_{i+1})$ and $\gamma \in (0, 1]$ as the discount factor. In this equation, we recursively update Q-values from terminal state $\mathcal{S}_{N+1}$ with $V(\mathcal{S}_{N+1}) = 0$. This simplifies under a deterministic MDP to direct value assignment in Equation 10.

$$\mathcal{Q}(\mathcal{S}_i, R_i) = s_i^{\text{pred}} + \gamma \cdot \mathbb{E}_{\mathcal{S}_{i+1}}[V(\mathcal{S}_{i+1})] \quad (9)$$

$$\mathcal{Q}(\mathcal{S}_i, R_i) = s_i^{\text{pred}} + \gamma \cdot V(\mathcal{S}_{i+1}) \quad (10)$$

Score Normalization Final adjusted scores are computed via softmax scaling:

$$\hat{s}_i^{\text{pred}} = \frac{\exp(\mathcal{Q}(\mathcal{S}_i, R_i))}{\sum_{j=1}^N \exp(\mathcal{Q}(\mathcal{S}_i, R_i))}$$

4 Experiment

This section will answer four logically connected questions to present our experiments. §4.2 explores whether it is statistically feasible to predict step correctness based on internal states. Furthermore, §4.3 answers two questions: (1) How do existing meta-cognition lenses perform towards the meta-cognition observation? (2) Is our proposed stepwise adjustment method effective in improving these meta-cognition lenses? Afterwards, §4.4 validates whether PRM-as-a-Judge is a reasonable method towards the meta-cognition evaluation.

4.1 Experiments Setup

Datasets We focus on mathematical reasoning task with different difficulties. We evaluate the existing meta-cognition lenses on three datasets: the grade school math problems from GSM8K (Cobbe et al., 2021), competition mathematics problems from MATH500 (Hendrycks et al.), and undergraduate- or Olympiad-level mathematical problems from MinervaMATH (Lewkowycz et al., 2022). We choose the first 250 problems of GSM8K due to its relatively low difficulty, which is also the English split of the multilingual math word problems from MGSM (Shi et al.). MATH500 contains 500 pieces selected by Lightman et al. (2023) from the MATH dataset. MinervaMATH includes 272 problems that require quantitative reasoning.

Backbone Models For PRM, we use Qwen2.5-Math-PRM-7B (Zhang et al., 2025b) in our experiments, as it is one of the best mathematical process reward models according to recent research by Zheng et al. (2024). For LLMs, we consider three open-sourced models, Qwen2.5-7B (Yang et al., 2024), Llama-3-8B-Instruct (AI@Meta, 2024), and Mistral-7B-Instruct (Jiang et al., 2023b), to conduct reasoning on the three datasets above.

Baselines We conduct six meta-cognition lenses to predict the step correctness, which results in the intrinsic rewards of each step in the reasoning chains. These baselines are: (1) CoE-C (Wang et al., 2025a); (2) CoE-R (Wang et al., 2025a); (3) ΔEntropy (Yin et al., 2024); (4) Max Probability (Maxprob); (5) Perplexity (PPL) (Huang et al., 2023); (6) Entropy (Si et al., 2022). Among them, (1) and (2) require access to hidden states of LLMs, and (3)-(6) only require probability distributions. More details are in Appendix A.

Implementation Details We set the maximal number of new tokens for Qwen2.5-7B as 768 and those for the other two models as 2048 because of their extra instruction-tuning. In the AutoMeco evaluation, we set $\text{temperature} = 1.0$ and $\text{num\_sequences} = 1$ to evaluate the model under the condition of greedy generation. In the Best-of-N (BoN) evaluation, we set $\text{temperature} = 0.8$ and $\text{N} = 6$. We select the best response from the

N samples by choosing the one with the highest intrinsic reward averaged over steps. For Mistral-7B-Instruct, we use Mistral-7B-Instruct-v0.2 (Jiang et al., 2023a). Appendix C shows the prompt templates for the three datasets. All the experiments are conducted on four 24G 3090 GPUs or one A100 80G GPU. For clear visualization, we use the kdeplot function from the seaborn libary (Waskom, 2021) with contour levels of 7, density thresholds of 0.15 for correct and 0.1 for incorrect samples, and bw_adjust $= 1.5$.

4.2 Statistical Feasibility

We prompt Qwen2.5-7B to reason on GSM8K, MATH500, and MinervaMATH. We conduct the self-evaluation measures to predict the step correctness, which results in the intrinsic rewards of each step in the reasoning chains. Subsequently, with the step rewards annotated by PRM, we calculate the correlation of the intrinsic and PRM rewards on the data split conditioned on PRM reward$\in (0, 1] \cup [0.9, 1)$. PRM denotes Qwen2.5-PRM-Math-7B in the following, if without reclarification. Besides, we show the distinguishability of step correctness by visualizing the kernel density estimation of Magnitude and Angle, two internal features proposed by Wang et al. (2025a). Appendix A presents details of the two features.

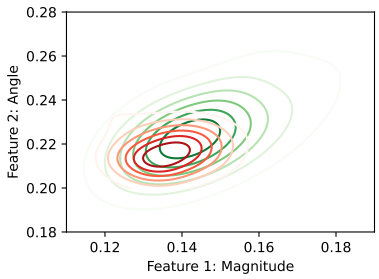
Table 1 presents that the intrinsic and PRM rewards have a significant positive correlation on the GSM8K dataset, which proves the feasibility of LLM meta-cognition observation. However, the premise is that the LLM can handle the task to some extent, as the correlation drops significantly with the rise of task difficulty. Furthermore, entropy is statistically the most promising method to capture the inherent presentation of step correctness, which consistently has the highest correlation coefficients on the three datasets. Besides, Figure 2 illustrates the feature distributions of the correct and wrong steps. It shows the decline of the step correctness predictability when the dataset gets harder, which is consistent with the statistics above.

4.3 Comparison and Ablation Study

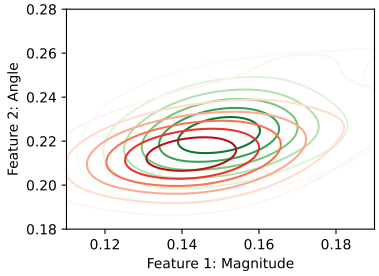
We evaluate the meta-cognition lenses in a more challenging and valuable setting with the three language models. In this setting, we test the ability of these methods to distinguish the correct and incorrect steps, including wrong and unsure ones, instead of merely the wrong steps.

Table 1: Spearman coefficient (Spearman, 1961) and Kendall's Tau (Kendall, 1938) between the intrinsic and PRM rewards of Qwen-2.5-7B on three datasets. All values are formatted as `coefficient (p-value)` with p-values being in scientific notation, retaining two significant figures (e.g., 1.7e-29 denotes 1.7×10^{-29}). *S* and *K* stands for Spearman and Kendall, respectively. Bold and underlined denote the best and second-best.

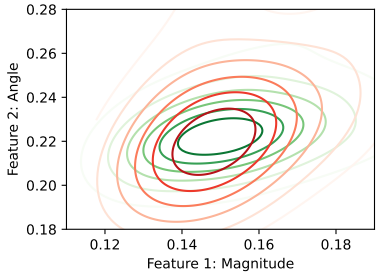
Methods	GSM8K		MATH500		MinervaMATH	
	S	*K*	*S*	*K*	*S*	*K*
CoE-C	0.040 (0.174)	0.035 (0.074)	0.068 (0.0001)	0.046 (0.0001)	0.010 (0.710)	0.007 (0.685)
CoE-R	0.325 (1.7e-29)	0.225 (4.1e-30)	0.212 (1.4e-33)	0.145 (2.2e-34)	0.255 (3.7e-21)	0.175 (1.7e-21)
Maxprob	0.498 (1.3e-72)	0.354 (7.3e-72)	0.241 (2.1e-43)	0.164 (8.1e-44)	<u>0.265</u> (9.6e-23)	<u>0.182</u> (3.1e-23)
PPL	<u>0.499</u> (6.1e-73)	<u>0.355</u> (4.9e-72)	<u>0.243</u> (4.3e-44)	<u>0.165</u> (2.6e-44)	0.263 (2.2e-22)	0.180 (9.1e-23)
Entropy	**0.521** (1.6e-80)	**0.370** (3.1e-78)	**0.246** (3.0e-45)	**0.168** (1.1e-45)	**0.270** (1.7e-23)	**0.185** (6.8e-24)
ΔEntropy	-0.060 (0.041)	-0.039 (0.052)	0.147 (7.5e-17)	0.097 (1.7e-16)	0.066 (0.017)	0.044 (0.016)



(1) GSM8K



(2) MATH500



(3) MinervaMATH

Figure 2: Intrinsic feature distributions of correct and incorrect steps of Qwen2.5-7B on GSM8K, MATH500, and MinervaMATH. Green and red contours represent features of correct and wrong steps.

Table 2: AutoMeco and Best-of-N evaluation results of meta-cognition lenses with and without our proposed reward adjustment strategy, MIRA, across three mathematics reasoning datasets and three LLMs. Green and red denote whether MIRA improves the meta-cognition lenses. "Acc" denotes accuracy. "Majority" represents majority voting. "PRM" denotes voting based on the step-averaged rewards calculated by the PRM.

Methods	Qwen2.5-7B				Llama-3-8B-Instruct				Mistral-7B-Instruct			
	Best-of-N Acc↑	AUROC↑	FPR95↓	AUPR↑	Best-of-N Acc↑	AUROC↑	FPR95↓	AUPR↑	Best-of-N Acc↑	AUROC↑	FPR95↓	AUPR↑
GSM8K												
Maxprob	53.60	61.25	90.74	96.82	73.20	65.53	90.29	94.67	42.80	68.46	86.17	51.79
+ MIRA (ours)	58.80 (+5.20)	67.50 (+6.25)	81.48 (-9.26)	96.88 (+0.06)	73.60 (+0.40)	58.21 (-7.32)	85.44 (-4.85)	92.60 (-2.07)	45.20 (+2.40)	58.88 (-9.58)	79.86 (-6.31)	39.94 (-11.85)
PPL	57.20	61.19	90.74	96.81	80.40	65.67	90.29	94.70	46.40	68.93	83.79	51.68
+ MIRA (ours)	66.40 (+9.20)	70.92 (+9.73)	79.63 (-11.11)	97.65 (+0.84)	78.80 (-1.60)	59.32 (-6.35)	84.47 (-5.82)	94.60 (-0.10)	47.60 (+1.20)	61.74 (-7.19)	79.02 (-4.77)	64.54 (+12.86)
Entropy	56.40	60.62	92.59	96.84	80.00	66.99	85.44	94.86	44.80	71.68	78.43	54.57
+ MIRA (ours)	63.20 (+6.80)	71.90 (+11.28)	75.93 (-16.66)	97.56 (+0.72)	79.20 (-0.80)	60.87 (-6.12)	86.41 (+0.97)	93.67 (-1.19)	47.20 (+2.40)	64.14 (-7.54)	77.00 (-1.43)	54.58 (+0.01)
ΔEntropy	60.00	64.02	94.44	96.82	77.20	50.45	97.09	90.18	43.60	56.86	83.67	38.15
+ MIRA (ours)	61.60 (+1.60)	56.65 (-7.57)	94.44 (-0.00)	95.69 (-1.13)	74.40 (-2.80)	45.66 (-4.79)	98.28 (+1.19)	90.19 (+0.01)	49.60 (+6.00)	55.77 (-1.09)	86.77 (+3.10)	40.23 (+2.08)
CoE-R	54.80	64.78	88.89	97.14	77.60	63.84	91.26	94.32	44.80	39.24	96.78	32.47
+ MIRA (ours)	75.60 (+20.80)	65.23 (+0.45)	88.89 (-0.00)	96.90 (-0.24)	77.60 (+0.00)	63.52 (-0.32)	84.47 (-6.79)	94.40 (+0.08)	39.60 (-5.20)	53.58 (+14.34)	93.86 (-2.92)	37.08 (+4.61)
CoE-C	58.80	69.53	90.74	97.65	70.80	72.29	79.61	95.91	44.00	52.33	93.56	39.99
+ MIRA (ours)	57.60 (-1.20)	68.87 (-0.66)	70.37 (-20.37)	97.05 (-0.60)	72.80 (+2.00)	58.24 (-14.05)	88.35 (+8.74)	92.71 (-3.20)	44.40 (+0.40)	58.56 (+6.23)	80.33 (-13.23)	39.82 (-0.17)
Majority / PRM	86.80 / 75.20	-	-	-	86.80 / 89.20	-	-	-	60.40 / 70.80	-	-	-
MATH500												
Maxprob	35.00	58.12	95.92	91.43	18.60	63.76	88.72	72.51	6.20	56.00	94.72	12.40
+ MIRA (ours)	37.60 (+2.60)	64.66 (+6.54)	86.52 (-9.40)	92.52 (+1.09)	18.00 (-0.60)	62.03 (-1.73)	87.26 (-1.46)	68.04 (-4.47)	5.60 (-0.60)	56.89 (+0.89)	91.86 (-2.86)	12.71 (+0.31)
PPL	40.00	57.83	95.61	91.43	23.20	63.84	88.04	72.71	6.00	56.35	94.07	12.60
+ MIRA (ours)	44.20 (+4.20)	64.40 (+6.57)	86.52 (-9.09)	93.88 (+2.45)	21.00 (-2.20)	62.77 (-1.07)	86.48 (-1.56)	75.74 (+3.03)	6.20 (+0.20)	54.51 (-1.84)	91.70 (-2.37)	41.29 (+28.69)
Entropy	40.20	57.53	96.65	91.38	22.40	64.04	87.71	72.63	6.20	55.43	92.97	12.41
+ MIRA (ours)	43.00 (+2.80)	66.23 (+8.70)	85.58 (-11.07)	93.19 (+1.81)	19.60 (-2.80)	65.87 (+1.83)	86.48 (-1.23)	74.98 (+2.35)	5.60 (-0.60)	53.20 (-2.23)	92.88 (-0.09)	23.52 (+11.11)
ΔEntropy	41.40	54.21	97.81	90.91	17.80	53.77	96.42	62.51	4.80	49.63	95.54	12.60
+ MIRA (ours)	37.20 (-4.20)	64.62 (+10.41)	89.03 (-8.78)	92.69 (+1.78)	18.00 (+0.20)	62.70 (+8.93)	90.61 (-5.81)	68.37 (+5.86)	7.20 (+2.40)	61.45 (+11.82)	90.76 (-4.78)	17.51 (+4.91)
CoE-R	39.00	47.82	97.81	88.17	19.80	52.37	92.07	60.95	6.00	45.37	94.99	12.99
+ MIRA (ours)	44.00 (+5.00)	60.43 (+12.61)	85.89 (-11.92)	91.64 (+2.47)	17.60 (-2.20)	57.96 (+5.59)	92.07 (-0.00)	64.62 (+3.67)	4.40 (-1.60)	50.72 (+5.35)	93.15 (-1.84)	11.27 (-1.72)
CoE-C	37.40	59.71	94.36	91.35	16.80	59.27	93.52	68.83	6.60	59.80	86.22	13.43
+ MIRA (ours)	37.40 (+0.00)	65.03 (+5.32)	84.64 (-9.28)	92.69 (+1.34)	17.80 (+1.00)	61.32 (+2.05)	89.39 (-4.13)	67.65 (-1.18)	6.00 (-0.60)	57.75 (-2.05)	90.76 (+4.54)	13.03 (-0.40)
Majority / PRM	51.80 / 49.60	-	-	-	24.60 / 31.20	-	-	-	9.20 / 11.80	-	-	-
MinervaMATH												
Maxprob	8.46	52.75	97.01	88.23	6.62	56.13	92.51	72.04	1.47	53.33	91.91	17.24
+ MIRA (ours)	9.56 (+1.10)	63.37 (+10.62)	94.61 (-2.40)	90.57 (+2.34)	6.25 (-0.37)	60.18 (+4.05)	92.90 (+0.39)	71.33 (-0.71)	1.47 (+0.00)	60.20 (+6.87)	88.14 (-3.77)	18.82 (+1.58)
PPL	9.93	52.96	97.60	88.37	7.35	57.54	91.75	72.65	1.47	53.95	91.99	17.49
+ MIRA (ours)	11.03 (+1.10)	64.50 (+11.54)	92.81 (-4.79)	92.72 (+4.35)	5.15 (-2.20)	60.81 (+3.27)	93.09 (+1.34)	79.74 (+6.91)	1.47 (+0.00)	58.44 (+4.49)	88.88 (-3.11)	46.94 (+29.45)
Entropy	10.66	52.53	98.80	88.35	8.82	57.41	90.60	72.63	1.47	53.73	91.33	17.28
+ MIRA (ours)	10.66 (+0.00)	64.60 (+12.07)	94.61 (-4.19)	91.52 (+3.17)	6.25 (-2.57)	61.11 (+3.70)	92.51 (+1.91)	74.04 (+1.41)	1.47 (+0.00)	57.50 (+3.77)	92.64 (+1.31)	26.02 (+8.74)
ΔEntropy	10.29	49.84	98.80	88.42	8.09	48.15	96.74	65.71	1.47	48.20	93.54	14.61
+ MIRA (ours)	10.66 (+0.37)	60.86 (+11.02)	92.22 (-6.58)	90.20 (+1.78)	5.51 (-2.58)	58.18 (+10.03)	93.86 (-2.88)	70.89 (+5.18)	2.21 (+0.74)	60.48 (+12.28)	89.70 (-3.84)	19.78 (+5.17)
CoE-R	5.51	51.80	98.20	88.01	5.51	54.07	89.25	69.37	3.31	46.99	97.47	17.99
+ MIRA (ours)	12.13 (+6.62)	59.47 (+7.67)	93.41 (-4.79)	90.02 (+2.01)	5.51 (+0.00)	59.84 (+5.77)	90.60 (+1.35)	73.21 (+3.84)	0.74 (-2.57)	51.49 (+4.50)	96.30 (-1.17)	16.32 (-1.67)
CoE-C	9.93	54.99	97.60	88.29	6.25	59.65	91.55	75.17	1.84	59.94	90.27	20.31
+ MIRA (ours)	9.19 (-0.74)	63.63 (+8.62)	89.22 (-8.38)	90.64 (+2.35)	6.62 (+0.37)	60.47 (+0.82)	92.51 (+0.96)	71.76 (-3.41)	1.84 (+0.00)	60.68 (+0.74)	89.21 (-1.06)	19.09 (-1.22)
Majority / PRM	10.66 / 13.97	-	-	-	6.99 / 9.93	-	-	-	3.31 / 6.99	-	-	-

Performance Trends Across Difficulty Levels As illustrated in Table 2, model performance exhibits an overall correlation with problem difficulty. The aggregate metrics of BoN accuracy, AUROC, and AUPR demonstrate monotonic degradation as task complexity increases. Concurrently, FPR95 shows a statistically significant upward trajectory. Therefore, the performance of the meta-cognition lenses in reflecting reasoning step correctness diminishes substantially on more difficult tasks.

Effectiveness and Efficiency of MIRA The proposed reward adjust strategy demonstrates robust generalization across model architectures and task difficulties. As detailed in Table 2, MIRA achieves performance improvements in 61.1% (BoN) and 68.5% (AUROC) of experimental configurations (N=54). Figure 3 illustrates the particular efficacy of MIRA in enhancing the meta-cognition lenses for Qwen2.5-7B. Although MIRA works on less than half of the settings for Llama-3-8B-Instruct, Table 4.3 presents that it enhances more meta-cognition lenses on the difficult task, MinervaMATH, than the simple one, GSM8K. Besides, MIRA also performs well on a wide range of meta-cognition lenses for Mistral-7B-Instruct. The gains brought by MIRA across different architectures prove its cross-model compatibility. These experimental results confirm our hypothesis that stepwise adjustment of intrinsic rewards can effectively compensate for inherent calibration weaknesses in the existing lenses of LLM meta-cognition. Although we primarily focus on open-source models, these results are still insightful for closed-source black-box LLMs due to the potential of probability-based meta-cognition lenses such as Maxprob and PPL. Additionally, Table 3 demonstrates that MIRA adds only a negligible amount of latency per instance.

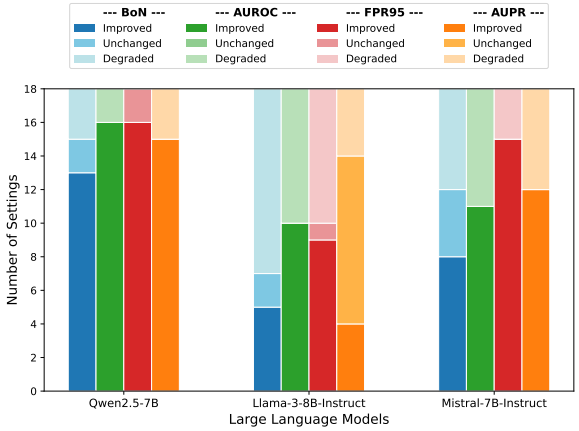


Figure 3: Frequency of MIRA leading to improved, unchanged, and degraded performance for meta-cognition lenses on three LLMs (Qwen2.5-7B, Llama-3-8B-Instruct, Mistral-7B-Instruct) and three datasets (GSM8K, MATH500, MinervaMATH).

The increase does not exceed 0.03 milliseconds, which validates the efficiency of MIRA.

Table 3: Latency measurements (unit: millisecond) for the meta-cognition lenses with (w/) and without (w/o) MIRA on Qwen2.5-7B (GSM8K subset). "Per Step" and "Per Instance" denote the latency averaged across all steps and all instances, respectively.

Methods	Per Step	Per Instance	
		w/o MIRA	w/ MIRA
Maxprob	0.12	0.64	0.65
PPL	0.04	0.20	0.21
Entropy	512.94	2669.63	2669.64
ΔEntropy	508.54	2646.73	2646.74
CoE-R	0.51	2.64	2.66
CoE-C	0.64	3.31	3.34

Validation on Difficult Math Tasks Practical validation on the MinervaMATH dataset reveals compelling advantages of meta-cognition lenses over conventional ensemble approaches. For instance, Qwen2.5-7B with adjusted CoE-R configuration outperforms majority voting baselines by 1.47% in BoN accuracy (12.13% vs 10.66%), while Llama-3-8B-Instruct achieves better performance through self-evaluation-based selection than majority voting (PPL: +0.36%, Entropy: +1.83%, Δentropy: +1.10%). This performance gap suggests that properly calibrated self-evaluation metrics enable more effective identification of high-quality reasoning paths than static aggregation methods. The findings align with our hypothesis that LLMs contain latent self-diagnostic capabilities that can be operationalized through appropriate metric design. Meanwhile, these results also indicate the potential of meta-cognition evaluation as a mechanism for autonomous LLM self-improvement through intrinsic process supervision.

4.4 PRM-as-a-Judge Analysis

To validate the reasonableness of utilizing PRM as an annotator, we conduct the following analysis: (1) Analyze the consistency between the rewards of two different PRMs, Qwen2.5-Math-PRM-7B and Skywork-o1-Open-PRM-Qwen-2.5-7B (He et al., 2024); (2) Compare AutoMeco with BoN by calculating the consistency of their benchmarking results of six meta-cognition lenses. The consistency metrics include top-K and last-K match rate, and consistency rate (CR) (Wang et al., 2025b). Appendix B presents more details on these metrics.

Agreement between Different PRMs Table 4 shows strong inter-annotator agreement between the two PRMs, confirming their reliability for evaluating step-level errors. Notably, the kappa score for annotations on Mistral-7B-Instruct's reasoning traces reaches 0.7334. However, the PRMs exhibit lower agreement on step-level annotations than on instance-level ones. For example, the step-level Kappa for the reasoning steps of Qwen2.5-7B is 0.3274, compared to 0.5878 at the instance level. This discrepancy indicates that while the PRMs are effective judges overall, there is room to improve their fine-grained annotation accuracy.

Consistency between AutoMeco and BoN Our analysis demonstrates that PRM-as-a-Judge achieves reasonable consistency with BoN evaluation, validating its utility as a complementary method for efficient method ranking. While perfect alignment is not observed, the results highlight meaningful agreement trends. As shown in Table 5, AutoMeco exhibits alignment with BoN rankings for top-tier methods, particularly at K=3. BoN's top methods include its top three in 66.67% settings on average, with Mistral-7B-Instruct achieving 100% consistency. Furthermore, the average CR of 48.15% supports the reliability of AutoMeco to benchmark the meta-cognition lenses. Moreover, it achieves the highest alignment on Llama-3-8B-Instruct (CR=53.33%), suggesting model-specific optimization potential.

Table 4: Inter-annotator agreement for step- and instance-level rewards between Qwen2.5-Math-PRM-7B and Skywork-o1-Open-PRM-Qwen-2.5-7B on the responses of the three LLMs to the three math datasets. "Spearman" and "Pearson" are the Spearman (Spearman, 1961) and Pearson's correlation coefficients (Pearson, 1895). All the correlations in this table are statistically significant ($p < 0.05$). "Cohen" represents the inter-annotator agreement score, Cohen's Kappa (McHugh, 2012). The reported Kappa score is maximized by finding the optimal binarization thresholds through a grid search over the range $\{0.1, 0.2, \cdots, 0.8, 0.9\}$.

Model	Spearman↑	Pearson↑	Cohen↑
Step Level			
Qwen2.5-7B	0.6309	0.3552	0.3274
Llama-3-8B-Instruct	0.5850	0.3473	0.2657
Mistral-7B-Instruct	0.5758	0.6109	0.5349
Instance Level (Averaged on Steps)			
Qwen2.5-7B	0.7630	0.6236	0.5878
Llama-3-8B-Instruct	0.7179	0.6617	0.6108
Mistral-7B-Instruct	0.7250	0.7981	0.7334

Table 5: Consistency metrics of Best-of-N (BoN) and AutoMeco results across three large language models. Top-k and Last-K Match evaluate whether the best/worst method chosen by AutoMeco is in the top/last K methods ranked by BoN. Top-K Order considers both the best and the worst. CR stands for consistency rate.

Model	Top-K Match		Last-K Match		Top-K Order		CR
	K=1	K=3	K=1	K=3	K=1	K=3	
Qwen2.5	0.00	66.67	0.00	66.67	0.00	16.67	44.44
Llama-3	0.00	33.33	0.00	33.33	0.00	0.00	53.33
Mistral	33.33	100.00	0.00	33.33	0.00	16.67	46.67
Average	11.11	66.67	0.00	44.44	0.00	11.11	48.15

5 Conclusion

We investigate the LLM meta-cognition observation capability of self-evaluation measures as meta-cognition lenses for language models, through an automated benchmarking framework to evaluate these lenses and a fine-grained self-evaluation adjustment strategy to enhance them. These meta-cognition lenses can capture the LLM meta-cognition, and the stepwise modification further improves their observation ability.

Our study points to several directions with a considerable range of research for future work, including constructing an LLM meta-cognition benchmark with manually annotated step correctness and the LLM internal states for errorless step labels, developing more accurate self-evaluation measures for meta-cognition observation, applying meta-cognition signals to realize LLM self-improvement, and aligning LLM with human preferences more efficiently via meta-cognition loss. Moreover, utilizing the meta-cognition lenses for response refinement in other scenarios, such as agentic tasks, constitutes a crucial direction.

Limitations

Our work focuses on introducing meta-cognition into language model evaluation by incorporating internal model states into self-assessment mechanisms. While our approach demonstrates promising results, several limitations warrant discussion:

Model Accessibility Requirements Our method incorporates internal model representations such as hidden states to enable fine-grained self-evaluation. This design, however, requires access to model internals and is mainly limited to open-source architectures. While logits- and probability-based self-evaluation methods are compatible with both open-source and mainstream closed-source LLMs with access to the last-layer states, more delicate approaches using hidden states cannot be directly applied to closed-source models such as GPT-4 (Achiam et al., 2023). Exploring approximations or hybrid strategies may help bridge the gap between white-box and black-box interpretability.

Computational and Memory Overhead Our Markovian Intrinsic Reward Adjustment (MIRA) strategy utilizes internal model signals to dynamically refine self-evaluation. As a result, it introduces additional computation during the reasoning phase and requires extra memory to store intermediate hidden states. Though the overhead remains moderate in scale, these factors may influence efficiency in practical deployment settings.

Extension to Large Reasoning Models Our framework has been validated across multiple model families, including Qwen, Llama, and Mistral. However, generalizing it to more sophisticated Large Reasoning Models (LRMs) involves non-trivial considerations. The dynamic and complex nature of their reasoning processes calls for more adaptive step-wise modeling techniques. Integrating these with recent advances in cognitive behavior analysis of RL (Yue et al., 2025; Gandhi et al., 2025) or reasoning boundary analysis (Chen et al., 2024) may offer a promising path forward.

Acknowledgments

The authors would like to thank all of the anonymous reviewers for their valuable comments. This research is funded by the National Natural Science Foundation of China (Grant No.62176053). This work is supported by the Big Data Computing Center of Southeast University.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

AI@Meta. 2024. Llama 3 model card.

Kendrick Boyd, Kevin H Eng, and C David Page. 2013. Area under the precision-recall curve: point estimates and confidence intervals. In Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13, pages 451–466. Springer.

Qiguang Chen, Libo Qin, Jiaqi WANG, Jingxuan Zhou, and Wanxiang Che. 2024. Unlocking the capabilities of thought: A reasoning boundary framework to quantify and optimize chain-of-thought. In The Thirty-eighth Annual Conference on Neural Information Processing Systems.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D Goodman. 2025. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. arXiv preprint arXiv:2503.01307.

Amelia Gangemi, Sacha Bourgeois-Gironde, and Francesco Mancini. 2015. Feelings of error in reasoning—in search of a phenomenon. Thinking & Reasoning, 21(4):383–396.

Maxime Griot, Coralie Hemptinne, Jean Vanderdonckt, and Demet Yuksel. 2025. Large language models lack essential metacognition for reliable medical reasoning. Nature communications, 16(1):642.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948.

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. 2024. Skywork-o1 open series. https://huggingface.co/Skywork.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2).

Yuheng Huang, Jiayang Song, Zhijie Wang, Shengming Zhao, Huaming Chen, Felix Juefei-Xu, and Lei Ma. 2023. Look before you leap: An exploratory study of uncertainty measurement for large language models. arXiv preprint arXiv:2307.10236.

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

Albert Jiang, Alexandre Sablayrolles, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Louis Ternon, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Théophile Gervet, and 4 others. 2023a. Mistral-7b-v0.2.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023b. Mistral 7b. Preprint, arXiv:2310.06825.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. Transactions of the Association for Computational Linguistics, 9:962–977.

Maurice G Kendall. 1938. A new measure of rank correlation. Biometrika, 30(1-2):81–93.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. Advances in Neural Information Processing Systems, 35:3843–3857.

Shuangtao Li, Shuaihao Dong, Kexin Luan, Xinhan Di, and Chaofan Ding. 2025. Enhancing reasoning through process supervision with monte carlo tree search. arXiv preprint arXiv:2501.01478.

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

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

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

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. arXiv preprint arXiv:2303.08896.

Christopher Manning and Hinrich Schutze. 1999. Foundations of statistical natural language processing. MIT press.

Julian Matthews, Pia Schröder, Lisandro Kaunitz, Jeroen JA Van Boxtel, and Naotsugu Tsuchiya. 2018. Conscious access in the near absence of attention: critical extensions on the dual-task paradigm. Philosophical Transactions of the Royal Society B: Biological Sciences, 373(1755):20170352.

Mary L McHugh. 2012. Interrater reliability: the kappa statistic. Biochemia medica, 22(3):276–282.

Karl Pearson. 1895. Vii. note on regression and inheritance in the case of two parents. proceedings of the royal society of London, 58(347-352):240–242.

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

Nicholas Shea and Chris D Frith. 2019. The global workspace needs metacognition. Trends in cognitive sciences, 23(7):560–571.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, and 1 others. Language models are multilingual chain-of-thought reasoners. In The Eleventh International Conference on Learning Representations.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. 2022. Prompting gpt-3 to be reliable. arXiv preprint arXiv:2210.09150.

Charles Spearman. 1961. The proof and measurement of association between two things.

Valerie A Thompson, Jamie A Prowse Turner, and Gordon Pennycook. 2011. Intuition, reason, and metacognition. Cognitive psychology, 63(3):107–140.

Francesco Tonolini, Nikolaos Aletras, Jordan Massiah, and Gabriella Kazai. 2024. Bayesian prompt ensembles: Model uncertainty estimation for black-box large language models. In Findings of the Association for Computational Linguistics ACL 2024, pages 12229–12272.

Gladys Tyen, Hassan Mansoor, Victor Cărbune, Yuanzhu Peter Chen, and Tony Mak. 2024. Llms cannot find reasoning errors, but can correct them given the error location. In Findings of the Association for Computational Linguistics ACL 2024, pages 13894–13908.

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

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9426–9439.

Yiming Wang, Pei Zhang, Baosong Yang, Derek F. Wong, and Rui Wang. 2025a. Latent space chain-of-embedding enables output-free LLM self-evaluation. In The Thirteenth International Conference on Learning Representations.

Zhaoyang Wang, Weilei He, Zhiyuan Liang, Xuchao Zhang, Chetan Bansal, Ying Wei, Weitong Zhang, and Huaxiu Yao. 2025b. CREAM: Consistency regularized self-rewarding language models. In The Thirteenth International Conference on Learning Representations.

Michael L Waskom. 2021. Seaborn: statistical data visualization. Journal of Open Source Software, 6(60):3021.

Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. 2025. Evaluating mathematical reasoning beyond accuracy. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 39, pages 27723–27730.

Hanqi Yan, Linhai Zhang, Jiazheng Li, Zhenyi Shen, and Yulan He. 2025. Position: Llms need a bayesian meta-reasoning framework for more robust and generalizable reasoning. In 2025 International Conference on Machine Learning: ICML25.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian

Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024. Qwen2 technical report. arXiv preprint arXiv:2407.10671.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Zhiyuan Zeng, Xiaonan Li, Junqi Dai, Qinyuan Cheng, Xuanjing Huang, and Xipeng Qiu. 2024. Reasoning in flux: Enhancing large language models reasoning through uncertainty-aware adaptive guidance. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2401–2416, Bangkok, Thailand. Association for Computational Linguistics.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. 2025. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? arXiv preprint arXiv:2504.13837.

Zhongshen Zeng, Pengguang Chen, Shu Liu, Haiyun Jiang, and Jiaya Jia. 2025. MR-GSM8k: A meta-reasoning benchmark for large language model evaluation. In The Thirteenth International Conference on Learning Representations.

Zhongshen Zeng, Yinhong Liu, Yingjia Wan, Jingyao Li, Pengguang Chen, Jianbo Dai, Yuxuan Yao, Rongwu Xu, Zehan Qi, Wanru Zhao, Linling Shen, Jianqiao Lu, Haochen Tan, Yukang Chen, Hao Zhang, Zhan Shi, Bailin Wang, Zhijiang Guo, and Jiaya Jia. 2024. MR-ben: A meta-reasoning benchmark for evaluating system-2 thinking in LLMs. In The Thirty-eighth Annual Conference on Neural Information Processing Systems.

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. 2025a. Reasoning models know when they're right: Probing hidden states for self-verification. arXiv preprint arXiv:2504.05419.

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

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

Yujia Zhou, Zheng Liu, Jiajie Jin, Jian-Yun Nie, and Zhicheng Dou. 2024. Metacognitive retrieval-augmented large language models. In Proceedings of the ACM Web Conference 2024, pages 1453–1463.

A LLM Self-Evaluation Measures

A.1 Chain-of-Embedding (CoE)

For a reasoning step R_i with T_i tokens, Wang et al. (2025a) quantifies two features, Magnitude and Angle, to represent the layer-by-layer changes of query understanding based on hidden states:

$$\text{Mag}(\boldsymbol{H}_i) = \frac{1}{L}\sum_{\ell=0}^{L-1}\frac{||\boldsymbol{h}_{\ell+1}-\boldsymbol{h}_\ell||_2}{||\boldsymbol{h}_L-\boldsymbol{h}_0||_2}$$

$$\text{Ang}(\boldsymbol{H}_i) = \frac{1}{L}\sum_{\ell=0}^{L-1}\frac{\arccos\left(\frac{\boldsymbol{h}_{\ell+1}^\intercal\cdot\boldsymbol{h}_\ell}{||\boldsymbol{h}_{\ell+1}||_2\cdot||\boldsymbol{h}_\ell||_2}\right)}{\arccos\left(\frac{\boldsymbol{h}_L^\intercal\cdot\boldsymbol{h}_0}{||\boldsymbol{h}_L||_2\cdot||\boldsymbol{h}_0||_2}\right)}$$

$$\boldsymbol{h}^\ell = \frac{1}{T_i}\sum_{t=1}^{T_i}\boldsymbol{h}_t^\ell,\ \forall\,\ell\in[0,...,L]$$

Two basic components in Magnitude and Angle, magnitude change $M(\boldsymbol{h}_\ell,\boldsymbol{h}_{\ell+1})$ and angle change $A(\boldsymbol{h}_\ell,\boldsymbol{h}_{\ell+1})$, are defined as follows:

$$M(\boldsymbol{h}_\ell,\boldsymbol{h}_{\ell+1}) = ||\boldsymbol{h}_{\ell+1}-\boldsymbol{h}_\ell||_2$$

$$A(\boldsymbol{h}_\ell,\boldsymbol{h}_{\ell+1}) = \arccos\left(\frac{\boldsymbol{h}_{\ell+1}^\intercal\ \cdot\ \boldsymbol{h}_\ell}{||\boldsymbol{h}_{\ell+1}||_2\ \cdot\ ||\boldsymbol{h}_\ell||_2}\right)$$

As shown in Equation 11 and 12, CoE-R and CoE-C capture the correctness by combining the magnitude and angle changes in the real and complex spaces, respectively (Wang et al., 2025a).

A.2 Entropy

Entropy reflects the uncertainty of reasoning steps based on their token probabilities (Si et al., 2022):

$$\text{Entropy}(\boldsymbol{P}_i) = \frac{1}{T_i}\sum_{t=1}^{T_i}(-\boldsymbol{p}_t^\intercal\cdot\log\boldsymbol{p}_t) \quad (13)$$

We utilize its reciprocal as the correctness score.

A.3 ΔEntropy

Yin et al. (2024) proposes that the abnormal uncertainty fluctuation is useful for judging wrong steps during reasoning, which is formally the uncertainty change between two adjacent reasoning steps. The more the LLM's uncertainty fluctuates, the more likely the LLM makes mistakes. We choose Entropy as the uncertainty metric to formulate the entropy fluctuation:

$$\Delta\text{Entropy}(\boldsymbol{P}_i) = \begin{cases}0, \text{ if } i=0,\\ \text{Entropy}(\boldsymbol{P}_i)-\text{Entropy}(\boldsymbol{P}_{i-1}), \text{ else.}\end{cases} \quad (14)$$

We utilize the opposite number of ΔEntropy as the step correctness score.

A.4 Max Probability (Maxprob)

Maxprob calculates the average of the maximal elements in the probability distributions, which assumes that the top-1 probability reflects the certainty of LLMs:

$$\text{Maxprob}(\boldsymbol{P}_i) = \frac{1}{T_i}\sum_{t=1}^{T_i}\max(\boldsymbol{p}_t) \quad (15)$$

A.5 Perplexity (PPL)

PPL reflects the uncertainty of LLMs by considering the negative logarithm of the maximal probability:

$$\text{PPL}(\boldsymbol{P}_i) = -\frac{1}{T_i}\sum_{t=1}^{T_i}\log\ \max(\boldsymbol{p}_t) \quad (16)$$

We utilize its reciprocal as the correctness score.

B Consistency Metrics

For an LLM Θ and a dataset $\mathcal{D}$, AutoMeco and BoN evaluate M meta-cognition lenses, which results in two ranks denoted by $\boldsymbol{\alpha}=\alpha_1,\alpha_2,...,\alpha_M$ and $\boldsymbol{\beta}=\beta_1,\beta_2,...,\beta_M$. Top-K Match, Last-K Match, and Top-K Order are measured as follows:

$$\text{TopMatch}_{\Theta,\mathcal{D}}(K) = \mathbb{I}[\text{argmin}\ \alpha_m\in\text{argsort}(\boldsymbol{\beta})[:K]]$$
$$\text{LastMatch}_{\Theta,\mathcal{D}}(K) = \mathbb{I}[\text{argmax}\ \alpha_m\in\text{argsort}(\boldsymbol{\beta})[-K:]]$$
$$\text{TopOrder}_{\Theta,\mathcal{D}}(K) = \text{TopMatch}_{\Theta,\mathcal{D}}(K)\cdot\text{LastMatch}_{\Theta,\mathcal{D}}(K)$$

where $K\in[1,2,3]$, $\mathbb{I}[\cdot]$ is the indicator function and argsort($\boldsymbol{\beta}$) outputs the indices that sort $\boldsymbol{\beta}$ in ascending order. Equation 17 calculates the consistency rate of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ (Wang et al., 2025b).

$$\text{CR}_{\Theta,\mathcal{D}} = \frac{2}{M(M-1)}\sum_{1\le m<m'\le M}[\mathbb{I}[(\alpha_m-\alpha_{m'})(\beta_m-\beta_{m'})>0]- \mathbb{I}[(\alpha_m-\alpha_{m'})(\beta_m-\beta_{m'})<0]] \quad (17)$$

Ultimately, we average these metrics on three datasets to present the consistency of AutoMeco and BoN for the LLM Θ.

$$\text{CoE-R}(\boldsymbol{H}_i) = \text{Mag}(\boldsymbol{H}_i) - \text{Ang}(\boldsymbol{H}_i) = \frac{1}{L}\sum_{\ell=0}^{L-1}\left(\frac{M(\boldsymbol{h}_\ell, \boldsymbol{h}_{\ell+1})}{M(\boldsymbol{h}_0, \boldsymbol{h}_L)} - \frac{A(\boldsymbol{h}_\ell, \boldsymbol{h}_{\ell+1})}{A(\boldsymbol{h}_0, \boldsymbol{h}_L)}\right) \tag{11}$$

$$\begin{aligned}\text{CoE-C}(\boldsymbol{H}_i) &= \frac{1}{L}\left|\sum_{\ell=0}^{L-1} M(\boldsymbol{h}_\ell, \boldsymbol{h}_{\ell+1}) \cdot e^{i \cdot A(\boldsymbol{h}_\ell, \boldsymbol{h}_{\ell+1})}\right| \\ &= \frac{1}{L}\left|\sum_{\ell=0}^{L-1} [M(\boldsymbol{h}_\ell, \boldsymbol{h}_{\ell+1})\cos(A(\boldsymbol{h}_\ell, \boldsymbol{h}_{\ell+1})) + i \cdot M(\boldsymbol{h}_\ell, \boldsymbol{h}_{\ell+1})\sin(A(\boldsymbol{h}_\ell, \boldsymbol{h}_{\ell+1}))]\right| \\ &= \frac{1}{L}\sqrt{\left(\sum_{\ell=0}^{L-1} M(\boldsymbol{h}_\ell, \boldsymbol{h}_{\ell+1})\cos(A(\boldsymbol{h}_\ell, \boldsymbol{h}_{\ell+1}))\right)^2 + \left(\sum_{\ell=0}^{L-1} M(\boldsymbol{h}_\ell, \boldsymbol{h}_{\ell+1})\sin(A(\boldsymbol{h}_\ell, \boldsymbol{h}_{\ell+1}))\right)^2}\end{aligned} \tag{12}$$

C Prompt Templates

• GSM8K

> Solve this math problem step by step. Give the reasoning steps using 'Step n:' before each step to distinguish between different steps, where n is a positive integer starting from 1, representing the current step number. Then give the final answer on the last line by itself in the format of `"Answer:"`
>
> Do not add anything other than the integer answer after `"Answer:"`
>
> **Question:** `{input_data}`

• MATH500 and MinervaMATH

> **Question:** `{input_data}`
>
> Please reason step by step. Use 'Step n:' before each step to distinguish between different steps, where n is a positive integer starting from 1, representing the current step number. Then, give your final answer on the last line in the format of `"Answer: \boxed{}"`

D Statistical Feasibility on Other LLMs

As shown in Table 6 and 7, meta-cognition lenses perform weakly in capturing meta-cognition of Llama-3-8B-Instruct and Mistral-7B-Instruct. Interestingly, however, these methods are statistically better at hard tasks than simple ones. For instance, CoE-C has a stronger correlation with PRM rewards on MinervaMATH than GSM8K and MATH500 for the two LLMs.

E Statistics of Reasoning Steps

Table 8: Statistics on correct, wrong, and uncertain reasoning steps annotated by Qwen2.5-Math-PRM-7B in mathematical reasoning tasks.

Model	Dataset	Step Correctness		
		Correct	Wrong	Uncertain
Qwen2.5	GSM8K	1127	17	122
	MATH500	3079	106	471
	MinervaMATH	1278	47	277
Llama-3	GSM8K	1130	34	229
	MATH500	2900	401	1135
	MinervaMATH	1328	113	637
Mistral	GSM8K	523	223	828
	MATH500	3079	106	471
	MinervaMATH	371	1130	2322

F Case Study

Figure 4, 5, and 6 illustrate three cases of Qwen2.5-7B on GSM8K, MATH500, and MinervaMATH datasets. In these cases, self-evaluation measures fail to predict the rightness of the wrong step, while MIRA successfully calibrates the intrinsic meta-cognition by conducting stepwise adjustments to these meta-cognition lenses.

Table 6: Spearman coefficient (Spearman, 1961) and Kendall's Tau (Kendall, 1938) between the intrinsic and PRM rewards of Llama-3-8B-Instruct on three datasets.

Methods	GSM8K		MATH500		MinervaMATH	
	Spearman	*Kendall*	*Spearman*	*Kendall*	*Spearman*	*Kendall*
CoE-C	0.179 (8.0e-10)	0.119 (1.1e-9)	0.095 (4.0e-8)	0.054 (3.1e-6)	0.203 (8.2e-15)	0.135 (2.0e-14)
CoE-R	**0.219** (3.8e-14)	**0.139** (1.4e-12)	**0.123** (1.2e-12)	**0.083** (9.2e-13)	0.155 (3.8e-9)	0.103 (4.5e-9)
Maxprob	0.102 (0.0005)	0.081 (3.6e-5)	0.098 (1.5e-8)	0.063 (6.9e-8)	0.022 (0.4058)	0.013 (0.4569)
PPL	0.100 (0.0006)	0.080 (4.2e-5)	0.101 (6.1e-9)	0.064 (3.4e-8)	**0.263** (2.2e-22)	**0.180** (9.1e-23)
Entropy	0.119 (5.0e-5)	0.092 (2.3e-6)	0.104 (2.0e-9)	0.065 (2.6e-8)	0.044 (0.098)	0.027 (0.127)
ΔEntropy	-0.208 (8.0e-13)	-0.136 (3.4e-12)	0.024 (0.167)	0.019 (0.094)	-0.074 (0.0047)	-0.049 (0.00499)

Table 7: Spearman coefficient (Spearman, 1961) and Kendall's Tau (Kendall, 1938) between the intrinsic and PRM rewards of Mistral-7B-Instruct on three datasets.

Methods	GSM8K		MATH500		MinervaMATH	
	Spearman	*Kendall*	*Spearman*	*Kendall*	*Spearman*	*Kendall*
CoE-C	0.194 (9.0e-08)	0.130 (1.1e-07)	**0.215** (3.9e-17)	**0.139** (5.9e-16)	**0.333** (2.5e-18)	**0.221** (2.7e-17)
CoE-R	-0.282 (3.9e-15)	-0.190 (9.1e-15)	-0.118 (4.8e-06)	-0.078 (5.8e-06)	-0.101 (0.0099)	-0.064 (0.0148)
Maxprob	0.306 (1.1e-17)	0.203 (1.1e-16)	0.036 (0.158)	0.023 (0.185)	0.011 (0.783)	0.005 (0.848)
PPL	0.320 (3.1e-19)	0.213 (3.7e-18)	0.051 (0.048)	0.032 (0.061)	0.033 (0.396)	0.020 (0.437)
Entropy	**0.326** (6.8e-20)	**0.217** (7.2e-19)	0.035 (0.181)	0.023 (0.191)	0.012 (0.754)	0.007 (0.794)
ΔEntropy	0.154 (2.4e-05)	0.105 (2.1e-05)	0.009 (0.713)	0.006 (0.721)	-0.040 (0.303)	-0.027 (0.301)

Question: John drives for 3 hours at a speed of 60 mph and then turns around because he realizes he forgot something very important at home. He tries to get home in 4 hours but spends the first 2 hours in standstill traffic. He spends the next half-hour driving at a speed of 30mph, before being able to drive the remaining time of the 4 hours going at 80 mph. How far is he from home at the end of those 4 hours?	**PRM**		**PPL**		**MIRA+PPL**	
	Reward	Label	Reward	Prediction (thres=1.800)	Reward	Prediction (thres=0.987)
Step 1: Calculate the distance John travels in the first 3 hours. Distance = Speed * Time Distance = 60 mph * 3 hours = 180 miles	0.99	1	5.43	1	1.00	1
Step 2: Calculate the distance John travels in the next 2 hours in standstill traffic. Since he is not moving, the distance traveled is 0 miles.	0.99	1	4.49	1	1.00	1
Step 3: Calculate the distance John travels in the next 0.5 hours at a speed of 30 mph. Distance = Speed * Time Distance = 30 mph * 0.5 hours = 15 miles	0.99	1	26.85	1	1.00	1
Step 4: Calculate the remaining time John has to travel after the first 3.5 hours. Total time = 4 hours Time spent in standstill traffic = 2 hours Time spent driving at 30 mph = 0.5 hours Remaining time = 4 hours - 2 hours - 0.5 hours = 1.5 hours	0.99	1	6.19	1	1.00	1
Step 5: Calculate the distance John travels in the remaining 1.5 hours at a speed of 80 mph. Distance = Speed * Time Distance = 80 mph * 1.5 hours = 120 miles	0.99	1	213.45	1	1.00	1
Step 6: Calculate the total distance John travels in 4 hours. Total distance = Distance in first 3 hours + Distance in standstill traffic + Distance at 30 mph + Distance at 80 mph Total distance = 180 miles + 0 miles + 15 miles + 120 miles = 315 miles	0.99	1	14.04	1	1.00	1
Step 7: Calculate the distance John is from home at the end of the 4 hours. Since he traveled 315 miles away from home, he is 315 miles from home.	0.34	0	4.22	1 ✗	0.985	0 ✓

Answer: 315 miles **Correct Answer: 45**

Figure 4: A demonstration of MIRA enhancing meta-cognition observation for Qwen2.5-7B on GSM8K. The thresholds are determined by maximizing the F1 score of step correctness prediction.

Question: Solve for x: $3^{2x} + 19 = 10^x$	PRM		Maxprob		MIRA +Maxprob		Entropy		MIRA +Entropy	
	Reward	Label	Reward	Prediction (thres=0.59)	Reward	Prediction (thres=0.68)	Reward	Prediction (thres=0.47)	Reward	Prediction (thres=0.76)
Step 1: To solve the equation $3^{2x} + 19 = 10^x$, we can start by isolating the exponential terms on one side of the equation.	0.98	1	0.84	1	1.000	1	1.24	1	1.00	1
Step 2: Subtract 19 from both sides of the equation to get $3^{2x} = 10^x - 19$.	0.99	1	0.95	1	1.000	1	3.88	1	1.00	1
Step 3: Now, we can take the natural logarithm (ln) of both sides of the equation to bring down the exponents.	0.99	1	0.83	1	1.000	1	1.37	1	1.00	1
Step 4: Applying the natural logarithm, we get $ln(3^{2x}) = ln(10^x - 19)$.	0.99	1	0.92	1	1.000	1	3.15	1	1.00	1
Step 5: Step 5: Using the logarithm property $ln(a^b) = b\, ln(a)$, we can rewrite the equation as $2x \cdot ln\ 3 = ln(10^x - 19)$.	0.99	1	0.93	1	0.99	1	3.85	1	1.00	1
Step 6: Now, we can divide both sides of the equation by $ln\ 3$ to isolate the x term: $2x = ln(10^x - 19)/ln\ 3$.	0.99	1	0.90	1	0.97	1	2.42	1	1.00	1
Step 7: To solve for x, we can divide both sides of the equation by 2: $x = (ln(10^x - 19)/ln(3))/2$.	0.99	1	0.94	1	0.93	1	3.53	1	1.00	1
Step 8: Unfortunately, this equation cannot be solved algebraically for an exact value of x. However, we can use numerical methods or graphing to approximate the solution.	0.99	1	0.83	1	0.83	1	1.26	1	0.90	1
Step 9: Using a numerical solver or graphing calculator, we find that the approximate solution for x is 1.113.	0.31	0	0.75	1 ✗	0.67	0 ✓	0.99	1 ✗	0.73	0 ✓

Answer: 1.113 **Correct Answer:** $\frac{1}{2}$

Figure 5: A demonstration of MIRA enhancing meta-cognition observation for Qwen2.5-7B on MATH500. The thresholds are determined by maximizing the F1 score of step correctness prediction.

Question: In Cambridge, shoppers can buy apples from two sources: a local orchard, and a store that ships apples from out of state. The orchard can produce up to 50 apples per day at a constant marginal cost of 25 cents per apple. The store can supply any remaining apples demanded, at a constant marginal cost of 75 cents per unit. When apples cost 75 cents per apple, the residents of Cambridge buy 150 apples in a day. Assume that the city of Cambridge sets the price of apples within its borders. What price should it set, in cents?	**PRM**		**COE-C**		**MIRA +COE-C**	
	Reward	Label	Reward	Prediction (thres=0.10)	Reward	Prediction (thres=0.53)
Step 1: Determine the total demand for apples in Cambridge. The total demand for apples in Cambridge is 150 apples per day.	0.99	1	0.12	1	0.68	1
Step 2: Determine the supply from the local orchard. The local orchard can produce up to 50 apples per day at a marginal cost of 25 cents per apple.	0.99	1	0.13	1	0.65	1
Step 3: Determine the remaining demand after the local orchard's supply. The remaining demand after the local orchard's supply is 150 - 50 = 100 apples per day.	0.99	1	0.13	1	0.59	1
Step 4: Determine the supply from the store for the remaining demand. The store can supply any remaining apples demanded at a marginal cost of 75 cents per apple.	0.99	1	0.12	1	0.56	1
Step 5: Calculate the total cost for the local orchard and the store to supply the apples. The total cost for the local orchard is 50 apples * 25 cents = 1250 cents. The total cost for the store to supply the remaining 100 apples is 100 apples * 75 cents = 7500 cents. The total cost for both the local orchard and the store is 1250 cents + 7500 cents = 8750 cents.	0.99	1	0.12	1	1.00	1
Step 6: Calculate the price per apple that should be set by the city of Cambridge. To minimize the total cost, the city should set the price per apple equal to the marginal cost of the store, which is 75 cents per apple.	0.45	0	0.12	1 ✗	0.529	0 ✓

Answer: $\boxed{75}$ **Correct Answer: 75**

Figure 6: A demonstration of MIRA enhancing meta-cognition observation for Qwen2.5-7B on MinervaMATH. The thresholds are determined by maximizing the F1 score of step correctness prediction.