CURE: Advancing Reasoning via Self-Consistent Reward in Test-Time Experience

Anonymous ACL submission

Abstract

Test-Time Scaling (TTS) has emerged as an effective paradigm for improving the reasoning performance of Large Language Models by allocating additional computation during inference. Existing TTS frameworks frequently utilize process reward models to improve performance, yet the substantial computational cost of training PRMs remains a major limitation. To address this limitation, we propose **Context-Aware Unlabeled Reward Reasoning** (CURE), a novel TTS framework designed for both intensive reasoning and knowledge-intensive tasks. Given an input question, CURE first retrieves the most relevant questions from the test set. Conditioned on retrieved questions, LLMs then perform **Context-Reward Reasoning** to generate candidate answers to the original question. The final answer is obtained via majority voting over these candidate answers. Since the retrieved questions lack ground-truth labels, we sample multiple predictions and get pseudo-labels via majority voting, which are then utilized to generate reward messages. CURE is evaluated on competitive reasoning and knowledge-intensive tasks, where it demonstrates state-of-the-art potential. For example, CURE markedly improves Qwen2.5-7B by 25.29% on average. Crucially, CURE-augmented smaller models exhibit competitive superiority over massive baselines, with Qwen2.5-7B exceeding the performance of Qwen2.5-72B by 2.08 points. Extensive ablation studies and analyses further validate the effectiveness and robustness of our approach. Our code is available at this URL.

1 Introduction

Large Language Models (LLMs) (Achiam et al., 2023; Guo et al., 2025; Jaech et al., 2024) have demonstrated remarkable advancements across a wide range of domains. Recently, OpenAI's o1 (Jaech et al., 2024) and Deepseek-R1 (Guo et al., 2025) have shown that Test-Time Scaling (TTS) (Zhang et al., 2025a; Balachandran et al., 2025) can significantly enhance the reasoning capabilities of LLMs by leveraging additional computational resources during inference. These reasoning models have reached or approached human-level proficiency in tasks such as mathematics and code generation (Guo et al., 2025; Team et al., 2025; Li et al., 2025).

TTS methods can be broadly categorized into two paradigms: training-based and inference-based (Liu et al., 2025). Training-based methods enhance a model's reasoning ability by leveraging long Chain-of-Thought (CoT) data. For example, the cold-start phase of DeepSeek-R1 (Guo et al., 2025) fine-tunes base models on long CoT data, thereby improving multi-step reasoning. In contrast, inference-based methods improve reasoning performance by allocating additional computation during inference. Representative techniques include Process Reward Models (PRMs), which guide step-by-step reasoning (Yuan et al., 2024; Zhang et al., 2025c); Self-Consistency methods that apply majority voting over multiple sampled reasoning paths (Stiennon et al., 2020); and search-based approaches such as Monte Carlo Tree Search (MCTS) (Zhang et al., 2024). Despite their promise, both paradigms face notable challenges. Training-based approaches are often hindered by the high cost of acquiring quality long-form CoT data and the substantial computational resources required for model optimization. Inference-based methods often rely on special PRMs to verify intermediate reasoning steps (Jiang et al., 2025; Zhang et al., 2025c), which introduces additional training overhead and limits their generality.

To mitigate these challenges, we introduce Context-Aware Unlabeled Reward Reasoning (CURE), a novel Test-Time Scaling framework that bolsters model performance without the need for fine-tuning and the guidance of PRMs. CURE functions via a structured three-stage pipeline. First,

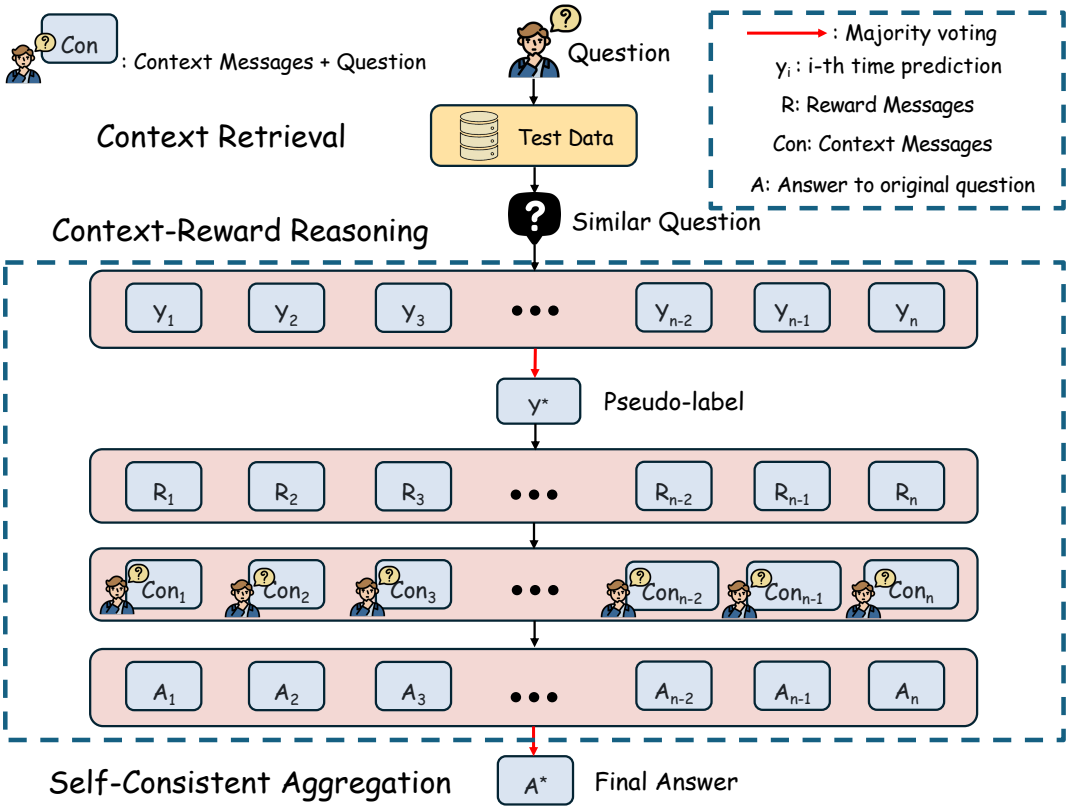

Figure 1: Overview of the CURE framework, illustrating its three-stage pipeline: Context Retrieval, Context-Reward Reasoning, and Self-Consistent Aggregation.

given an input question, we retrieve the most similar questions from the test set. The retrieved questions exhibit strong topical coherence and provides informative context that supports accurate inference. Then, in the **Context-Reward Reasoning** stage, each retrieved question is used to guide the model through a **Prediction–Reward–Reasoning** process, forming a layered reasoning architecture. In the Prediction process, the model generates multiple candidate predictions. Since retrieved questions are typically unlabeled, pseudo-labels are derived via majority voting across the sampled predictions to establish a consensus. Then, the pseudo-labels serve as reference answers during the Reward process, evaluating candidate responses and generating their corresponding reward messages. Subsequently, in the Reasoning process, the model synthesizes all contextual messages, including the predictions for the retrieved question and their corresponding reward messages, to reason the original question. Finally, in the Self-Consistent Aggregation stage, all candidate answers for the original query are compiled, and the final answer is determined via majority voting.

In our experiments, we evaluate CURE across a diverse suite of models, including two instruction-tuned models and two Large Reasoning Models (LRMs). We evaluate CURE on 6 reasoning benchmarks and 3 knowledge-intensive benchmarks to assess the its versatility. Notably, applying CURE to LLaMA3.1-Instruct results in an average improvement of **23.5%** across all benchmarks, with a remarkable **117.4%** increase on the challenging AIME2024 task. Furthermore, Qwen2.5-7B-Instruct enhanced by CURE outperforms the significantly larger Qwen2.5-72B by an average of 2.08 points, suggesting that CURE can effectively bridge the performance gap between disparate model scales. For large reasoning models, CURE facilitates a 33.57-point improvement for DeepSeek-R1-Qwen3-8B on the AMC benchmark. Our analysis further identifies two critical factors for CURE's efficacy: the sampling number and the design of reward messages. Finally, we compare CURE with several commonly used TTS methods. In conclusion, the main contributions of our paper are as follows:

- Inspired by In-Context Learning, we propose Context-Aware Unlabeled Reward Reasoning (CURE), a novel Test-Time Scaling frame-

work .

- We evaluate CURE on multiple LLMs across six mathematical reasoning and three knowledge-intensive benchmarks, demonstrating consistent and significant improvements over the base models.
- We further investigate the mechanism of CURE by varying the number of sampled responses and reward messages, providing deeper insights into its effectiveness.

2 Related Work

2.1 In-Context Learning

In-Context Learning (ICL) (Brown et al., 2020) is a method where LLMs perform new tasks by using examples or instructions provided directly within the input prompt, without requiring fine-tuning or additional training. By embedding task demonstrations in natural language, ICL allows models to identify patterns from just a few examples, leveraging the extensive semantic knowledge acquired during pre-training. Unlike supervised learning (Stiennon et al., 2020), which depends on backpropagation (Rojas and Rojas, 1996) and large labeled datasets, ICL leverages the latent space of pre-trained models to produce accurate predictions without modifying their parameters or retaining task-specific knowledge after inference. Prompt engineering (White et al., 2023) plays a key role by designing prompts that guide the model's reasoning and improve its outputs, enabling LLMs to tackle complex tasks by drawing analogies from limited examples, thereby offering a flexible and efficient alternative to traditional machine learning approaches.

2.2 Test-Time Scaling

Test-Time Scaling (Zhang et al., 2025a; Balachandran et al., 2025) enhances the reasoning capabilities of LLMs during inference by leveraging additional computational resources without altering model weights. A foundational technique is CoT (Wei et al., 2022), which encourages models to "think step by step" (Lightman et al., 2023), significantly improving performance on complex tasks. More structured approaches include Best-of-N (BoN) sampling (Brown et al., 2024), beam search (Snell et al., 2024), and Monte Carlo Tree Search (Zhang et al., 2024). These methods generate multiple candidate solutions, often applying majority voting (Stiennon et al., 2020), PRM (Yuan et al., 2024) as verifier, or LLM-as-a-judge (Zheng et al., 2023) for greater accuracy.

3 Methodology

Our framework, illustrated in Figure 1, consists of three stages: (1) Context Retrieval: Given an input question, we retrieve the most semantically similar questions from the test set to provide topic-coherent contextual support (Section 3.1). (2) Context-Reward Reasoning: The retrieved question is leveraged to guide a structured Prediction–Reward–Reasoning process (Section 3.2). (3) Self-Consistent Aggregation: The reasoning outputs from all contexts are aggregated, and the final answer is determined by majority voting (Section 3.3).

3.1 Stage One: Context Retrieval

When encountering unseen questions, it is essential to equip the model with relevant domain knowledge and similar prior cases. While many datasets show strong correlations between their training and test sets, some evaluation instances appear exclusively in the test set. Consequently, retrieving similar examples exclusively from the training set is not always feasible. To address this limitation, we introduce a Context Retrieval stage that identifies the most similar cases directly from the test data. To ensure that the retrieved examples closely resemble the target examples, we employ an embedding model to vectorize the questions. We then compute the cosine similarity between the current question and all other questions in the test set, retrieving the most similar questions. This retrieved questions are subsequently used as the starting questions in the Context-Reward Reasoning stage.

3.2 Stage Two: Context-Reward Reasoning

Some studies (Dai et al., 2022) have demonstrated a duality between ICL and fine-tuning, providing theoretical support for the effectiveness of ICL, they overlook a crucial aspect: fine-tuning requires models to first generate predictions and then compute gradients based on these predictions and the corresponding labels. To address this limitation, we propose the Context-Reward Reasoning stage, which allows the LLM to learn from both predictions and reward messages, as illustrated in Figure 2.

Inspired by Self-consistency with CoT (Wang et al., 2022), which shows that correct answers

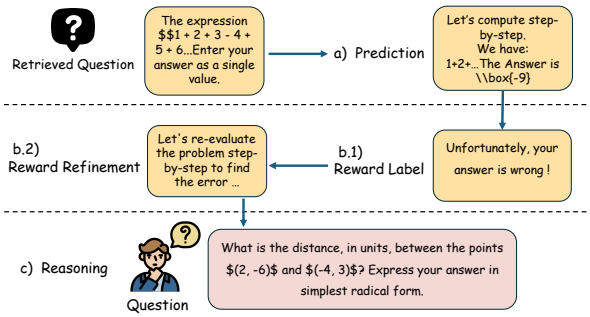

Figure 2: Context-Reward Reasoning pipeline at test time.

tend to form dense and consistent clusters among multiple model outputs, we generate multiple predictions for the retrieved questions and then apply majority voting to get the pseudo-labels. In the reward process, this pseudo-label is used to evaluate each prediction and generates reward messages based on its correctness. Then the LLM learns from the prediction and reward messages, enabling it to reason about the question more comprehensively. We provide a detailed description of the overall process below.

Prediction Given the retrieved questions from the context retrieval stage, Context-Reward reasoning first enables the LLM to generate the multiple predictions, in the format of zero-shot CoT reasoning (Kojima et al., 2022).

For each retrieved question $\hat{x}$, the LLM input α_i is formatted as:

$$\alpha_i = \text{Q: } \hat{x}. \text{ A: [Z]}, \tag{1}$$

where i represents the i-th time sample, and [Z] represents zero shot trigger (Guo et al., 2025). More details about the triggers used for different benchmarks are described in Appendix A. Subsequently, based on the input α_i, LLM is instructed to generate the prediction for the retrieved question, obtaining the y_i which can be formulated as:

$$y_i = \text{LLM}(\alpha_i). \tag{2}$$

Furthermore, recognizing that some LLMs may not fully comply with the instructions when answering a query (e.g., by refusing to answer), CURE incorporates an additional filtering step to exclude abnormal responses that deviate from the given instructions.

Reward During the Reward process, we use majority voting to get a pseudo-label y^*, *i.e.*,

$$y^* = \arg\max_{c \in \mathcal{C}} \sum_{i=1}^{i} \mathbf{1}\{y_i = c\}. \tag{3}$$

Then the prediction y_i is evaluated against the corresponding the pseudo-label y^*, which can be given as:

$$R_i = \begin{cases} R_{\text{correct}}, & \text{if } y^* = y_i, \\ R_{\text{wrong}}, & \text{otherwise.} \end{cases} \tag{4}$$

We associate each prediction with a reward message comprising two distinct components. The first, the Reward Label, provides a binary reward of correctness (e.g., "Well done! Your answer is correct."). The second, the Reward Refinement, is a response conditioned on the reward label: for correct predictions, it reinforces the underlying logic; for incorrect ones, it diagnoses errors and rectifies the intermediate reasoning steps. More details can be found in Appendix A.

Reasoning Finally, during the reasoning process, the original question is appended to the Context-Reward messages, and the combined messages are presented to the model. The model generates the final response from the enriched context messages, enabling reasoning guided by the prediction to similar question and the associated reward messages. Each generated response can be given as:

$$A_i = \text{LLM}(D_i, x), \tag{5}$$

where D_i denotes the message of Context-Reward messages in i-th time sample, incorporating contextual information from both the prediction and reward stages.

3.3 Stage Three: Self-Consistent Aggregation

Through Context Retrieval and Context-Reward reasoning, the model generates N corresponding responses to the original question. Each generated response is considered a candidate answer, $A = \{A_1, A_2, \ldots, A_N\}$, and the final answer is determined by majority voting, selecting the candidate that appears most frequently. This approach achieves high efficiency with minimal computational overhead. The final answer A^* can be formulated as:

$$A^* = \underset{c \in \mathcal{C}}{\text{argmax}} \sum_{i=1}^{N} \mathbf{1}(A_i = c). \tag{6}$$

4 Experiments

4.1 Experimental Setup

Models To evaluate the generality of Context-Aware Unlabeled Reward Reasoning across different backbones, we conduct experiments using Qwen2.5-7B-Instruct (Yang et al., 2024) and Llama3.1-8B-Instruct (Grattafiori et al., 2024) as instruct-tuned models. For large reasoning models (LRMs), we employ Qwen3-8B (Yang et al., 2025) and DeepSeek-R1-0528-Qwen3-8B (Guo et al., 2025).

Benchmarks To evaluate the applicability of CURE Reasoning across reasoning and knowledge-intensive tasks of varying difficulty, we assess its performance on three widely used reasoning benchmarks: MATH500 (Hendrycks et al., 2021), AMC (Li et al., 2024), and GSM8K (Cobbe et al., 2021), as well as on three more challenging reasoning benchmarks, AIME2024[1] , AIME2025[2] , and AMO-Bench (An et al., 2025). In the knowledge-intensive tasks, we evaluate CURE on two standard medical benchmarks: MedQA (Jin et al., 2021) and MedMCQA (Pal et al., 2022), in addition to the challenging medical knowledge benchmark MedXpertQA[3].

Baselines We compare our methods with two categories of LLMs: 1) Small Parameter LLMs: Mistral-Instruct-7B-v0.3 (Jiang et al., 2023), DeepSeek-Distill-Llama8B-Instruct (Guo et al., 2025), Qwen3-8B (Yang et al., 2025), GLM4-9B (GLM et al., 2024), Gemma2-9B (Team et al., 2024); and 2) Large Parameter LLMs: Qwen2.5-32B-Instruct (Yang et al., 2024), Qwen2.5-72B-Instruct (Yang et al., 2024), Qwen3-32B (Yang et al., 2025) and Llama3.1-70B-Instruct (Grattafiori et al., 2024).

Implementation Details We employ vLLM (Kwon et al., 2023) for online inference, deploying the model on 2*NVIDIA A100 (80GB) GPUs. For context retrieval, we utilize Qwen3-8B-Embedding (Zhang et al., 2025b) to generate vector representations and retrieve the **top-1** most similar question. During inference, we set the temperature to 0.6, top_p to 0.8, and the maximum number of generated tokens to 8192. In the majority voting, we select the answer that appears most frequently as the final prediction. If there are multiple options with the same frequency, we randomly select one as the final answer. We use accuracy as the evaluation metric.

[1] https://huggingface.co/datasets/HuggingFaceH4/aime_2024

[2] https://huggingface.co/datasets/opencompass/AIME2025

[3] https://huggingface.co/datasets/TsinghuaC3I/MedXpertQA/tree/main/Text

4.2 Main Results

CURE performs well on most tasks and models CURE achieves consistent and substantial improvements across various benchmarks compared to the same parameter-level LLMs. As shown in Table 1, CURE achieves an average performance gain of 25.29% on Qwen2.5-Intruct-7B and 23.5% on LLama3.1-Instruct-8B, demonstrating consistent gains on standard mathematical reasoning tasks. More notably, on the more challenging reasoning benchmarks, including AIME2024, AIME2025 and AMO-Bench, CURE leads to dramatic relative improvements ranging from 110% to over 250% for both Qwen2.5-7B and LLaMA3.1-8B. These results indicate that CURE is particularly effective at enhancing complex, multi-step reasoning capabilities where base models struggle most.

Beyond reasoning tasks, CURE also exhibits strong generalization to knowledge-intensive tasks. On the MedQA, LLaMA3.1-8B equipped with CURE outperform their respective backbones by 27.5%. On more challenging medical benchmark MedXpertQA, Qwen2.5-7B with CURE surpasses the backbone by 18.6%. These results underscore the broad applicability and robustness of CURE, demonstrating its effectiveness across both reasoning and knowledge-intensive tasks.

CURE Is Comparable to or Outperforms Large Parameter LLMs As shown in Table 1, Qwen2.5-7B with CURE achieves performance comparable to Qwen3-32B (non-thinking mode) across reasoning benchmarks. Notably, on the MATH500 benchmark, Qwen2.5-7B with CURE surpasses all other evaluated models. Furthermore, Llama3.1-8B with CURE even exceeds Llama3.1-70B-Instruct by 10.54 and 4.17 points on AMC and GSM8K, respectively. On MedQA and MedXpertQA, LLaMA3.1-8B with CURE slightly outperforms Qwen2.5-72B.

CURE Performs well on LRMs LRMs are increasingly becoming central to contemporary research and applications. We thus conducted experiments to evaluate the effectiveness of CURE on

Model	MATH	AMC	GSM8K	AIME24	AIME25	AMO	MedQA	MedCA	MedX	Avg
Small Parameter LLMs										
Mistral-7B-Instruct	17.00	2.42	48.27	0.00	0.00	0.81	48.20	44.90	11.40	19.22
DeepSeek-Distill-LLaMA-8B	59.20	21.80	58.28	2.71	1.46	0.19	48.20	44.90	11.40	27.57
GLM4-9B	48.40	17.07	72.21	5.00	0.00	0.69	58.90	49.80	12.78	29.43
Gemma2-9B	50.20	19.30	81.76	0.00	0.00	0.56	61.80	55.90	13.76	31.48
Qwen3-8B*	60.80	57.80	89.84	21.46	23.44	0.19	60.64	54.51	13.55	42.47
Large Parameter LLMs										
Qwen2.5-32B	57.70	32.80	86.07	7.90	10.21	1.44	75.26	64.83	13.87	38.90
Qwen2.5-72B	62.10	41.11	86.22	18.90	15.00	2.31	74.55	66.60	14.91	42.41
Llama3.1-70B	62.60	29.22	84.76	10.42	3.33	0.94	**78.40**	**70.05**	18.16	39.76
Qwen3-32B*	61.62	**59.33**	88.22	**31.35**	**25.42**	1.19	74.00	67.20	**18.28**	**47.40**
Qwen2.5-7B-Instruct	60.50	34.80	82.86	7.90	6.88	1.44	57.00	55.60	12.60	35.51
w/CURE	**71.00**	53.01	**90.45**	16.67	20.00	**4.00**	70.15	60.24	14.94	44.49
Δ	+10.50	+18.21	+7.59	+8.77	+13.12	+2.56	+13.15	+4.64	+2.34	+8.98
LLaMA3.1-8B-Instruct	48.40	23.30	80.91	4.60	1.46	0.56	58.70	56.00	13.20	31.90
w/CURE	56.60	39.76	88.93	10.00	3.33	2.00	74.86	63.78	15.43	39.41
Δ	+8.20	+16.46	+8.02	+5.4	+1.87	+1.44	+16.16	+7.78	+2.23	+7.51

Table 1: Main results in reasoning and knowledge-intensive tasks. The first column corresponds to small parameter LLMs, while the second column corresponds to large parameter LLMs. Within each segment, **bold** denotes the best score, underline indicates the second-best score, and Δ represents the gap between the original model and CURE. * indicates non-thinking mode.

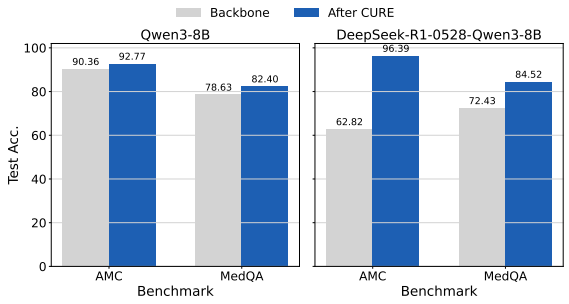

Figure 3: The evolution of LRM performance.

Models	Reasoning	Knowledge
Qwen2.5-Instruct-7B	32.40	41.73
w CURE	**42.52**	**48.44**
w/o retrival	40.56	45.49
w/o reward	38.99	45.91
w/o aggregation	36.55	44.46
Llama3.1-8B-instruct	26.54	42.63
w CURE	**33.44**	**51.36**
w/o retrival	31.99	48.83
w/o reward	31.52	49.38
w/o aggregation	26.18	47.66

Table 2: Ablation study of core components within CURE.

LRMs. Our results demonstrate that LRMs achieve significant improvements by integrating their inherent reasoning capabilities with our Context-Reward reasoning. As illustrated in Figure 3, on AMC, DeepSeek-R1-0528-Qwen3-8B achieves a substantially larger gain of 33.57 points. Furthermore, on the MedQA benchmark, Qwen3-8B demonstrates a substantial gain of 21.76 points, representing a near 35% relative increase. Similarly, DeepSeek-R1-0528-Qwen3-8B benefits from this approach, achieving a significant 16.70% improvement.

4.3 Ablation Study

To further assess the contribution of each stage in CURE, we conduct ablation studies on all three components: (1) w/o retrieval: contexts are selected by randomly sampling a question from the test set, rather than based on contextual similarity; (2) w/o reward: the original question is directly appended to the retrieved context without applying reward messages; (3) w/o aggregation: a single context reward messages is randomly selected and appended to the original question. The results are summarized in Table 2, with the detailed descriptions are provided in Appendix B. Notably, the w/o reward setting exhibits a pronounced performance drop, suggesting that effective contextual learning for reasoning relies heavily on the integration of reward messages.

5 Analysis

5.1 Samping more is better?

To investigate the relationship between the number of samples and model accuracy, we conduct experiments with sample@4, sample@8, and sample@16, as illustrated in Figure 4.

Effect of Increasing Sample Size We observe a consistent performance gain as the number of samples increases. Specifically, when scaling from

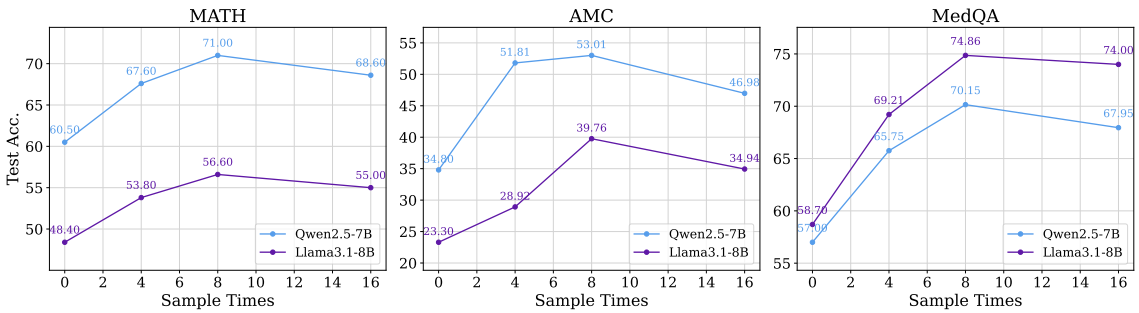

Figure 4: Effect of sample times on test accuracy across benchmarks.

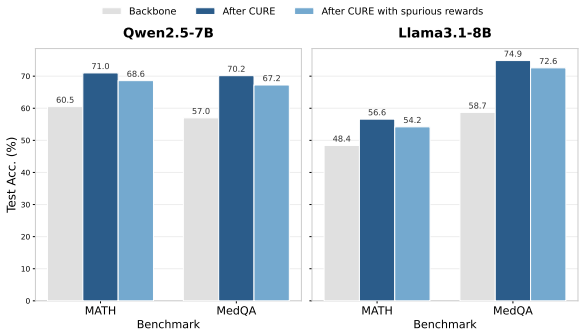

Figure 5: Results of spurious rewards.

sample@4 to sample@8, Qwen2.5-7B exhibits a substantial improvement on MedQA, with accuracy rising from 65.75 to 70.15. Notably, this gain is not confined to knowledge-intensive tasks; reasoning performance also improves accordingly.

The effectiveness of increasing the sample size is rooted in the principles of self-consistency, which posits that while incorrect reasoning is often idiosyncratic, correct reasoning paths tend to converge toward a consistent answer. By generating multiple samples, we increase the likelihood that the resulting pseudo-labels represent this consensus, thereby mitigating the risk of introducing noisy or erroneous reward messages during the reward process.

Performance Degradation with Excessive Sampling Despite the benefits of moderate sample scaling, we observe a consistent performance decline once the number of votes exceeds a certain threshold. Specifically, increasing the voting budget from sample@8 to sample@16 leads to performance degradation across all evaluated benchmarks. We observe that when the model encounters problems at or above the upper limit of its capabilities, the fundamental assumption of self-consistency—that correct reasoning paths converge while errors remain idiosyncratic—begins to fail. In these high-difficulty regimes, the model becomes prone to systematic hallucinations, where multiple reasoning trajectories converge on the same plausible but incorrect conclusion.

5.2 Do Reward Messages Help?

Inspired by (Shao et al., 2025), we wanted to explore the effectiveness of reward messages generated by pseudo-labels. We also conducted a spurious reward experiment.

As shown in Figure 5, we observe a consistent performance gap between CURE and the spurious rewards setting across all configurations. For Qwen2.5-7B-Instruct, CURE outperforms spurious rewards by 1.20 points on MATH500 and 2.91 points on MedQA. A similar trend is observed for Llama3.1-8B-Instruct, where CURE achieves gains of 2.40 and 2.28 points on the two benchmarks, respectively.

These results highlight the critical role of reward message correctness. When rewards aligned with the majority consensus, the model is encouraged to consolidate reliable reasoning trajectories. In contrast, spurious reward introduces systematic inconsistencies between correct answers and feedback signals, thereby distorting the learning signal and limiting performance improvements.

Interestingly, the spurious reward setting still outperforms the vanilla baseline. We attribute this phenomenon to several factors. Integrating the prediction reward into the context messages facilitates a more effective dual descent gradient. This approach establishes a more precise dual formulation that bridges ICL and fine-tuning. For relatively simple questions, the model often derives the correct answer based on prior knowledge or commonsense reasoning. Even when prompted to revise its reasoning, the model tends to converge to the same conclusion, meaning that spurious rewards have minimal impact on the final prediction.

Models	GSM8K	MedQA
Qwen2.5-7B	82.68	57.00
CURE	**90.45**	**70.15**
Retrieval from Training	89.70	68.42
Llama3.1-8B	80.91	58.70
CURE	**88.93**	**74.86**
Retrieval from Training	88.25	72.43

Table 3: Results of retrieval from training and test.

Models	MATH	AMC	MedQA
Qwen2.5-7B	60.50	34.80	57.00
CURE	**71.00**	**53.01**	**70.15**
BoN	69.60	39.76	64.02
MCTS	65.00	36.14	62.13

Table 4: Comparison of CURE with other TTS methods.

In contrast, for sufficiently difficult questions where the model consistently fails to produce correct answers, spurious rewards actually tell them that they answered incorrectly and help the model reconsider its reasoning trajectory.

5.3 On the Authenticity of Pseudo-Labels

For benchmarks that only have test sets, we generate pseudo-labels via multiple samplings followed by majority voting. To assess whether these pseudo-labels reliably approximate the true labels, we conduct controlled experiments on benchmarks that include training sets. Specifically, for each query, we retrieve the most similar question from the training set, perform multiple samplings, and directly compare the predictions with the ground-truth answers to assign reward messages.

As shown in Table 3, the performance gap between the two approaches is minimal on reasoning tasks. In contrast, pronounced differences emerge on knowledge-intensive tasks. We attribute this discrepancy to distributional shifts in knowledge domains between the training and test sets, which constrain the effectiveness of training-set retrieval and result in performance variations. These results indicate that the pseudo-labels are highly reliable and closely approximate ground truth. Despite the absence of labeled training data, CURE effectively aggregates model-consistent signals through repeated sampling, producing reward messages that are comparable to those obtained with true labels. This observation underscores the robustness of CURE's pseudo-labeling mechanism and supports its applicability in settings where annotated training data are unavailable.

5.4 Comparisons with TTS Methods

We compare CURE with several commonly used TTS methods, including BoN and MCTS, as shown in Table 4. Both BoN and MCTS are implemented using OpenR (Wang et al., 2024a), more details are available in C.5.

Overall, CURE consistently achieves the best performance across all evaluated tasks, substantially outperforming both the base model and existing TTS baselines such as BoN and MCTS. These improvements suggest that CURE more effectively guides the model toward high-quality reasoning trajectories, rather than relying solely on increased deeper search. In particular, while BoN benefits from diversified sampling, its gains on AMC remain limited, and MCTS exhibits weaker overall performance under the same computational budget. By contrast, CURE leverages informative contexts that predictions and reward messages, which the model internalizes through In-Context Learning, resulting in more reliable inference trajectories. In summary, CURE's ability to substantially enhance accuracy while maintaining high efficiency highlights its practical value and strong potential for reasoning tasks.

6 Conclusion

In this paper, we propose a novel framework for Test-Time Scaling, Context-Aware Unlabeled Reward Reasoning. This framework offers a simple yet effective approach to enhancing model performance at test time by leveraging contextual and reward messages from unlabeled data, making it applicable to a wide range of reasoning and knowledge-intensive tasks. A key innovation of CURE is its Context-Reward reasoning stage, which consists of three core processes: Prediction, Reward, and Reasoning. These processes enable the model to dynamically refine its reasoning by incorporating self-reflective contextual cues, without requiring direct intervention. CURE improves model performance by constructing reward-informed input contexts that guide the model toward more accurate predictions through indirect reasoning refinement. This approach preserves the model's autonomy, leading to significant improvements in prediction accuracy. As a result, CURE presents itself as a promising method for Test-Time Scaling.

Limitations

Since CURE relies on dialogue-based interaction, the current implementation is limited to instruction-tuned LLMs for introducing predictions. Extending CURE to base pretrained LLMs may be necessary to further improve its generality. In addition, future work should evaluate our approach on both open-source and closed-source large-scale models with substantially more parameters. In scenarios involving ambiguous models, relying solely on a majority vote of the inferred results may be overly simplistic and one-sided. To improve the robustness of the decision-making process, it is essential to distinguish between different inference paths. One effective way to achieve this is by incorporating metrics such as Perplexity (PPL) or entropy to quantify the uncertainty associated with each inference. By considering these metrics, we can better differentiate between competing hypotheses, and ultimately, use the majority vote of the final results to select the most plausible answer.

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*.

Shengnan An, Xunliang Cai, Xuezhi Cao, Xiaoyu Li, Yehao Lin, Junlin Liu, Xinxuan Lv, Dan Ma, Xuanlin Wang, Ziwen Wang, and Shuang Zhou. 2025. Amo-bench: Large language models still struggle in high school math competitions. *Preprint*, arXiv:2510.26768.

Vidhisha Balachandran, Jingya Chen, Lingjiao Chen, Shivam Garg, Neel Joshi, Yash Lara, John Langford, Besmira Nushi, Vibhav Vineet, Yue Wu, and 1 others. 2025. Inference-time scaling for complex tasks: Where we stand and what lies ahead. *arXiv preprint arXiv:2504.00294*.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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*.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2022. Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers. *arXiv preprint arXiv:2212.10559*.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, and 1 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

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 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. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

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

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in*

neural information processing systems, 35:22199–22213.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, and 1 others. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9.

Junlong Li, Daya Guo, Dejian Yang, Runxin Xu, Yu Wu, and Junxian He. 2025. Codei/o: Condensing reasoning patterns via code input-output prediction. *arXiv preprint arXiv:2502.07316*.

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*.

Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen Zhou. 2025. Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling. *arXiv preprint arXiv:2502.06703*.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pages 248–260. PMLR.

Raul Rojas and Raúl Rojas. 1996. The backpropagation algorithm. *Neural networks: a systematic introduction*, pages 149–182.

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, and 1 others. 2025. Spurious rewards: Rethinking training signals in rlvr. *arXiv preprint arXiv:2506.10947*.

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

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

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

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024b. 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.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.

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

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. 2024. Free process rewards without process labels. *arXiv preprint arXiv:2412.01981*.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024. Rest-mcts*: Llm self-training via process reward guided tree search. *Advances in Neural Information Processing Systems*, 37:64735–64772.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Zhihan Guo, Yufei Wang, Irwin King, Xue Liu, and Chen Ma. 2025a. What, how, where, and how well? a survey on test-time scaling in large language models. *arXiv preprint arXiv:2503.24235*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, and 1 others. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

A CURE Implementation Details

A.1 Question prompt template

In MedQA and MedCAQA, we use zero-shot CoT template adapted from Deepseek-R1.

```
Q: {question}\nA: Please reason step by step, and put your final answer (selected from options A to D) within \boxed{}.
```

In MedXpertQA, because the range of answer is different, this is another template.

```
Q: {question}\nA: Please provide a step-by-step explanation, followed by your final answer (selected from options A to J) within \boxed{}.
```

In Reasoning benchmark, we will use the following template to guide the responses.

```
Q: {question}\nA: Please reason step by step, and put your final answer within \boxed{}.
```

A.2 Reward template

After the model generates a prediction for a retrieval question, it is verified against the corresponding the pseudo-label. If the prediction is correct, a positively reward label is appended to the context, affirming the validity of the reasoning process. In cases of incorrect predictions, instead of explicitly pointing out the error, a supportive and constructive reward label is introduced. This approach encourages further reflection and exploration without directly identifying the mistake.
When the prediction is correct.

```
User: Well done! Your answer is correct.
```

When the prediction is wrong.

```
User: Unfortunately, your answer is wrong! Review your previous answer. Find the reason for the mistake.
```

B A Detail ablation study

In ablation analysis, we selected MATH500, AMC, GSM8K, AIME2024, AIME2025 and AMO-Bench as the reasoning benchmarks, and MedQA, MedCAQA and MedXpertQA as the knowledge-intensive benchmarks. The detailed evaluation scores are presented table 6.

Context retrival Replacing the context similarity–based retrieval strategy with random retrieval leads to a consistent performance degradation across models. For Qwen2.5-7B-Instruct, the average score of reasoning tasks drops by 1.96 points, while the average score of knowledge-intensive tasks decreases by 2.95 points. A similar trend is observed for Llama3.1-8B-Instruct, with declines of 1.75 points in reasoning and 2.53 points in knowledge-intensive tasks. Notably, on MedXpertQA, Llama3.1 under the w/o retrieval setting performs even worse than the baseline, highlighting the critical role of effective retrieval in this domain. This shows that learning from unrelated context can make LLMs more prone to hallucinations, ultimately undermining both their reasoning and knowledge capabilities.

Reward To evaluate the contribution of the reward messages, we directly append the original question to the context messages without applying the reward messages. We observe a substantial 8.3% decline in the reasoning performance of Qwen2.5-7B, which is larger than the corresponding 5.2% reduction in knowledge-intensive performance. On the AMC benchmark, the performance of Qwen2.5-7B decreases by 7.23 points, while

Dataset	Train Num	Test Num	Options Num
MATH500	0	500	N/A
AMC	0	83	N/A
GSM8K	7473	1319	N/A
AIME2024	0	30	N/A
AIME2025	0	30	N/A
AMO-Bench	0	50	N/A
MedQA	10178	1273	4
MedCAQA	182822	4183	4
MedXpertQA	5	2450	10

Table 5: Statistics of benchmarks

Llama3.1 exhibits a decline of 4.82 points. These results suggest that contextual learning for reasoning is particularly sensitive to the design and application of reward messages.

Self-Consistent Aggregation When a single context–reward message is randomly sampled and appended to the original query, we observe a pronounced performance degradation. Specifically, Llama3.1-8B-Instruct exhibits a 7.33 point decrease in reasoning tasks and a 3.70 point decrease in knowledge-intensive tasks. Under the w/o aggregation setting, Llama3.1-8B consistently achieves the worst performance across all configurations, even underperforming the baseline on reasoning tasks. A similar, though slightly less severe, degradation is observed for Qwen2.5-7B, indicating that unvoted single-path contextual inference substantially harms model performance. The voting mechanism aggregates multiple reasoning paths, effectively filtering out spurious or misleading reward messages caused by pseudo labels. Removing voting liminates this safeguard, causing performance to collapse, even below the baseline, suggesting that poor contextual signals are worse than no context at all.

C Additional Experiments Details

C.1 Baseline Models

For all baseline models, we use zero-shot to inference. For Large Reasoning Models, we follow the corresponding recommended prompting guidelines to remove the system prompt.
The zero-shot template is :

```
Q: {question}\nA: Put your final
answer within\boxed{}.
```

C.2 Data Statistics

The detailed benchmark statistics are shown in Tables 5.

C.3 Evaluation Metrics

We employ accuracy as our evaluation metric. To ensure statistical robustness on the AIME 2024 and AIME 2025 datasets, we report the mean accuracy across 32 independent trials. For AMO-Bench, results are averaged over 8 runs. Performance on all remaining datasets is reported based on a single experimental trial.

C.4 Answer cleaning

As we guide the model in generating answers, we use the `\boxed{}` format to standardize the final answer output. However, due to differences across models, we apply various regular expressions to extract the final answer accurately, as shown in Listing 1.

C.5 Details in TTS Methods

Both Best-of-N and MCTS are implemented using OpenR (Wang et al., 2024a). For reasoning tasks, we employ Math-Shepherd-Mistral-7B-PRM (Wang et al., 2024b) as the process reward model (PRM). For knowledge-intensive tasks, due to the lack of a mature PRM, we use the base model itself as the PRM and assign a score in the range [0,1] to each generated step.

For Best-of-N, we set the temperature to 0.6, generate 8 candidate sequences with a maximum of 4096 new tokens, and select the final prediction via majority voting.

For MCTS, we set the temperature to 0.6, generate a single sequence, constrain the maximum tree width to 4 and the maximum depth to 10, and allow up to 4096 new tokens.

Models	MATH500	AMC	GSM8K	AIME2024	AIME2025	AMO	MedQA	MedCA	MedX
Qwen2.5-Instruct-7B	60.50	34.80	82.86	7.90	6.88	1.44	57.00	55.60	12.60
w CURE	**71.00**	**53.01**	**90.45**	**16.67**	**20.00**	**4.00**	**70.15**	**60.24**	**14.94**
w/o retrieval	70.40	51.81	90.30	15.00	13.85	2.00	65.83	57.42	13.22
w/o reward	69.00	45.78	90.22	14.90	12.81	1.25	66.38	58.83	12.53
w/o aggregation	64.40	45.78	88.78	11.77	7.81	0.75	63.31	56.63	13.43
Llama3.1-8B-instruct	48.40	23.30	80.91	4.60	1.46	0.56	58.70	56.00	13.20
w CURE	**56.60**	**39.76**	**88.93**	**10.00**	**3.33**	**2.00**	**74.86**	**63.78**	**15.43**
w/o retrieval	54.00	37.34	88.55	10.00	1.04	1.00	72.82	61.70	11.96
w/o reward	54.60	34.94	87.34	9.15	1.04	2.00	71.87	61.30	14.98
w/o aggregation	42.28	25.30	82.41	5.61	1.25	0.25	70.15	57.90	14.94

Table 6: Accuracy in ablation study of retrieval, reward, and aggregation components to reasoning and knowledge-intensive tasks.

Listing 1: Implementation of the boxed answer extraction function.

```
def extract_boxed_answer_r1(text):
    if text is None or len(text) == 0:
        return None
    if len(text) == 1:
        return text
    match = re.search(r'\\boxed{((?:[^{}]|\{[^{}]*\})*)}', text)
    if match:
        inner_text = match.group(1)
        if len(inner_text) == 0:
            return None
        elif len(inner_text) != 1:
            text_match = re.search(r'\\text\{([A-Za-z])\}', inner_text)
            if text_match:
                return text_match.group(1)
            else:
                return inner_text
        else:
            return inner_text
    else:
        match = re.search(r'\\boxed{(.*)}', text)
        if match:
            inner_text = match.group(1)
            if inner_text.startswith('(') and inner_text.endswith(')'):
                inner_text = inner_text[1:-1]
            return inner_text

    answer_match = re.search(
        r'(?:Final\s+)?Answer\s*:\s*\(?([A-Z])\)?', text, re.IGNORECASE)
    if answer_match:
        return answer_match.group(1)
    return None
```