CG-MAV: Confidence-Guided Multi-Agent Verification for LLM Reasoning

Anonymous ACL submission

Abstract

Large Language Models have shown great potential in reasoning tasks through test-time scaling methods like generating multiple candidate solutions for a given task. However, reliably selecting the correct answer from these candidates remains challenging. Existing Self-certainty-based selection methods are effective on easy tasks but become unreliable on hard ones. We propose Confidence-Guided Multi-Agent Verification (CG-MAV) that uses confidence to distinguish easy and hard tasks and applies different strategies accordingly. CG-MAV leverages Self-certainty-based Borda voting not only as a selection signal but also as an indicator of task difficulty, enabling a classification of tasks into easy and hard categories. Easy tasks are handled through direct selection, while hard tasks are processed through multi-agent verification. Each verification agent is assigned a clear and specific persona, focusing on a distinct aspect of solution correctness. Extensive experiments on two reasoning datasets across multiple models demonstrate the superiority and generalization of our proposed CG-MAV. [1]

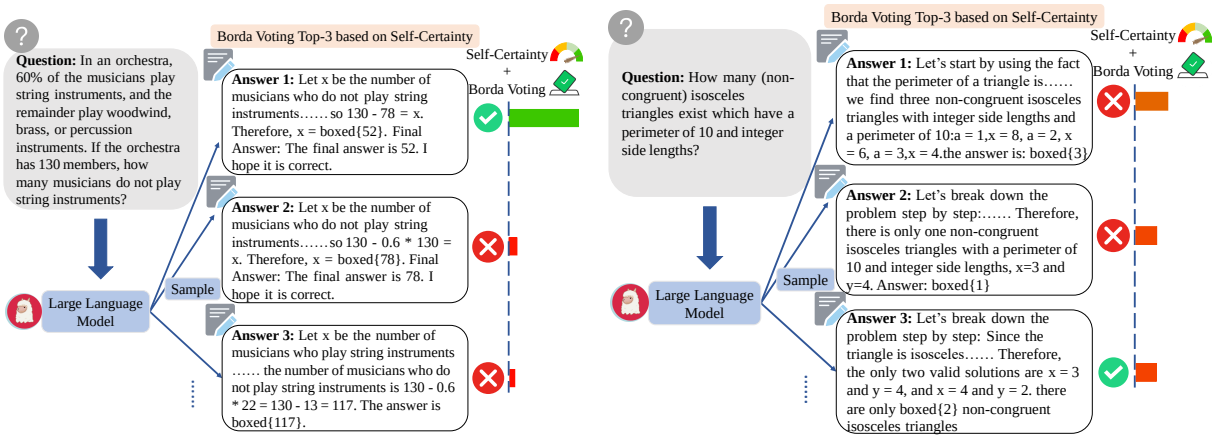
1 Introduction

Large Language Models (LLMs) have demonstrated strong capabilities in reasoning tasks. Recent work on test-time scaling (Ye et al., 2025; Wang et al., 2025) shows that LLM reasoning can be further enhanced by allocating more computation at test time, typically through aggregating multiple reasoning paths. However, how to aggregate multiple reasoning paths into a correct answer remains a challenge.

A representative strategy is to select the most frequently occurring final answer, such as self-consistency (Wang et al., 2022). While effective on simple tasks, the assumption of self-consistency that correct reasoning chains dominate the sampled outputs often breaks down on challenging tasks, where multiple similar yet flawed solutions are generated (Tan et al., 2025). Recently, confidence-based methods (Geng et al., 2024; Ren et al., 2023) have gained increasing attention. These approaches (Razghandi et al., 2025; Leang et al., 2025) rank candidate outputs using token-level likelihoods or confidence scores derived from the model's output distribution. Among them, Self-certainty (Kang et al., 2025) measures the LLM's confidence based on its predictive distribution, and selects the response with the highest confidence as the final answer. Although Self-certainty performs well on simple problems where the LLM's predictive distribution is sharp, it becomes less effective for challenging reasoning tasks with diffuse predictive distributions, where the model exhibits limited confidence in candidate responses, making direct answer selection based solely on Self-certainty difficult. To overcome this problem, many study proposes multi-agent frameworks, which leverages LLM collaboration to evaluate candidate responses (Agashe et al., 2023; Wang et al., 2024a). These approaches exploit the diversity of reasoning perspective to mitigate the risk of over-dependence on a single solution (Chen et al., 2024; Li et al., 2023). However, these methods treat all tasks uniformly and fail to allocate computational resources to tasks that require more careful evaluation (Ma et al., 2025).

In this paper, we propose a **C**onfidence-**G**uided **M**ulti-**A**gent **V**erification framework, named **CG-MAV**. First, we propose the Self-Certainty based Borda Voting (Emerson, 2013) (SCBV) to evaluate multiple candidate responses. Specifically, candidates are ranked by Self-certainty and assigned votes based on their relative ranks. SCBV is not only a selection signal, but also an indicator of task difficulty. Figure 1a shows that when the LLM's predictive distribution is sharply concentrated, the model exhibits high confidence in its response, indicating that the task is easy. Conversely, when all

[1] The code will be open upon acceptance of the paper.



(a) easy task (b) hard task

Figure 1: The distribution of SCBV for the top-3 candidate responses on (a) easy task and (b) hard task.

candidates exhibit low SCBV as shown in Figure 1b, the LLM is unconfident, indicating the task is hard. We use SCBV to classify each task into easy or hard, relying on it for answer selection in easy tasks while avoiding over-reliance in hard tasks. Second, for hard tasks, we introduce a multi-agent verification that focuses exclusively on answer validation. Unlike prior multi-agent frameworks that entangle reasoning and debate within each agent and consequently overburden individual agents, our framework assigns each agent a distinct verification persona, which consists of a particular verification perspective and a corresponding verification strategy. Then, each agent produces a persona-focused binary judgment, i.e., true or false, on each of the top-3 SCBV candidate solutions to indicate whether the candidate answer satisfies the corresponding verification criterion. This enables more reliable evaluations and facilitates effective aggregation across complementary perspectives. By assigning each agent a focused verification persona, we reduce the burden placed on individual agents and improve the reliability of their judgments.

Our contributions can be summarized as follows: (1) We identify a fundamental limitation of Self-certainty based answer selection and propose the SCBV-guided task classification strategy that mitigates this limitation by adaptively handling easy and hard tasks (2). We design a multi-agent verification framework, where agents assigned with different persona focus on distinct verification perspective. The final output is obtained by aggregating these judgments. (3) We conduct extensive experiments on two datasets across models. The consistent superior performance of CG-MAV demonstrates its effectiveness and generalization.

2 Related Work

Recent efforts to improve the reasoning capabilities of LLMs follow two directions: reinforcement learning–based post-training and test-time scaling. The former enhances reasoning through explicit optimization with reward signals, while the latter improves performance by increasing test-time computation without additional training.

Reinforcement learning for LLMs. Reinforcement learning has been a central paradigm for post-training LLMs in both alignment and reasoning. Early work on reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022) optimizes outputs toward human preferences by training a reward model and applying outcome supervision over complete responses. More recently, reinforcement learning with verifiable rewards (RLVR) has emerged as effective alternatives, where automated verifiers provide end-of-generation rewards, leading to substantial performance improvements on challenging reasoning tasks such as mathematical problems (Guo et al., 2025; Zeng et al., 2025; Hu et al., 2025; Lambert et al., 2024). Algorithms such as Group Relative Policy Optimization (GRPO) have been shown to be particularly effective (Shao et al., 2025). Beyond outcome-level rewards, process supervision has gained attention for offering finer-grained guidance during reasoning. Most existing methods implement process supervision by training a process reward model (PRM) from human-annotated or automatically generated signals and using it as a static reward during reinforcement learning (Lightman et al., 2023; Luo et al., 2024; Wang et al., 2024b; Setlur et al., 2024). However, static PRM are vulnerable to distribution shift

and reward hacking as training progresses (Kazemnejad et al., 2024). These limitations highlight the difficulty of maintaining reliable reward signals, and motivate training-free approaches like test-time scaling.

Test-Time Scaling. As high-quality and unexplored training data become increasingly difficult to obtain, Test-Time Scaling has gained more and more attention, which improves reasoning performance by scaling test-time computation without any training. Prior work can be broadly categorized into parallel and sequential approaches. Parallel methods sample many candidate generations and select among them, such as Best-of-N research (Amini et al., 2024; Sessa et al., 2024) and Self-consistency (Wang et al., 2022). These methods exploit sampling diversity and have demonstrated near log-linear improvements as the number of samples grows. Sequential methods expand computation along a single trajectory, such as Chain of Thought (Wei et al., 2022) and Self-refine (Madaan et al., 2023), where the model iteratively evaluate and improve responses. More recent work integrates parallel and sequential test-time scaling. For example, Monte Carlo Tree Search (Guan et al., 2025) expands a search tree over reasoning steps and selection actions based on iterative evaluation. However, these test-time scaling methods allocate computation uniformly across tasks and rely on frequency or likelihood, which can be inefficient and unreliable for hard tasks.

In contrast, our proposed CG-MAV focuses on improving answer selection at test time through confidence-guided verification, rather than uniformly increasing inference computation or relying on frequency or likelihood. We leverage Self-certainty as an indicator of task difficulty to distinguish between easy and hard tasks, and selectively invoke multi-agent verification on hard tasks where additional reasoning is necessary. This design enables more effective allocation of test-time computation, while avoiding both unnecessary overhead on easy tasks and unreliable selection on hard tasks.

3 Methodology

The overview of CG-MAV is illustrated in Figure 2. CG-MAV is centered on SCBV, which reflects the reliability of generated solutions. Each task is categorized as either easy or hard based on a SCBV threshold. For easy tasks, the solution with highest SCBV is selected as the final answer directly, whereas hard tasks are further verified by multiple verification agents that focus on different perspective and strategy. The algorithm for the proposed CG-MAV is detailed in Algorithm 1.

Algorithm 1 Confidence-Guided Multi-Agent Verification (CG-MAV)

Input: prompt x; model $\mathcal{M}$; confidence metric $C(\cdot)$; threshold τ; verification agents $\mathcal{A} = \{a_1, \ldots, a_D\}$; sample size K
Output: selected answer y^*
Sample K candidate answers:
  $\mathcal{Y} = \{y^{(k)} \sim G(\cdot \mid x)\}_{k=1}^{K}$
Compute self-confidence for each candidate:
  $c^{(k)} = C\left(y^{(k)}\right), \ \forall y^{(k)} \in \mathcal{Y}$
Determine task difficulty:
if $\exists\, k \in [1, K]$ s.t. $c^{(k)} \geq \tau$ **then**
  classify as easy task
else
  classify as hard task
end if
if easy task **then**
  Select the most confident answer:
    $y^* = \arg\max_{y^{(k)} \in \mathcal{Y}} c^{(k)}$
else
  Perform multi-agent verification:
  for $y^{(k)} \in \mathcal{Y}$ **do**
    for $a_d \in \mathcal{A}$ **do**
      Obtain binary approval $z_d(y^{(k)}) \in \{0, 1\}$
    end for
  end for
  Aggregate verification scores:
    $\mathcal{S}(y^{(k)}) = \frac{1}{D} \sum_{d=1}^{D} z_d(y^{(k)})$
  Select the best verified answer:
    $y^* = \arg\max_{y^{(k)} \in \mathcal{Y}} \mathcal{S}(y^{(k)})$
end if
return y^*

3.1 LLM Background

We consider a reasoning task where LLM is prompted to generate answers $y = (y_1, ..., y_m)$ from an input $x = (x_1, ..., x_n)$. At step i, the LLM outputs a vector of logits $l \in \mathbb{R}^{|V|}$, where V denotes the vocabulary and $|\cdot|$ denotes the cardinality of a set. These logits are transformed into a probability distribution $p(\cdot|x, y_{<i}) \in [0, 1]^{|V|}$ over candidate next tokens. This distribution represents the LLM's estimated likelihood for each token being generated at step i.

3.2 Self-certainty based Borda voting

Prior work has shown that distribution over the full vocabulary provide a more reliable estimate of LLM's confidence than token-level likelihood alone, particularly in open-ended generation and reasoning tasks. Specifically, Self-certainty derived from the divergence between the model's predictive distribution and a uniform distribution have been empirically validated as effective indicators

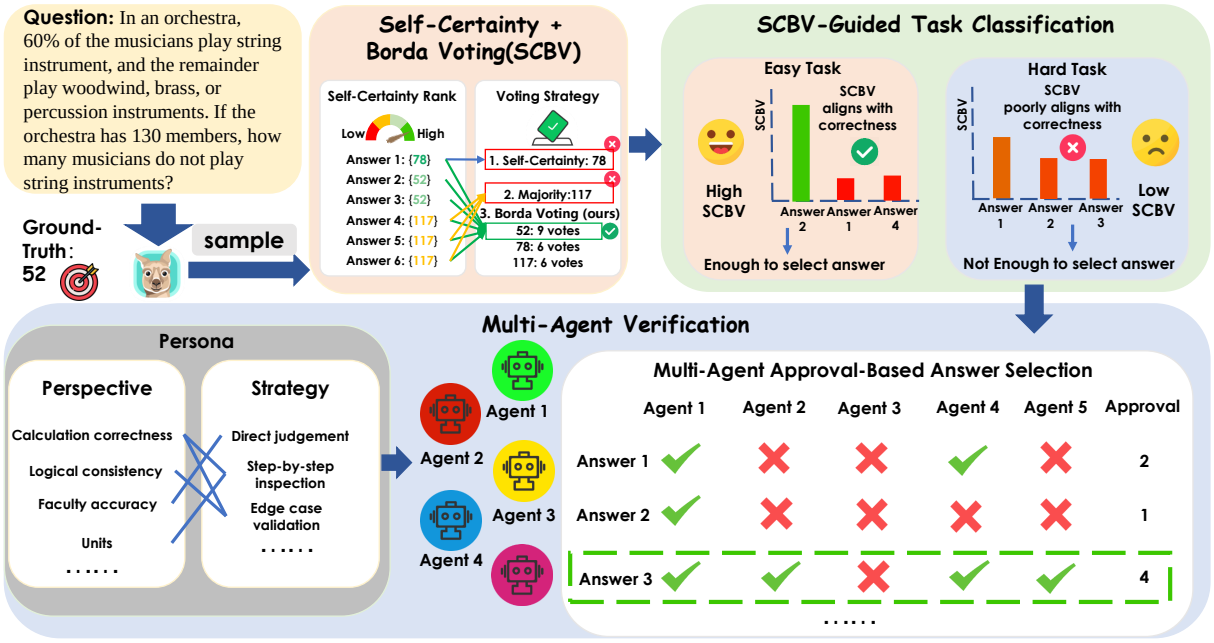


Figure 2: An overview of our proposed CG-MAV.

of internal certainty in LLMs. Following this line of work, we adopt a confidence metric as:

$$C(x,y) = \frac{1}{n|V|}\sum_{i=1}^{n}\sum_{j=1}^{|V|} log(|V| \cdot p(j|x, y_{<i})), \tag{1}$$

where n denotes the length of the output y.

Since selecting the single most certain sample can ignore answer frequency, we apply Borda voting to aggregate certainty rankings across samples, combining confidence with consensus. Specifically, given N outputs, we rank them by Self-certainty and assign each output a vote based on its rank:

$$v(r) = (N - r + 1)^p, \tag{2}$$

where r denotes the rank position and p controls ranking influence. Votes from all outputs that produce the same final answer are summed.

While Self-certainty provides an effective confidence signal in many settings, its effectiveness diminishes in challenging reasoning tasks. In such case, the model's internal uncertainty is intrinsically high, leading to uniformly low and weakly separated Self-certainty across sampled candidates. As a result, rather than relying on confidence for direct answer selection, we introduce a SCBV-guided strategy to identify cases where require further verification.

3.3 SCBV-Guided Task Classification

Given an input x, we sample a set of candidate outputs $y = \{y^{(1)}, y^{(2)}, ..., y^{(K)}\}$. For each candidate $y^{(k)}$, we compute its SCBV $C^{(k)}$. An intuitive idea is selecting the candidate with the highest SCBV. However, extensive empirical evidence suggests that, in challenging reasoning tasks, the SCBV of multiple sampled candidates are consistently low and exhibit weak separation. In such cases, SCBV based ranking becomes unreliable and often fails to distinguish correct solutions from plausible but incorrect ones.

To address this issue, we propose a SCBV-guided task classification mechanism that explicitly distinguishes between easy and hard tasks. Empirically, easy tasks exhibit concentrated SCBV, with at least one answer achieving a high SCBV, which makes SCBV based selection reliable. In contrast, hard tasks are characterized by uniformly low and weakly separated SCBV. Based on this observation, we introduce a decision boundary that determines whether SCBV along is sufficient for task solution instead of using SCBV as a ranking signal solely.

Let $C = \{C^{(1)}, C^{(2)}, ..., C^{(K)}\}$ denote the set of SCBV for all candidates of a task. We define a SCBV threshold τ and partition the tasks set into two categories via the following decision rule:

$$\mathcal{T}(x) = \begin{cases} Easy, & \text{if } \max\limits_{k\in\{1,...,K\}} C^{(k)}(x) \geq \tau, \\ Hard, & \text{otherwise}, \end{cases} \tag{3}$$

where $\mathcal{T}(\cdot)$ denotes a classifier that determines the difficulty of task based on its SCBV.

When $\mathcal{T}(x) = Easy$, at least one answer ex-

hibits sufficiently high SCBV, indicating that the model's predictive distribution is relatively concentrated. In this case, SCBV based selection is reliable, and we select the answer by :

$$y^* = \arg\max_{y^{(k)} \in \mathcal{Y}} C^{(k)}. \quad (4)$$

In contrast, when $\mathcal{T}(x) = Hard$, all sampled candidates fall below the SCBV threshold. Under this condition, SCBV becomes a weak indicator of correctness and do not provide a stable ordering among candidates. Rather than forcing a fragile decision based on marginal SCBV differences, we explicitly defer direct answer selection and process all candidates with the subsequent multi-agent verification stage, where they are evaluated using distinct verification personas beyond the model's internal certainty estimates. This mechanism allows SCBV to be exploited when it is informative, while preventing overcommitment when it collapses.

3.4 Multi-Agent Verification

For a hard task, where no candidate exceeds the SCBV threshold and SCBV based selection becomes unreliable, we process the entire candidates set $y = \{y^{(1)}, ..., y^{(K)}\}$ with subsequent verification stage. We define a set of verification agents $\mathcal{A} = \{a_1, a_2, ..., a_D\}$, where each agent a_d is assigned a distinct verification persona. Each persona is specified by a particular verification perspective together with a corresponding verification strategy . The verification personas are designed to emphasize complementary aspects of correctness, encouraging agents to focus on distinct evaluation aspects. Each verification agent independently evaluates a candidate solution and outputs a binary judgment (true or false). For a given candidate $y^{(k)}$, we collect the binary judgments from all D agents, denoted as $z(y^{(k)})$, which we refer to as the verification outcome of the candidate. To qualify the degree of multi-agent agreement, we introduce a function:

$$\mathcal{S}(y^{(k)}) = \frac{1}{D} \sum_{d=1}^{D} z_d(y^{(k)}), \quad (5)$$

where $z(y^{(k)}) = 1$ if the d-th agent outputs True, and 0 otherwise. Eq. 5 measures the outcome of verification agents that output True for the candidate. Rather than reflecting the strength of any individual agent's belief, $\mathcal{S}(\cdot)$ captures collective consistency across independent verification personas. Candidate selection is then formulated as a consensus maximization problem:

$$y^* = \arg\max_{y^{(k)} \in \mathcal{Y}} \mathcal{C}(y^{(k)}). \quad (6)$$

This decision rule does not rely on a single uncertain signal, but instead bases correctness on the agreement among multiple independent verification agents.

4 Experiments

4.1 Experimental Setting

Datasets. We conduct experiments on two widely used mathematical reasoning datasets, GSM8K (Cobbe et al., 2021) and MATH(Hendrycks et al., 2021), which are commonly adopted to evaluate multi-step reasoning capabilities of LLMs. GSM8K is a benchmark of grade-school–level math word problems designed to evaluate multi-step numerical reasoning. It consists of 7.47k training examples and 1.32k test questions, each requiring a sequence of arithmetic reasoning steps to obtain the final answer. MATH is a substantially more challenging dataset consisting of 7.50k training and 5.00k test problems spanning diverse domains, including algebra, geometry, number theory, and precalculus. Each problem is annotated with a full solution and a final boxed answer. Compared to GSM8K, MATH exhibits higher structural complexity, longer reasoning chains, and greater ambiguity in intermediate steps, making it a representative setting for difficult reasoning scenarios. For both datasets, we evaluate on official test splits and follow prior work in extracting and normalizing final answer for automatic evaluation.

Dataset	Type	Level	Train	Test
GSM8K	1	1	7.47k	1.32k
MATH	7	5	7.50k	5.00k

Table 1: Statistics of experimental dataset. Type denotes the kinds of problems in the dataset, and level measures the difficulty of problems.

Implementation Details. We use the Llama-3-8B-Instruct (Dubey et al., 2024) as our base model for all experiments. When sampling multiple candidate answers, we set the temperature to 0.6 and top-p to 0.9. During the multi-agent verification stage, we adopt greedy decoding. All experiments are run on NVIDIA A40 GPUs

Baselines. In order to comprehensively evaluate the effectiveness of our proposed CG-MAV,

we compare it with various baselines, covering both single model baselines and multi-agent baselines. Specifically, the single model baselines include Greedy (Brown et al., 2020), FirstAns (Kang et al., 2025), Perplexity(Hu et al., 2024), Self-consistency(Wang et al., 2022), and Self-certainty(Kang et al., 2025). Detailed description are provided below.

- **Greedy** (Brown et al., 2020) generates a single output by selecting the most probable token at each step, without any aggregation or verification across multiple samples.
- **FirstAns** (Kang et al., 2025) selects the first extractable final answer from the N sampled outputs, following the generation order of the LLM.
- **Perplexity** (Hu et al., 2024) selects the candidate with the lowest average negative log-likelihood under the LLM.
- **Self-consistency** (Wang et al., 2022) samples multiple reasoning paths and selects the most frequent final answer.
- **Self-certainty** (Kang et al., 2025) selects the answer with the highest estimated confidence among multiple sampled outputs.

The multi-agent baselines cover Debate(Du et al., 2023), STaR(Zelikman et al., 2024), Multi-agent FT(Subramaniam et al., 2025), which exploit multi-agent collaboration.

- **Debate** (Du et al., 2023) proposes a interaction-based multi-agent debate framework, where agents iteratively exchange arguments and revise their responses through structured discussion.
- **STaR** (Zelikman et al., 2024) iteratively fine-tunes the LLM using ground-truth answers, adding correctly solved instances to the training set and re-prompting incorrect ones with ground-truth–derived hints until convergence.
- **Multi-agent FT** (Subramaniam et al., 2025) enables self-improvement by fine-tuning LLMs on diverse multi-agent–generated reasoning traces.

We repeat each experiment five times with different random seeds and report the mean results.

4.2 Overall Performance

To verify the effectiveness of CG-MAV, as described in Section 3, we present the performance comparison. The results in Table 2 demonstrate the effectiveness of CG-MAV. From the results shown in Table 2, we can draw the following findings:

We first compare CG-MAV with widely adopted single model baselines, where CG-MAV consistently nearly outperforms all baselines across datasets. On GSM8K, CG-MAV slightly underperforms self-certainty when $N = 8$, which can be attributed to the limited sampling budget. Despite differences in sampling and aggregate strategies, these methods share a key limitation: they treat all tasks uniformly and lack an explicit mechanism to identify easy or hard task that require additional verification. As a result, confidence signals are applied uniformly, which can be unreliable when the LLM exhibits high uncertainty. In contrast, CG-MAV introduces a threshold based splitting strategy that fundamentally changes how these signals are utilized. Rather than uniformly aggregation all tasks, we partition inputs into easy and hard subsets based on Self-certainty. This mechanism yields two advantages. First, it prevents hard tasks from being overwhelmed by noisy majority signals. Second, it ensures that verification capacity is allocated precisely where it is most beneficial.

We further compare our approach with representative multi-agent baselines, and CG-MAV consistently outperforms them across datasets. These methods improve performance via multi-agent interaction or iterative fine-tuning, by coordinating reasoning and generation across multiple agents or by relying on additional training data. While effective in certain settings, such designs typically place substantial complexity on individual agents, which limit scalability and robustness. CG-MAV differs fundamentally from these approaches in that multi-agent capacity is used exclusively in verification for hard tasks, not generation or learning. Each agent is assigned a distinctive perspective to evaluate candidate answers and identify flaws. By removing the burden of generation and interaction, agents can focus on distinct error detection. This functional separation significantly reduces task complexity for each agent. Quantitatively, CG-MAV outperforms multi-agent baselines with large margins on tasks. Compared to STaR and Multi-agent FT, our approach achieves excellent performance without any additional training, highlighting its practicality and scalability

4.3 Ablation Studies

In this section, we analyze the contributions of SCBV-guided task classification and multi-agent verification. We consider three settings: (1) Vanilla CG-MAV, our framework; (2) Integration + MAV,

Type	Methods	MATH			GSM8K			Avg.
		N = 8	*N = 32*	*N = 64*	*N = 8*	*N = 32*	*N = 64*	
Single	Greedy	-	47.96	-	-	84.00	-	65.98
	FirstAns	49.08	49.08	49.09	82.08	82.08	82.08	65.70
	Perplexity	51.96	52.56	53.34	85.01	85.16	85.81	68.97
	Self-consistency	54.60	55.53	55.83	85.19	85.65	86.18	70.50
	Self-certainty	54.54	54.86	56.04	**85.27**	85.93	86.79	72.23
Multiple	Debate	-	51.62	-	-	78.44	-	65.03
	STaR	-	50.34	-	-	77.45	-	63.90
	Multi-agent FT	-	57.43	-	-	86.61	-	72.02
Ours	CG-MAV	**55.37**	**58.71**	**60.19**	85.04	**87.68**	**88.34**	**72.56**

Table 2: Performance Comparison. The best results are highlighted in **bold**. For baselines that do not require multiple sampling, we directly report the results from a single sample.

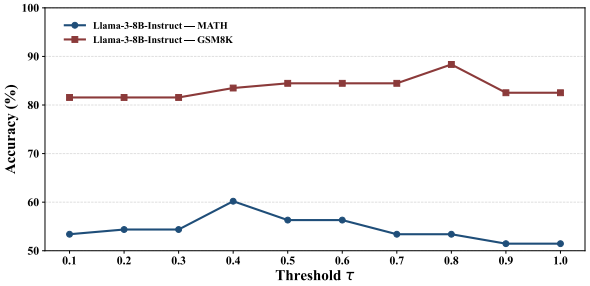


Figure 3: Performance of CG-MAV under different SCBV thresholds on the training sets of MATH and GSM8K, with an optimal threshold of 0.4 for MATH and 0.8 for GSM8K.

which removes task classification and uniformly applies multi-agent verification to all tasks; and (3) CG + Selection, which replaces multi-agent verification with single model selecting final answer from top-3 candidates ranked by SCBV. From the results presented in Table 3, the following observations can be drawn:

Overall, vanilla CG-MAV consistently outperforms both Integration + MAV and CG + Selection. Compared with Integration + MAV, CG-MAV achieves better performance by selectively invoking verification only on hard tasks. The result shows that blindly allocating multiple verification to easy tasks is unnecessary and harmful, whereas SCBV-guided task classification enables more efficient and reliable decision making.Compared with CG + Selection, CG-MAV shows clear advantages. The reason is that simply expanding the context of a single model is insufficient when SCBV in unreliable. In contrast, leveraging multi-agent verification provides diverse and complementary perspectives, leading to more robust judgments on hard tasks.

4.4 Hyperparameter Tuning

The SCBV threshold is treated as a hyperparameter and is tuned on the training set. Specifically, we perform a search over threshold values ranging from 0.1 to 1.0, with a step of 0.1, and select the value that yields the best accuracy on the training data. The selected threshold is then applied to the test set. Figure 3 illustrates that the optimal threshold is 0.4 on MATH, while a higher threshold of 0.8 yields the best performance on GSM8K. This is mainly because the model is more confident on the easier GSM8K dataset and therefore requires a higher threshold. On the more difficult MATH dataset, the model's confidence is generally lower, which results a lower threshold.

4.5 Cross- Moodel Generalization

In this section, we conduct experiments on a diverse set of LLMs to evaluate whether CG-MAV generalizes across different backbone LLMs, such as Qwen2.5-3B (Team, 2024), Qwen2.5-7B (Team, 2024), and Mistral-7B-Instruct-v0.2 (Chaplot, 2023). As shown in Table 4, our proposed CG-MAV consistently improves performance over the corresponding baseline, demonstrating reliable generalization across models.

4.6 SCBV as a Reliability Indicator

Figure 4 analyzes the distribution of SCBV scores over correctly answered instances. We observe a clear separation: instances with higher SCBV consistently exhibit a much higher proportion of correct answers, whereas low-certainty regions contain a large fraction of incorrect or unreliable predictions. This observation supports the use of SCBV as a reliability signal and motivates our SCBV-guided classification of easy and hard tasks.

Ablation Setting	MATH			GSM8K		
	N = 8	*N = 32*	*N = 64*	*N = 8*	*N = 32*	*N = 64*
Vanilla CG-MAV	**55.37**	**58.71**	**60.19**	**85.04**	**87.68**	**88.34**
Integration + MAV	48.95	50.03	51.19	73.67	77.46	78.65
CG + Selection	45.46	47.88	48.47	73.27	75.18	76.82

Table 3: The impact of SCBV-Guided Task Classification and Multi-Agent Verification.

Model	Method	MATH			GSM8K		
		N = 8	*N = 32*	*N = 64*	*N = 8*	*N = 32*	*N = 64*
Qwen2.5-3B	Self-certainty	67.39	70.17	72.83	81.23	84.37	85.98
	CG-MAV	**67.85**	**71.79**	**74.90**	**82.08**	**84.28**	**87.99**
Qwen2.5-7B	Self-certainty	71.46	74.88	76.54	82.16	86.37	88.03
	CG-MAV	**71.68**	**75.02**	**78.54**	**82.74**	**87.87**	**89.58**
Mistral-7B-Instruct-v0.2	Self-certainty	18.68	22.50	23.02	36.47	44.75	45.83
	CG-MAV	**18.70**	**23.17**	**25.13**	**36.75**	**46.33**	**48.56**

Table 4: Comparison with Self-certainty across different models. The best results are highlighted in **bold**.

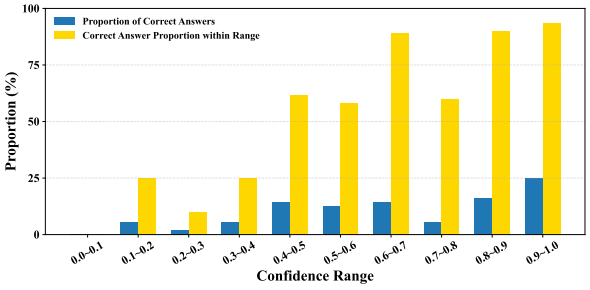


Figure 4: Distribution of Self-certainty scores and accuracy across confidence Ranges. Higher Self-certainty corresponds to a larger proportion of correct answers.

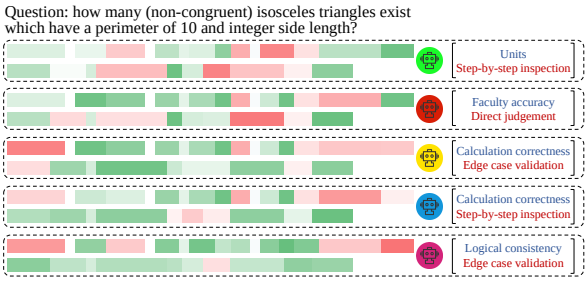


Figure 5: Attention heatmaps showing diverse focus across agents.

4.7 Perspective Attention Analysis

To better understand how different verification perspectives operate and whether they indeed attend to complementary information, we conduct an attention based qualitative analysis of the verification stage. Specifically, for each candidate solution, we visualize attention distributions to show which parts of the task and answer each verification agent focus on during verification.

Concretely, given a task x and a candidate answer y, we extract the token-level attention scores computed during the verification process of agents conditioned on (x, y). For each token in the problem and the generated answer, we compute the average attention weight assigned by the agent's output tokens, yielding a token-level attention matrix that we visualize as a heatmap for each verification perspective. We observe that different verification agents tend to focus on different regions of the task-answer pair. Figure 5 shows representative attention heatmap for the five verification perspectives, which illustrates this observation.

5 Conclusion

Selecting correct answers from multiple candidate solutions remains a fundamental challenge for test-time scaling methods, especially when plausible but incorrect solutions are generated. To solve this challenge, we propose a confidence-guided multi-agent verification framework for robust answer selection, named CG-MAV. in this paper. Specifically, CG-MAV leverages confidence not only as a selection signal but also as an indicator of task difficulty, enabling tasks to be divided into easy and hard. Easy tasks are solved through direct selection, while hard tasks are handled by multi-agent verification with distinct persona. Extensive experimental results on two reasoning datasets across multiple models show that CG-MAV consistently outperforms the baselines, which demonstrates CG-MAV's superiority and generalization.

6 Limitations

The main limitations of this paper are as follows: (1) The SCBV threshold for task classification is determined empirically through experiments on the train set, rather than derived from a principled or theoretically grounded criterion. (2) The CG-MAV introduces additional computational overhead in hard tasks, as multi-agent verification requires invoking multiple reasoning processes, which may limit scalability under strict latency or resource constraints. These limitations suggest promising directions for future research, such as reducing the reliance on empirically tuned confidence thresholds and developing more efficient verification strategies for hard tasks.

References

Saaket Agashe, Yue Fan, and Xin Eric Wang. 2023. Evaluating multi-agent coordination abilities in large language models.

Afra Amini, Tim Vieira, Elliott Ash, and Ryan Cotterell. 2024. Variational best-of-n alignment. *arXiv preprint arXiv:2407.06057*.

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

Devendra Singh Chaplot. 2023. 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, william el sayed. *arXiv preprint arXiv:2310.06825*, 3.

Justin Chen, Swarnadeep Saha, and Mohit Bansal. 2024. Reconcile: Round-table conference improves reasoning via consensus among diverse llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7066–7085.

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

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Peter Emerson. 2013. The original borda count and partial voting. *Social Choice and Welfare*, 40(2):353–358.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2024. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 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*.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. 2025. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*.

Yutong Hu, Quzhe Huang, Mingxu Tao, Chen Zhang, and Yansong Feng. 2024. Can perplexity reflect large language model's ability in long text understanding? *arXiv preprint arXiv:2405.06105*.

Zhewei Kang, Xuandong Zhao, and Dawn Song. 2025. Scalable best-of-n selection for large language models via self-certainty. *arXiv preprint arXiv:2502.18581*.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. 2024. Vineppo: Unlocking rl potential for llm reasoning through refined credit assignment.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. 2024. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Joshua Ong Jun Leang, Zheng Zhao, Aryo Pradipta Gema, Sohee Yang, Wai-Chung Kwan, Xuanli He, Wenda Li, Pasquale Minervini, Eleonora Giunchiglia, and Shay B Cohen. 2025. Picsar: Probabilistic confidence selection and ranking for reasoning chains. *arXiv preprint arXiv:2508.21787*.

Huao Li, Yu Chong, Simon Stepputtis, Joseph P Campbell, Dana Hughes, Charles Lewis, and Katia Sycara. 2023. Theory of mind for multi-agent collaboration via large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 180–192.

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, et al. 2024. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592*.

Chiyu Ma, Enpei Zhang, Yilun Zhao, Wenjun Liu, Yaning Jia, Peijun Qing, Lin Shi, Arman Cohan, Yujun Yan, and Soroush Vosoughi. 2025. Judging with many minds: Do more perspectives mean less prejudice? *arXiv preprint arXiv:2505.19477*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Ali Razghandi, Seyed Mohammad Hadi Hosseini, and Mahdieh Soleymani Baghshah. 2025. Cer: Confidence enhanced reasoning in llms. *arXiv preprint arXiv:2502.14634*.

Jie Ren, Yao Zhao, Tu Vu, Peter J Liu, and Balaji Lakshminarayanan. 2023. Self-evaluation improves selective generation in large language models. In *Proceedings on*, pages 49–64. PMLR.

Pier Giuseppe Sessa, Robert Dadashi, Léonard Hussenot, Johan Ferret, Nino Vieillard, Alexandre Ramé, Bobak Shariari, Sarah Perrin, Abe Friesen, Geoffrey Cideron, et al. 2024. Bond: Aligning llms with best-of-n distillation. *arXiv preprint arXiv:2407.14622*.

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

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

Vighnesh Subramaniam, Yilun Du, Joshua B Tenenbaum, Antonio Torralba, Shuang Li, and Igor Mordatch. 2025. Multiagent finetuning: Self improvement with diverse reasoning chains. *arXiv preprint arXiv:2501.05707*.

Hexiang Tan, Fei Sun, Sha Liu, Du Su, Qi Cao, Xin Chen, Jingang Wang, Xunliang Cai, Yuanzhuo Wang, Huawei Shen, and Xueqi Cheng. 2025. Too consistent to detect: A study of self-consistent errors in LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4755–4765, Suzhou, China. Association for Computational Linguistics.

Qwen Team. 2024. Qwen2. 5: A party of foundation models, september 2024. *URL https://qwenlm. github. io/blog/qwen2*, 5(4).

Danqing Wang, Zhuorui Ye, Fei Fang, and Lei Li. 2024a. Cooperative strategic planning enhances reasoning capabilities in large language models. *arXiv preprint arXiv:2410.20007*.

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*.

Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, et al. 2025. Thoughts are all over the place: On the underthinking of o1-like llms. *arXiv preprint arXiv:2501.18585*.

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

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D Goodman. 2024. Star: Self-taught reasoner bootstrapping reasoning with reasoning. In *Proc. the 36th International Conference on Neural Information Processing Systems*, volume 1126.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*.

A Experimental Details

A.1 The Details of CG-MAV

This appendix lists the prompts used for the five agents in CG-MAV.

Units Step-by-step inspection agent is designed to verify the correctness of unit throughout the reasoning process. It focus exclusively on the definition, transformation and combination of physical or abstract units at each step of the solution. By isolating unit consistency from numerical computation, this agent targets a common but often overlooked source of errors.

Prompt

You are a critical verifier tasked with evaluating mathematical problem. You will be presented with a question and a proposed solution. Your job is to carefully go over and analyze the solution. Follow the instructions.
INSTRUCTION:
Check whether units are handled correctly at each step of the solution. Focus only on unit definitions, transformations, and combinations as they appear in the reasoning steps. If you find any issues with the units, stop and reply 'FINAL VERRIFICATION ANSWER: False'. If all units are handled correctly, reply 'FINAL VERIFICATION ANSWER: True'.

Figure 6: Prompt of Units Step-by-step inspection agent.

Faculty accuracy Direct judgment agent simulates the holistic evaluation of an experienced faculty reviewer. Rather than decomposing the solution into isolated components, it applies direct judgment to assess whether the solution would be approved. If any conceptual flaw or reasoning gap is identified, the agent rejects the solution. The agent acts as a human-aligned verifier that determines whether the solution would be approved by a faculty reviewer.

Calculation correctness Edge case validation agent is responsible for testing the validity of the solution under extreme and boundary conditions. It systematically examines whether the calculations remain correct when inputs approach limiting values. This targeted validation complements standard calculation checking by identifying errors that only appear in extreme or boundary cases.

Prompt

You are a critical verifier tasked with evaluating mathematical problem. You will be presented with a question and a proposed solution. Your job is to carefully go over and analyze the solution. Follow the instructions.
INSTRUCTION:
Using direct faculty judgement, assess whether the solution is correct as a whole. Consider whether the reasoning and final answer align with standard expectations for a correct solution. If any part of solution would be judged incorrect from a faculty perspective, reply 'FINAL VERRIFICATION ANSWER: False'. If the solution would be accepted as correct by a faculty reviewer, reply 'FINAL VERIFICATION ANSWER: True'.

Figure 7: Prompt of Faculty accuracy Direct judgment agent.

Prompt

You are a critical verifier tasked with evaluating mathematical problem. You will be presented with a question and a proposed solution. Your job is to carefully go over and analyze the solution. Follow the instructions.
INSTRUCTION:
Check the calculation correctness of the solution under edge cases and boundary conditions. Examine whether calculation remains valid for extreme or special values. If any edge cases leads to incorrect calculations or invalid results, reply 'FINAL VERRIFICATION ANSWER: False'. If calculations are correct for all relevant edge cases, reply 'FINAL VERIFICATION ANSWER: True'.

Figure 8: Prompt of Calculation correctness Edge case validation agent.

Calculation correctness Step-by-step inspection agent conducts a step-by-step verification of all arithmetic and algebraic operations in the solution. It checks each intermediate computation for numerical accuracy and valid mathematical manipulation. By focusing on individual calculation steps, this agent provides fine-grained signals of numerical correctness that are independent of the overall reasoning structure or conceptual validity.

Prompt

You are a critical verifier tasked with evaluating mathematical problem. You will be presented with a question and a proposed solution. Your job is to carefully go over and analyze the solution. Follow the instructions.
INSTRUCTION:
Inspection the solution step by step, focusing on calculation correctness at each stage. Check for arithmetic mistakes, incorrect manipulations, or common calculation errors. If any step contains a calculation error, reply 'FINAL VERRIFICATION ANSWER: False'. If all calculations are correct step by step, reply 'FINAL VERIFICATION ANSWER: True'.

Figure 9: Prompt of Calculation correctness Step-by-step inspection agent.

Logical consistency Edge case validation agent evaluates the logical correctness of the solution by examining its consistency with domain knowl-

edge and established theories across all relevant cases. In particular, it focus on edge scenarios to determine whether the remains valid beyond typical settings. The agent checks whether the solution remains logically consistent and aligned with established domain knowledge, particularly under edge cases and extreme conditions.

Prompt

You are a critical verifier tasked with evaluating mathematical problem. You will be presented with a question and a proposed solution. Your job is to carefully go over and analyze the solution. Follow the instructions.
INSTRUCTION:
Check the logical consistency of the solution when applying domain knowledge and established theories for this type of task. Evaluate whether the reasoning remains logically consistent across all cases. If any logical inconsistency or misuse of domain knowledge if found, reply 'FINAL VERRIFICATION ANSWER: False'. If the solution is logically consistent in all cases, reply 'FINAL VERIFICATION ANSWER: True'.

Figure 10: Prompt of Logical consistency Edge case validation agent.