BELIEVE YOUR MODEL: DISTRIBUTION-GUIDED CONFIDENCE-AWARE VOTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Reasoning Models have demonstrated remarkable performance with the advancement of test-time scaling techniques, which enhance prediction accuracy by generating multiple candidate responses and selecting the most reliable answer. While prior work has shown that internal model signals such as confidence scores can partly indicate response correctness and exhibit a distributional correlation with accuracy, the relative structure of each query's confidence distribution has not been fully utilized to guide answer selection. Motivated by this, we propose *DistriVoting*, which explicitly exploits this per-query distributional structure during voting. Specifically, our method (1) decomposes the mixed confidence distribution into higher- and lower-mean candidate components using a Gaussian Mixture Model, and (2) applies a reject filter that uses the lower-mean component as counter-evidence to mitigate overlap between the two components. Besides, to further alleviate the overlap from the perspective of the distribution itself, we propose *SelfStepConf*, which uses step-level confidence to dynamically adjust the inference process, increasing the separation between the two components to improve the reliability of confidence in voting. Experiments across 16 models and 5 benchmarks demonstrate that our method outperforms state-of-the-art approaches.

1 INTRODUCTION

The advent of techniques like Chain of Thought (Wei et al., 2022; Yao et al., 2022) and Test-Time Scaling (TTS) (Zhang et al., 2025) has led to remarkable performance improvements in Large Reasoning Models (LRMs). However, even though TTS can generate multiple answers or increase token overhead for the same question by increasing test-time computation, enhancing LRMs' test-time performance remains a critical research direction. The core issue is the lack of label or reward signals during the test-time phase, which makes it difficult to evaluate the quality of generated answers and dynamically adjust the generation process. To address this problem, some approaches, like s1 (Muennighoff et al., 2025), enhance generation by extending the model's reasoning process without introducing any feedback signals. Other approaches, such as MoB (Rakhsha et al., 2025) and DORA (Wang et al., 2025d), utilize external reward models to score multiple generated results and aggregate them to obtain the final answer. In contrast, methods like Self-Consistency (Wang et al., 2022), BoN (Lightman et al., 2023), Self-Certainty (Kang et al., 2025), and DeepConf (Fu et al., 2025) have demonstrated that internal information can be effective enough to predict the quality of different answers without introducing additional models. This category of approach balances effectiveness and efficiency, and its benefits become more pronounced as the reliability of internal information improves.

Nevertheless, current work on applying internal information to TTS mainly focuses on designing better ways to obtain internal information (Geng et al., 2024), such as using sentence-level probabilistic measures like perplexity and average log-probability, or using distributional information like Kullback-Leibler divergence and entropy. We provide a more detailed introduction to related work in §A. While these studies have identified a distributional relationship between internal confidence and answer correctness, where correct and incorrect trajectories typically follow distinct statistical distributions, this observation has mainly been used to assess confidence reliability. We argue that the relative structure of the confidence scores sampled for each query can also guide confidence-based voting during answer selection.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

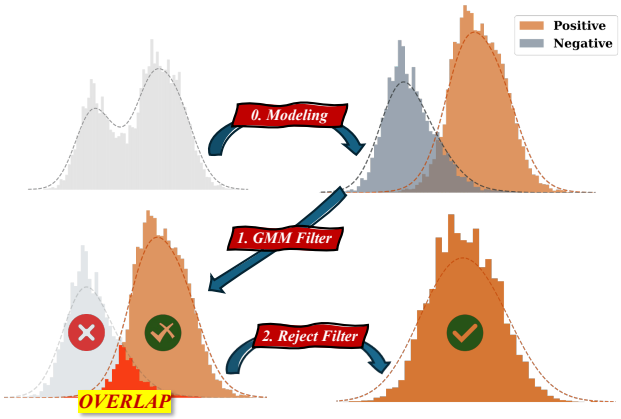


Figure 1: Overview of *DistriVoting*. We model the per-query confidence distribution using GMM, form higher- and lower-mean candidate pools, and use the lower-mean pool as counter-evidence to reject potential false positives before final voting.

Motivated by this, we propose *DistriVoting*, which enhances confidence-based voting through a two-step filtering process. Specifically, the process begins by modeling each query's confidence distribution as a mixture of two Gaussian components using a Gaussian Mixture Model (GMM). The higher- and lower-mean components provide candidate positive and negative pools rather than correctness labels, allowing likely negative trajectories to be filtered without external supervision. However, significant overlap can persist between the two components, primarily due to high-confidence incorrect samples and low-confidence correct samples, which may still introduce false positives during final voting. To address this issue, we show that the answer supported by the lower-mean candidate pool can serve as counter-evidence for rejecting potential false positives. Additionally, to further alleviate overlap from a distributional perspective, we propose *SelfStepConf*, which applies step-level confidence during trajectory generation to trigger self-reflection without external supervision.

To evaluate the effectiveness of *DistriVoting* and *SelfStepConf*, we conducted experiments using 16 different models, including DeepSeek-Series (Guo et al., 2025) and Qwen3-Series with thinking and non-thinking modes (Yang et al., 2025), on 5 reasoning benchmarks, including HMMT2025, BRUMO2025 (Balunović et al., 2025), GPQA-D (Rein et al., 2023), and AIME (AIME, 2025). The results show that our proposed *DistriVoting* consistently demonstrates performance gains across different models and benchmarks. Furthermore, *SelfStepConf* increases the average separation between the candidate components, further boosting these improvements.

2 Preliminaries

2.1 Confidence of Trajectory

Prior work shows that LLM token distributions $\mathbf{P}_i(j)$ can reveal uncertainty and trajectory quality. Following DeepConf (Fu et al., 2025), we compute confidence from token negative log-probabilities. For a trajectory of N generated tokens, we define

$$C_{\text{traj}} = -\frac{1}{N_G \times k} \sum_{i \in G} \sum_{j=1}^{k} \log \mathbf{P}_i(j), \tag{1}$$

where $\mathbf{P}_i(j)$ is the j-th largest next-token probability at position i, N_v is the vocabulary size, and $\mathbf{P} \in \mathbb{R}^{N \times N_v}$. The token group G is the reasoning step containing the final answer, $N_G = |G|$, and k is the number of top probabilities used at each position. Under this score, a more concentrated top-k distribution gives a larger value because its nonmaximal probabilities are smaller. Accordingly, we interpret a larger C_{traj} as indicating higher confidence.

2.2 Distribution of Confidence

Following the TTS setting, we repeatedly sample trajectories for each question. We use a two-component Gaussian mixture as a working approximation to the resulting confidence distribution,

with a higher-mean positive component and a lower-mean negative component:

$$X_{\text{pos}} \sim \mathcal{N}(\mu_{\text{pos}}, \sigma^2_{\text{pos}}), \quad X_{\text{neg}} \sim \mathcal{N}(\mu_{\text{neg}}, \sigma^2_{\text{neg}}), \tag{2}$$

where $\mu_{\text{pos}} > \mu_{\text{neg}}$. This approximation is fitted independently for each question and does not assume that every empirical distribution is exactly bimodal or Gaussian. We examine this modeling choice empirically in §5.4.

2.3 Distribution Separation and Voting Accuracy

As the distance between the positive and negative confidence distributions increases, the two-component approximation in §2.2 can model and distinguish the two distributions more accurately, thereby improving voting accuracy. Theorem 2.1 formalizes this separation through a midpoint tail-mass ratio, while Theorem 2.2 connects the same mean gap to a lower bound on multi-answer weighted-voting accuracy. Proofs are provided in §B.

Theorem 2.1. *Let f and g be the densities of $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$, respectively, where $\mu_1 > \mu_2$ and $\sigma_1, \sigma_2 > 0$. Define*

$$a = \frac{\mu_1 + \mu_2}{2}, \qquad \delta = \mu_1 - \mu_2, \qquad R(\delta) = \frac{\int_a^\infty f(x)\,dx}{\int_a^\infty g(x)\,dx}.$$

For fixed σ_1 and σ_2, $R(\delta)$ is strictly increasing for $\delta > 0$.

Theorem 2.2. *Under the conditions of Theorem 2.1, let $\{w_i\}_{i=1}^n$ be non-negative weights of all n samples. Let $\mathcal{X}^0$ contain the confidence scores associated with the correct answer, each drawn from f, and let $\mathcal{X}^k$ contain those associated with the k-th incorrect answer, each drawn from g, for $k = 1, \ldots, m$ and $m \le n - 1$. Assume all scores are mutually independent. The weighted-voting accuracy*

$$P_{\text{vote}}(\delta) = \mathbb{P}\left(\sum_{x_i \in \mathcal{X}^0} w_i x_i > \max_{1 \le k \le m} \sum_{x_j \in \mathcal{X}^k} w_j x_j \right)$$

has a lower bound that is strictly increasing in $\delta = \mu_1 - \mu_2$ when the variances, weights, and μ_2 are fixed and the total weight assigned to $\mathcal{X}^0$ is positive.

Together, the results motivate assigning the higher-mean GMM component to the positive pool and applying confidence-weighted voting in §3.2. We evaluate the effect of SSC on empirical separation and downstream voting in §5.2.

3 Methodology

3.1 Overview

DistriVoting is a distribution-guided voting pipeline that operates on multiple trajectories sampled for the same query. It first models their relative confidence structure with a two-component GMM, forms higher- and lower-mean candidate pools, uses the lower-mean pool as counter-evidence to reject a potentially incorrect answer, and then performs final voting on the refitted higher-mean pool. Both rejection and final aggregation use a generic base voting operator. We additionally propose *SelfStepConf* (SSC) as an optional generation-stage enhancement that monitors step-wise confidence and triggers self-reflection when confidence declines significantly. Although SSC operates before voting chronologically, we first present the central *DistriVoting* pipeline and then describe how SSC can improve its input trajectories.

3.2 Distribution Voting

GMM Modeling. After inference, for each query's Budget trajectories $\mathcal{V}$, we compute trajectory-level confidence $\mathcal{C}$ via Eq 1. Building on the working approximation in §2.2, we model these unlabeled confidence scores using a two-component GMM:

$$p(x) = \pi_1 \mathcal{N}(x \mid \mu_1, \sigma_1^2) + \pi_2 \mathcal{N}(x \mid \mu_2, \sigma_2^2), \tag{3}$$

where $\pi_1, \pi_2 > 0$, $\pi_1 + \pi_2 = 1$, and $\mathcal{N}(x \mid \mu_i, \sigma_i^2)$ is the density of component i. The components capture relative structure among trajectories for the same query. They are not correctness labels, and correctness annotations are not used during fitting.

GMM Filter. We treat the larger-mean component as the candidate positive pool and the other as the candidate negative pool. Let

$$i_{\text{pos}} = \arg\max_{i\in\{1,2\}} \mu_i, \qquad i_{\text{neg}} = \arg\min_{i\in\{1,2\}} \mu_i. \tag{4}$$

For trajectory V_r with confidence C_r, its posterior responsibility for component i is

$$\gamma_{r,i} = \frac{\pi_i \mathcal{N}(C_r \mid \mu_i, \sigma_i^2)}{\sum_{\ell=1}^{2} \pi_\ell \mathcal{N}(C_r \mid \mu_\ell, \sigma_\ell^2)}. \tag{5}$$

We assign each trajectory to the component with larger posterior responsibility:

$$\mathcal{V}_{\text{pos}} = \{V_r \in \mathcal{V} \mid \gamma_{r,i_{\text{pos}}} \geq \gamma_{r,i_{\text{neg}}}\}, \quad \mathcal{V}_{\text{neg}} = \mathcal{V} \setminus \mathcal{V}_{\text{pos}}, \tag{6}$$

with corresponding confidence sets $\mathcal{C}_{\text{pos}}$ and $\mathcal{C}_{\text{neg}}$.

Reject Filter. Because the fitted components can overlap, high-confidence incorrect trajectories may remain in $\mathcal{V}_{\text{pos}}$. We therefore use $\mathcal{V}_{\text{neg}}$ as counter-evidence rather than simply discarding it. Using a generic base voting operator f_{vote}, we obtain

$$A_{\text{pos}} = f_{\text{vote}}(\mathcal{V}_{\text{pos}}, \mathcal{C}_{\text{pos}}), \qquad A_{\text{neg}} = f_{\text{vote}}(\mathcal{V}_{\text{neg}}, -\mathcal{C}_{\text{neg}}), \tag{7}$$

where the reversed confidence weights emphasize answers supported by lower-confidence trajectories. When $A_{\text{pos}} \neq A_{\text{neg}}$, this disagreement provides evidence that A_{neg} is a concentrated negative answer. We remove all trajectories that yield A_{neg} and refit the GMM because removing this answer changes the confidence mixture. When the two pools agree, the negative pool does not provide discriminative counter-evidence, so no rejection or refitting is performed:

$$\hat{\mathcal{V}}_{\text{pos}} = \begin{cases} \Phi_{\text{pos}}(\text{GMM}(\{C_r \mid V_r \in \mathcal{V},\ A_r \neq A_{\text{neg}}\})), & A_{\text{pos}} \neq A_{\text{neg}}, \\ \mathcal{V}_{\text{pos}}, & A_{\text{pos}} = A_{\text{neg}}, \end{cases} \tag{8}$$

where A_r is the answer extracted from V_r, and Φ_{pos} returns trajectories assigned by posterior responsibility to the refitted higher-mean component. This is an empirical counter-evidence mechanism rather than a correctness guarantee. If the lower-mean pool concentrates on the correct answer, rejection can be harmful, which motivates the agreement condition and our component-level evaluation.

Basic Voting. Both answer selection in Reject Filter and final aggregation use f_{vote}, allowing *DistriVoting* to work with different voting methods. In our implementation, we instantiate f_{vote} with Hierarchical Voting, which performs confidence-weighted voting within confidence intervals and then aggregates the interval winners. Its complete formulation is provided in §C.2. With the confidence set $\hat{\mathcal{C}}_{\text{pos}}$ corresponding to $\hat{\mathcal{V}}_{\text{pos}}$, the final answer is

$$A_{\text{final}} = f_{\text{vote}}(\hat{\mathcal{V}}_{\text{pos}}, \hat{\mathcal{C}}_{\text{pos}}). \tag{9}$$

Thus, *DistriVoting* follows a single sequence from per-query distribution modeling and component assignment to negative-answer rejection, refitting, and final voting.

3.3 SELFSTEPCONF

Reflection Trigger. SSC applies the same confidence signal used for trajectory-level voting at a finer step granularity during generation. For the generated i-th token, we calculate token confidence

$$C_{\text{token}-i} = -\frac{1}{k}\sum_{j=1}^{k} \log \mathbf{P}_i(j). \tag{10}$$

For the m-th reasoning step, the token group G_m is defined as

$$G_m = \{t_i \mid \text{index}(t_s^{m-1}) < i \leq \text{index}(t_s^m)\}, \quad m \in \mathbb{N}, \tag{11}$$

where t_s^m represents the m-th step token in the trajectory. The current step confidence is

$$C_{G_m} = -\frac{1}{N_{G_m} \times k} \sum_{i\in G_m} \sum_{j=1}^{k} \log \mathbf{P}_i(j). \tag{12}$$

We evaluate its relative change against the dynamically adaptive confidence threshold τ_{conf}:

$$\Delta_{\text{conf}} = \frac{C_{G_m}}{\tau_{\text{conf}}}, \quad m \geq 1. \tag{13}$$

If $\Delta_{\text{conf}} < \delta$, where δ is a control parameter, the system treats the current step as low quality without updating τ_{conf}. To maintain generation stability, reflection activates only when confidence is also declining:

$$\mathbb{I}_R = \mathbb{I}[(\Delta_{\text{conf}} < \delta) \wedge (C_{G_m} < C_{G_{m-1}})]. \tag{14}$$

The corresponding EMA update is

$$\tau_{\text{conf}} := \begin{cases} C_{G_m}, & m = 0, \\ \tau_{\text{conf}}, & m > 0 \wedge \mathbb{I}_R = 1, \\ \alpha\tau_{\text{conf}} + (1-\alpha)C_{G_m}, & m > 0 \wedge \mathbb{I}_R = 0. \end{cases} \tag{15}$$

Reflection Injection. We define a reflection token list $T_r = [t_r^1, t_r^2, \ldots, t_r^{N_r}]$. In our experiments, T_r is produced by tokenizing `wait`, and $N_r = |T_r|$. When the trigger activates, we swap the highest-probability token's probability with that of the corresponding reflection token and sample at temperature 0 for N_r steps:

$$\begin{aligned} \mathbf{P}_{r_i} &= [p_1, p_2, \ldots, p_k^*, \ldots, p_i, \ldots, p_{N_v}], \\ \mathbf{P}'_{r_i} &= [p_1, p_2, \ldots, p_i, \ldots, p_k^*, \ldots, p_{N_v}], \end{aligned} \tag{16}$$

where $\mathbf{P}_{r_i}$ represents the logits distribution of the r_i-th token in the trajectory, p_k^* is its highest probability, and p_i is the probability of reflection token t_r^i. Because this swap permutes two probabilities, it does not change $C_{\text{token}-r_i}$. It does, however, alter the generated continuation, allowing SSC to reshape the trajectories subsequently processed by *DistriVoting*.

4 Experiment

4.1 Experimental Description

In this section, we primarily validate the effectiveness of *DistriVoting* and its optional generation-stage enhancement, *SelfStepConf* (SSC). For *DistriVoting*, we compare with existing test-time scaling voting methods, including Self-Consistency (SC) (Wang et al., 2022), BoN (Lightman et al., 2023), MoB (Rakhsha et al., 2025), Weighted-SC (WSC) (Li et al., 2022), and DeepConf (WSC-Top50) (Fu et al., 2025). We then separate complete-method comparisons from matched component attribution. We evaluate the *GMM Filter* against Top50 and analyze *Reject Filter* and *HierVoting* in §5.1, §F.4, and §G.1. For SSC, we evaluate its contribution to voting accuracy and analyze confidence separation, sampling behavior, and inference dynamics in §5.2, §5.5, and §5.6, with complete results in §E. Additionally, for the Qwen3 model series, we employ *thinking mode* by default, with asterisk-marked **models*** denoting *non-thinking* variants. All experiments use temperature $t = 0.6$ unless otherwise noted, except for non-thinking models, which use $t = 0.7$. Comparisons at a given budget match the number of sampled trajectories, rather than generated tokens or wall-clock cost. Detailed descriptions, pseudocode, supporting experiments, and parameter analyses are provided in §C, §D, §E, and §F.

4.2 Main Results

In our main experiments, we evaluate DeepSeek-R1-8B and Qwen3-32B on five reasoning benchmarks. Table 1 compares external baselines with three matched *DistriVoting* versions. The G, $G{+}D$, and $G{+}D{+}S$ variants progressively add *GMM Filter*, the remaining distribution-voting stages (*Reject Filter* and *Hierarchical Voting*), and SSC.

The matched gains show that *GMM Filter* contributes 1.89 and 0.57 points, the remaining *DistriVoting* stages add 0.31 and 0.20 points, and SSC adds 0.89 and 0.54 points. The complete variant reaches 77.84% and 76.53%, outperforming the strongest listed external baseline by 2.54 and 1.17 points. The Top50 comparison is analyzed in §5.3.

Table 1: Main results at budget 128 over 64 repetitions. G, D, and S denote *GMM Filter*, the remaining *DistriVoting* stages (*Reject Filter* and *HierVoting*), and SSC. The three *DistriVoting* rows progressively add these components. $\pm$ reports the variation range. Bold and underlined averages denote the best and second-best model-level results. Complete benchmark-level variants and component ablations are provided in Table 8, Table 9, Table 10, and Table 11.

Model	Method	HMMT2025	GPQA-D	AIME2024	AIME2025	BRUMO2025	Avg.
DeepSeek-R1-8B	SC	69.11±0.23	67.50±0.07	86.67±0.00	80.36±0.13	93.07±0.13	73.09±0.06
	WSC	69.69±0.18	67.65±0.04	86.67±0.00	80.78±0.13	93.33±0.00	73.30±0.03
	BoN	70.05±0.34	67.91±0.17	90.52±0.11	77.34±0.30	92.24±0.19	73.43±0.13
	MoB-Adaptive	76.98±0.76	68.70±0.23	89.95±0.84	84.53±0.37	93.33±0.39	75.30±0.33
	WSC-Top50	73.80±0.44	68.58±0.04	90.00±0.18	82.55±0.34	93.33±0.00	74.75±0.07
	DistriVoting^{G}	82.50±0.49	69.58±0.13	93.13±0.15	83.91±0.99	93.59±0.08	76.64±0.11
	DistriVoting^{G+D}	82.55±0.31	69.82±0.09	93.13±0.18	85.52±0.60	93.70±0.10	<u>76.95±0.10</u>
	DistriVoting^{G+D+S}	84.95±0.86	70.63±0.17	93.23±0.05	86.46±0.55	94.27±0.17	**77.84±0.28**
Qwen3-32B	SC	62.08±0.31	70.30±0.20	86.46±0.33	76.98±0.26	93.33±0.00	73.85±0.21
	WSC	62.24±0.39	70.55±0.14	86.88±0.67	77.08±0.10	93.33±0.00	74.07±0.16
	BoN	59.48±0.46	72.00±0.14	87.29±0.28	74.01±0.35	87.08±0.32	73.87±0.31
	MoB-Adaptive	63.85±0.31	72.08±0.53	88.44±0.38	78.13±0.40	92.66±0.13	75.36±0.04
	WSC-Top50	63.96±0.29	72.03±0.17	88.28±0.31	76.93±0.26	92.71±0.00	75.22±0.12
	DistriVoting^{G}	64.84±0.23	72.42±0.11	88.65±0.31	79.22±0.34	92.71±0.03	75.79±0.10
	DistriVoting^{G+D}	64.43±0.21	72.74±0.25	89.01±0.39	78.70±0.23	93.23±0.08	<u>75.99±0.09</u>
	DistriVoting^{G+D+S}	65.73±0.05	73.18±0.02	89.11±0.50	80.05±0.44	93.33±0.08	**76.53±0.08**

4.3 Ablation Study

GMM. *DistriVoting* uses a two-component GMM as a distribution-based splitting method under the working approximation in §2.2. We compare it with Top50, K-Means, and MeanShift in terms of candidate-pool quality (Acc, WAcc, and AUROC), final voting accuracy, trajectory-correctness prediction, and computational efficiency. Correctness labels are used only for post-hoc evaluation, not for fitting any splitting method at inference time.

Table 2: Ablation of distribution splitting on DeepSeek-R1-8B with 128 trajectories/query (64 repeats). Complete benchmark-level results are in Table 12. Acc, WAcc, and AUROC measure candidate-pool quality, while correctness labels are used only to evaluate Predict Acc after inference.

Method	Top50	K-Means	MeanShift	GMM
Acc (%)	74.47	75.29	<u>76.55</u>	**77.60**
WAcc (%)	74.61	75.32	<u>76.57</u>	**77.68**
AUROC (↑)	0.5117	<u>0.5204</u>	0.5158	**0.5831**
Voting Acc (%)	75.10	75.19	<u>75.50</u>	**76.95**
Predict Acc (%)	53.64	56.19	<u>59.34</u>	**60.46**
Predict Time (ms/it)	**0.0935**	0.6014	1.8492	<u>0.3369</u>

As shown in Table 2, GMM obtains the best Acc, WAcc, AUROC, Predict Acc, and Voting Acc. It is $1.78\times$ as fast as K-Means and $5.49\times$ as fast as MeanShift, although the fixed Top50 operation remains cheaper. K-Means and MeanShift also outperform Top50 in voting accuracy, supporting adaptive distribution splitting rather than a fixed retention ratio. Complete benchmark-level results are provided in §G.2.

Budget. Budget is crucial for voting methods. We evaluate six settings (8, 16, 32, 64, 128, and 256) to study how the complete distribution-guided pipeline scales with the number of sampled trajectories. As shown in Table 3, *DistriVoting*+SSC improves from 73.18% at budget 8 to 78.18% at budget 256. Its advantage over the external WSC-Top50 baseline grows from 0.01 to 3.39 points, making the end-to-end scaling trend explicit rather than mixing it with every internal variant.

At budget 8, the two methods are nearly tied, consistent with noisier distribution estimates from small sample sets. The advantage grows steadily from 0.88 points at budget 16 to 3.39 points at budget 256. Complete method variants and benchmark-level results are provided in Table 13.

Table 3: Budget scaling using DeepSeek-R1-8B with B trajectories/query (64 repeats). The complete pipeline is compared with WSC-Top50, and Δ reports their difference. Complete method variants and benchmark-level results are in Table 13.

Method	8	16	32	64	128	256
WSC-Top50	73.17	73.86	74.56	74.62	74.75	74.79
DistriVoting^{G+D+S}	73.18	74.74	75.90	76.87	77.84	78.18
Complete-pipeline Δ	+0.01	+0.88	+1.34	+2.25	+3.09	+3.39

5 Analysis

5.1 The Effectiveness of DistriVoting

The overall gains in Table 1 do not reveal which filtering stage improves the candidate pool. We therefore measure the effect of *GMM Filter* and *Reject Filter* using $\mathrm{Acc} = |\mathcal{V}|^{-1} \sum_{r \in \mathcal{V}} \mathbb{I}(A_r = A_{\mathrm{gt}})$ and $\mathrm{WAcc} = |\mathcal{V}|^{-1} \sum_{r \in \mathcal{V}} \mathbb{I}(A_r = A_{\mathrm{gt}}) C_r$ across three stages: **(I)** before filtering (all samples), **(II)** after *GMM Filter* (candidate positive samples), and **(III)** after *Reject Filter* (final retained samples).

Table 4: Analysis of *DistriVoting* using DeepSeek-R1-8B, sampling 128 trajectories/query (64 repeats). Each cell shows **Acc / WAcc** (%). Complete results provided in Table 14 of §G.4.

Stage	HMMT2025	GPQA-D	AIME2024	AIME2025	BRUMO2025	Avg.
I	60.36 / 61.59	64.54 / 64.73	86.83 / 87.40	79.92 / 80.70	81.22 / 82.31	69.27 / 69.74
II	76.71 / 76.92	71.69 / 71.72	93.90 / 94.00	87.79 / 87.86	91.06 / 91.24	77.60 / 77.68
III	**77.43 / 77.63**	**75.86 / 75.89**	**94.08 / 94.17**	**88.75 / 88.83**	**91.34 / 91.52**	**80.41 / 80.48**

As shown in Table 4, both Acc and WAcc increase progressively across the three stages for DeepSeek-R1-8B, indicating that ***GMM Filter improves candidate-pool quality and Reject Filter further improves the retained pool.*** The complete results in Table 14 show the same model-average trend for all five evaluated configurations.

5.2 SSC's Role in Enhancing Voting Effectiveness

SSC improves voting accuracy, but aggregate scores alone do not explain how reflection changes the sampled confidence distribution. We therefore compare *SelfStepConf* and *BasicInference* using 512 trajectories per HMMT2025 query, with correctness used only for post-hoc visualization. As shown in Figure 2 (Up), SSC produces greater separation with less overlap in this example.

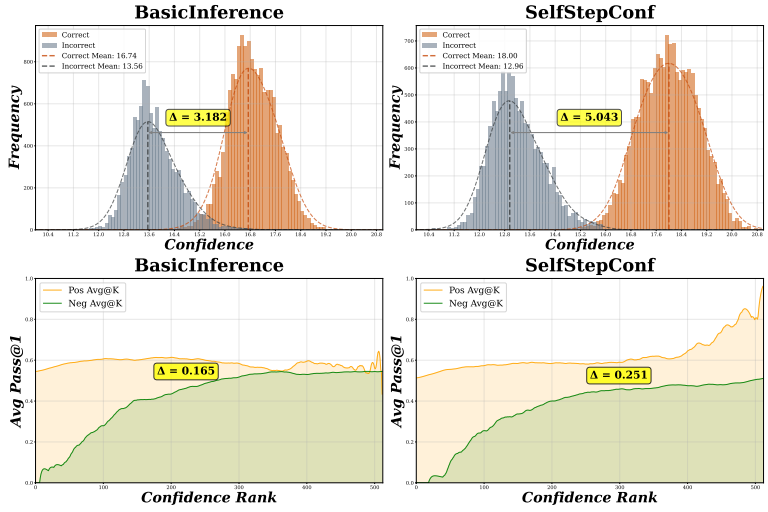


Figure 2: Confidence distributions and component correctness curves on HMMT2025 using DeepSeek-R1-8B with 512 trajectories/query. Complete results are in Table 15 of §G.5.

The results in §2.3 provide a theoretical motivation under the Gaussian assumptions: increasing the mean gap improves the midpoint tail ratio and the probability that one weighted Gaussian score exceeds another. They do not establish that SSC always enlarges the gap or that multiclass voting accuracy must increase.

As shown in Figure 2 (Down), we further calculate average correctness rates for confidence-sorted trajectories in the higher- and lower-mean components. Their difference reflects the correctness-density advantage of the higher-mean component and the reliability of the information available to *Reject Filter*. For this example, SSC increases the difference from 0.165 to 0.251. Across the five model configurations in Table 15, SSC improves model-average AUROC by 4.66%–13.60% and WAcc by 0.87%–5.52%, although individual benchmark exceptions occur. ***These results support an average separation improvement rather than a per-query guarantee.***

5.3 Adaptive Threshold Selection vs. Top-50 Filtering

A fixed retention ratio is reliable only when useful confidence cutoffs remain similar across tasks. To test this assumption and motivate per-question adaptation, we follow DeepConf's top-threshold definition and sweep WSC thresholds at 1% intervals across models and benchmarks.

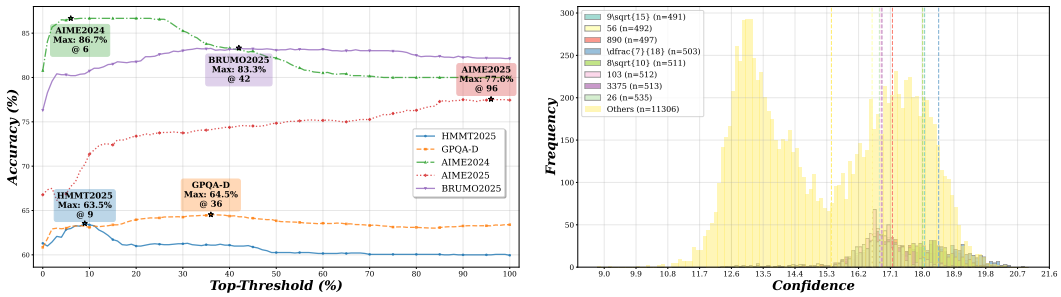


Figure 3: **Left:** Optimal thresholds for Qwen3-8B with 256 responses/query (64 repeats). **Right:** Answer-conditioned confidence distributions for DeepSeek-R1-8B with 512 responses/query on HMMT2025. Complete results are in Figure 13 and Figure 14.

As shown in Figure 3 (Left), ***the optimal thresholds vary substantially across benchmarks (6%, 9%, 36%, 42%, and 96%) and models.*** We therefore use 50% as the representative baseline for fixed-filter voting methods. In contrast, GMM partitions each question's sampled confidence distribution rather than applying a single benchmark-level threshold. Supplementary traversal results in §F.4 compare this question-level adaptation with the best benchmark-level fixed thresholds.

5.4 Interpretation of Gaussian Components in GMM

The two GMM components are intended to partition trajectory quality, but they could instead reflect different answer identities. To examine this alternative, we group trajectories by answer and visualize the confidence distributions of the eight most frequent answers in HMMT2025. Correctness is used only to evaluate the resulting pools, not to fit GMM at inference time, so the higher- and lower-mean components remain candidate positive and negative pools rather than guaranteed correct and incorrect sets. As shown in Figure 3 (Right), answer-conditioned distributions overlap substantially, and the means of the two most frequent answers differ by only 0.022. ***This overlap motivates confidence-based quality partitioning rather than assigning one Gaussian component to each answer.***

5.5 SSC's Impact on Model Sampling Behavior

Accuracy gains alone do not show whether SSC improves finite-budget sampling or merely changes generation cost. We therefore examine model-level Pass@1, Pass@K curves, and per-benchmark trajectory statistics from *SelfStepConf* and *BasicInference* (unintervened Next-Token Prediction).

At $K = 1$, SSC improves all four models in Table 17. The Pass@K curves in Figure 7 narrow as K increases, ***supporting improved finite-budget sampling efficiency rather than a changed reasoning***

Table 5: DeepSeek-R1-8B generation statistics at temperature zero with one response/query. Values are query averages, and time is per iteration. Complete results are in Table 18.

Benchmark	*BasicInference*			*SelfStepConf*		
	Step	Token	Conf.	Step	Token	Conf.
HMMT2025	154.00	28266.73	16.98	154.40	28604.80	17.20
GPQA-D	30.80	9560.65	13.20	29.27	9411.71	13.33
AIME2024	88.63	21239.20	17.52	83.33	20733.97	17.92
AIME2025	123.23	26673.87	17.47	128.10	26280.73	17.91
BRUMO2025	135.40	23137.50	17.21	124.50	22291.60	17.45
Avg.	66.47	15322.42	14.74	64.48	15097.02	14.95
Time (ms/it)		207.70			212.51	

limit. In Table 5, DeepSeek-R1-8B achieves improved confidence without compromising reasoning efficiency, requiring fewer average steps and tokens, while incurring only a marginal overhead in per-iteration latency (207.70 ms vs. 212.51 ms). Model-level Pass@1 results are in Figure 8.

5.6 SelfStepConf Effects on Inference Dynamics

Aggregate statistics do not show when reflection alters an individual reasoning trajectory. We therefore use **temperature 0** to align SSC and BasicInference before intervention, then record step confidence, token counts, and reflection positions. In Figure 4, triggers at steps 21 and 55 cause divergence.

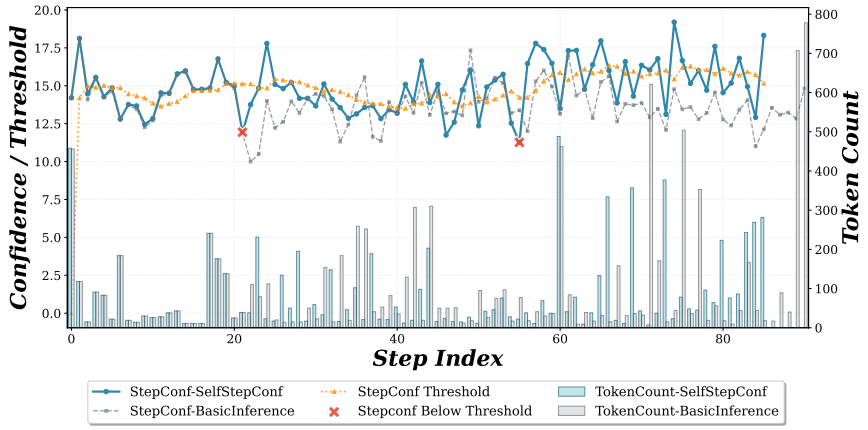


Figure 4: SSC and BasicInference confidence and token counts on one DeepSeek-R1-8B trajectory (GPQA-D question 188, temperature zero). Dashed lines mark reflection triggers.

After the second reflection, SSC reaches the correct answer in 86 steps and 7,750 tokens, versus 91 steps and 9,472 tokens for BasicInference. ***This case illustrates how reflection changes an individual trajectory rather than demonstrating uniform savings.*** Table 5 reports mixed aggregate costs, with complete results in Table 18.

6 Conclusion

This paper addresses confidently wrong predictions when model-inherent confidence is used for test-time-scaling voting. We propose *DistriVoting*, which exploits the relative structure of each query's confidence distribution: *GMM Filter* identifies candidate pools, *Reject Filter* uses the lower-mean pool as counter-evidence, and *HierVoting* performs the final aggregation. *SelfStepConf* (SSC) optionally adjusts generation and improves average confidence-distribution separation. Across 16 models and five benchmarks, the complete method consistently improves voting accuracy. On the two main models, it exceeds the strongest listed external baseline by 2.54 and 1.17 points. Matched ablations show that adaptive filtering provides the largest individual gain, while rejection, hierarchical aggregation, and SSC provide complementary refinements. Together, these results demonstrate the effectiveness of using per-query confidence-distribution structure for test-time voting without external supervision across the evaluated reasoning settings.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing and improving the clarity of written text. We have not used generative AI tools for generating research ideas, designing experiments, producing results, or writing code, and data collection and analysis are not applicable to AI-assisted tasks in this work. We have reviewed all AI-assisted writing to ensure accuracy and consistency with our intended meaning. We take responsibility for the final content of this work, including all text, claims, and artifacts presented herein.

ETHICS STATEMENT

This work does not raise any ethical concerns to the best of our knowledge. Our research involves no human subjects, no sensitive personal data, and no potentially harmful applications or methodologies. All datasets used in our experiments are publicly available and open-source, and their use complies with the respective licensing agreements. There are no conflicts of interest or sponsorship issues to disclose. We confirm that this work adheres to the ICLR Code of Ethics in all aspects.

REPRODUCIBILITY STATEMENT

We have made significant efforts to ensure the reproducibility of all results reported in this paper. Specifically: (1) **Experiments**: all experiments were repeated multiple times with different random seeds, and we report mean and standard deviation across runs to reflect statistical reliability, as detailed in the experimental setup section and the appendix; (2) **Datasets**: all datasets used in our experiments are fully open-source and publicly available, with complete descriptions of data preprocessing and splitting procedures provided in the appendix; (3) **Code**: the full implementation, including model architectures, training configurations, and evaluation scripts, will be made publicly available upon publication of this paper. We believe these measures collectively provide a solid foundation for the community to verify and build upon our findings.

REFERENCES

Aradhye Agarwal, Ayan Sengupta, and Tanmoy Chakraborty. The art of scaling test-time compute for large language models. *arXiv preprint arXiv:2512.02008*, 2025.

AIME. AIME problems and solutions, 2025. URL https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions, February 2025. URL https://matharena.ai/.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

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

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 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)*, pp. 6577–6595, 2024.

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

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. Tree search for llm agent reinforcement learning. *arXiv preprint arXiv:2509.21240*, 2025.

Zhewei Kang, Xuandong Zhao, and Dawn Song. Scalable best-of-n selection for large language models via self-certainty, 2025. *URL https://arxiv. org/abs/2502.18581*, 2025.

Aayush Karan and Yilun Du. Reasoning with sampling: Your base model is smarter than you think. *arXiv preprint arXiv:2510.14901*, 2025.

Yi-Chen Li, Tian Xu, Yang Yu, Xuqin Zhang, Xiong-Hui Chen, Zhongxiang Ling, Ningjing Chao, Lei Yuan, and Zhi-Hua Zhou. Generalist reward models: Found inside large language models. *arXiv preprint arXiv:2506.23235*, 2025.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. Making large language models better reasoners with step-aware verifier. *arXiv preprint arXiv:2206.02336*, 2022.

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

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025.

Bhuvanashree Murugadoss, Christian Poelitz, Ian Drosos, Vu Le, Nick McKenna, Carina Suzana Negreanu, Chris Parnin, and Advait Sarkar. Evaluating the evaluator: Measuring llms' adherence to task evaluation instructions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

Amin Rakhsha, Kanika Madan, Tianyu Zhang, Amir-massoud Farahmand, and Amir Khasahmadi. Majority of the bests: Improving best-of-n via bootstrapping. *arXiv preprint arXiv:2511.18630*, 2025.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. GPQA: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Linwei Tao, Yi-Fan Yeh, Minjing Dong, Tao Huang, Philip Torr, and Chang Xu. Revisiting uncertainty estimation and calibration of large language models. *arXiv preprint arXiv:2505.23854*, 2025.

Hong Wang, Zhezheng Hao, Jian Luo, Chenxing Wei, Yao Shu, Lei Liu, Qiang Lin, Hande Dong, and Jiawei Chen. Scheduling your llm reinforcement learning with reasoning trees. *arXiv preprint arXiv:2510.24832*, 2025a.

Kaiwen Wang, Jin Peng Zhou, Jonathan Chang, Zhaolin Gao, Nathan Kallus, Kianté Brantley, and Wen Sun. Value-guided search for efficient chain-of-thought reasoning. *arXiv preprint arXiv:2505.17373*, 2025b.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025c.

Xinglin Wang, Yiwei Li, Shaoxiong Feng, Peiwen Yuan, Yueqi Zhang, Jiayi Shi, Chuyi Tan, Boyuan Pan, Yao Hu, and Kan Li. Every rollout counts: Optimal resource allocation for efficient test-time scaling. *arXiv preprint arXiv:2506.15707*, 2025d.

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

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

Canhui Wu, Qiong Cao, Chang Li, Zhenfang Wang, Chao Xue, Yuwei Fan, Wei Xi, and Xiaodong He. Beyond token length: Step pruner for efficient and accurate reasoning in large language models. *arXiv preprint arXiv:2510.03805*, 2025.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*, 2023.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024a.

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

Ruixin Yang, Dheeraj Rajagopal, Shirley Anugrah Hayati, Bin Hu, and Dongyeop Kang. Confidence calibration and rationalization for llms via multi-agent deliberation. *arXiv preprint arXiv:2404.09127*, 2024b.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022.

Yuxuan Yao, Han Wu, Zhijiang Guo, Biyan Zhou, Jiahui Gao, Sichun Luo, Hanxu Hou, Xiaojin Fu, and Linqi Song. Learning from correctness without prompting makes llm efficient reasoner. *arXiv preprint arXiv:2403.19094*, 2024.

Jiahao Yu, Zelei Cheng, Xian Wu, and Xinyu Xing. Gpo: Learning from critical steps to improve llm reasoning. *arXiv preprint arXiv:2509.16456*, 2025.

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

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, et al. A survey on test-time scaling in large language models: What, how, where, and how well? *arXiv preprint arXiv:2503.24235*, 2025.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, et al. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*, 2025.

A Related Works

Test-Time Scaling (TTS). Current TTS research primarily focuses on increasing the inference depth of single sampling or improving information utilization among multiple sampling results to enhance the accuracy of final answers (Karan & Du, 2025; Agarwal et al., 2025; Wang et al., 2025b). The s1 (Muennighoff et al., 2025) study investigated the scaling relationship between response length and accuracy in single sampling results, using *BudgetForcing* to specify upper and lower bounds for thinking tokens, or forcibly replacing end tokens with wait tokens to extend inference length. Self-Consistency (Wang et al., 2022) leverages consistency information among multiple repeated sampling results to select the final consensus answer. Building upon this foundation, works such as MoB (Rakhsha et al., 2025) and DORA (Wang et al., 2025d) further utilize Reward Models to provide quality assessments for each sampling result, thereby refining the weights among different results and optimizing the voting methods through this weight information. In comparison, while our work similarly leverages weight information among different answers to improve final answer accuracy, we employ model-intrinsic information for **self-weighted**. Furthermore, we not only utilize this verification information to optimize voting schemes but also leverage it to dynamically adjust the model's generation process, achieving **self-enhancement** through self-weighted.

Intrinsic Information of LLMs. The widespread application of RL has enhanced the reasoning capabilities of LLMs, leading to the emergence of Large Reasoning Models (LRMs) (Liu et al., 2024). An increasing number of studies suggest that intrinsic information generated during the generation process of LRMs can, to some extent, reflect the quality of model responses (Tao et al., 2025). This information can be utilized not only in the training process to further extend the application scenarios of RLVR (Li et al., 2025; Zuo et al., 2025), but also in TTS to evaluate the quality of multiple answers and leverage this quality assessment to achieve serial, parallel, and tree-based answer search optimization. Average Log-Probability and Perplexity evaluate model response confidence by utilizing sentence-level probability information (Duan et al., 2024; Murugadoss et al., 2025; Yao et al., 2024; Yang et al., 2024b; Xiong et al., 2023). Entropy and Self-Certainty (Kang et al., 2025) further employ distribution-level confidence to assess relative confidence among multiple answers. DeepConf (Fu et al., 2025) already applies confidence-weighted voting after retaining a fixed top-50% subset. Our distinction is therefore not the first use of confidence in voting, but the combination of a per-query two-component GMM partition, negative-pool rejection, and SSC intervention instead of a fixed global retention fraction.

B THEOREM PROOF

B.1 PROOF OF THEOREM 2.1

Proof. Without loss of generality, assume $\mu_1 > \mu_2$, i.e., $\delta = \mu_1 - \mu_2 \geq 0$.

Let $\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$ be the cumulative distribution function of the standard normal distribution. For a normal distribution $N(\mu, \sigma^2)$, we have:

$$\int_a^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx = 1 - \Phi\left(\frac{a-\mu}{\sigma}\right). \tag{17}$$

Therefore:

$$\begin{aligned} I_1(\delta) &= \int_{\frac{\mu_1+\mu_2}{2}}^{\infty} f(x)dx = 1 - \Phi\left(\frac{\frac{\mu_1+\mu_2}{2} - \mu_1}{\sigma_1}\right) = 1 - \Phi\left(\frac{\mu_2 - \mu_1}{2\sigma_1}\right) \\ &= 1 - \Phi\left(-\frac{\delta}{2\sigma_1}\right) = 1 - \left(1 - \Phi\left(\frac{\delta}{2\sigma_1}\right)\right) = \Phi\left(\frac{\delta}{2\sigma_1}\right), \end{aligned} \tag{18}$$

$$\begin{aligned} I_2(\delta) &= \int_{\frac{\mu_1+\mu_2}{2}}^{\infty} g(x)dx = 1 - \Phi\left(\frac{\frac{\mu_1+\mu_2}{2} - \mu_2}{\sigma_2}\right) = 1 - \Phi\left(\frac{\mu_1 - \mu_2}{2\sigma_2}\right) \\ &= 1 - \Phi\left(\frac{\delta}{2\sigma_2}\right). \end{aligned} \tag{19}$$

The ratio function can be expressed as:

$$R(\delta) = \frac{\Phi\left(\frac{\delta}{2\sigma_1}\right)}{1 - \Phi\left(\frac{\delta}{2\sigma_2}\right)}. \tag{20}$$

Let $u = \frac{\delta}{2\sigma_1}$, $v = \frac{\delta}{2\sigma_2}$, and $\phi(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2}$ be the probability density function of the standard normal distribution. Then:

$$\frac{dR}{d\delta} = \frac{1}{(1 - \Phi(v))^2} \left[\frac{\phi(u)(1 - \Phi(v))}{2\sigma_1} + \frac{\Phi(u)\phi(v)}{2\sigma_2}\right]. \tag{21}$$

Since $\phi(z) > 0$ for all real z, $1 - \Phi(v) > 0$ when v is finite, and $\Phi(u) \geq 0$, we have $\frac{dR}{d\delta} > 0$ for all $\delta \geq 0$.

Therefore, $R(\mu_1, \mu_2)$ is strictly monotonically increasing with respect to $\delta = \mu_1 - \mu_2$. □

Using Theorem 2.1 we immediately get the following result:

Corollary B.1. *Under the conditions of Theorem 2.1, if $\sigma_1 = \sigma_2 = \sigma$, then:*

$$R(\delta) = \frac{\Phi\left(\frac{\delta}{2\sigma}\right)}{1 - \Phi\left(\frac{\delta}{2\sigma}\right)}, \tag{22}$$

and $\frac{dR}{d\delta} = \frac{\phi\left(\frac{\delta}{2\sigma}\right)}{2\sigma\left[1 - \Phi\left(\frac{\delta}{2\sigma}\right)\right]^2} > 0$.

Remark B.2. Theorem 2.1 shows that as the difference between the means of two normal distributions increases, the ratio of their right-tail integrals (with the midpoint of the means as the boundary) monotonically increases.

B.2 PROOF OF THEOREM 2.2

Proof. Let I_k index the samples in $\mathcal{X}^k$, and define

$$S_k = \sum_{i \in I_k} w_i x_i, \qquad W_k = \sum_{i \in I_k} w_i, \qquad W_k^{(2)} = \sum_{i \in I_k} w_i^2. \tag{23}$$

Independence and closure of Gaussian variables under weighted sums give

$$S_0 \sim \mathcal{N}(\mu_1 W_0, \sigma_1^2 W_0^{(2)}), \qquad S_k \sim \mathcal{N}(\mu_2 W_k, \sigma_2^2 W_k^{(2)}). \tag{24}$$

Writing $\mu_1 = \mu_2 + \delta$, for each incorrect answer k we have

$$\mathbb{P}(S_0 > S_k) = \Phi(z_k(\delta)), \qquad z_k(\delta) = \frac{\delta W_0 + \mu_2 (W_0 - W_k)}{\sqrt{\sigma_1^2 W_0^{(2)} + \sigma_2^2 W_k^{(2)}}}. \tag{25}$$

The union bound therefore yields

$$P_{\text{vote}}(\delta) = 1 - \mathbb{P}\left(\bigcup_{k=1}^{m} \{S_0 \le S_k\} \right) \tag{26}$$

$$\ge 1 - \sum_{k=1}^{m} \mathbb{P}(S_0 \le S_k) = \sum_{k=1}^{m} \Phi(z_k(\delta)) - (m-1) =: L(\delta). \tag{27}$$

For fixed variances, weights, and μ_2,

$$\frac{dL}{d\delta} = \sum_{k=1}^{m} \phi(z_k(\delta)) \frac{W_0}{\sqrt{\sigma_1^2 W_0^{(2)} + \sigma_2^2 W_k^{(2)}}} > 0, \tag{28}$$

because $\phi(z) > 0$, $W_0 > 0$, and each variance is positive. Thus $L(\delta)$ is a strictly increasing lower bound on the multi-answer weighted-voting accuracy. □

C Experiment Description

C.1 Baseline and Competitors

The core contributions of our proposed method lie in three key aspects: confidence computation, dynamic adjustment mechanisms within single question inference, and distribution-based voting methods. For confidence computation, we adopt the negative average log-probability from Self-Certainty (Kang et al., 2025) and the token group concept from DeepConf (Fu et al., 2025), selecting the tail group containing the answer to calculate the overall trajectory confidence. However, our approach differs in that we do not use fixed window groups, but instead partition semantic steps according to reasoning blocks. The experimental section analyzes the advantages of this approach (Figure 5 in §E.2). Regarding the single inference process adjustment mechanism, since dynamic adjustment of reasoning processes at test-time has not been extensively explored, we primarily compare against basic inference methods. For the voting method, we use weighted majority voting as our baseline and compare with other test-time scaling voting approaches, including Self-Consistency (SC) (Wang et al., 2022), BoN (Lightman et al., 2023), MoB (using confidence as the reward) (Rakhsha et al., 2025), Weighted-SC (WSC) (Li et al., 2022), DeepConf (WSC-Top50) (Fu et al., 2025).

C.2 Implementation Details

Regarding the implementation details of our methods, for trajectory step segmentation, we select "\n\n" as the reasoning block signal, which has been thoroughly studied in StepPruner (Wu et al., 2025). Regarding reflection information, we simply used "wait" as the predefined reflection token and supplemented more comparative experiments in Table 6 of §E.1. For the EMA update parameter α controlling τ_{conf} in *SelfStepConf* and the parameter δ controlling reflection trigger conditions, we set both to 0.8. For $N_{\mathcal{C}}$ in base voting, we simply set it to 10. These parameters are analyzed accordingly in the §F. For the device used in the experiment, unless otherwise specified, all experiments were conducted on NVIDIA H-Series GPU. In addition, for the parameter top-p in the inference process, we set it to 0.95.

Hierarchical Voting. Given trajectories $\mathcal{V}$ and their confidence weights $\mathcal{C}$, Hierarchical Voting first partitions the confidence range into $N_{\mathcal{C}}$ equal-width intervals:

$$\mathcal{C}^i = [c_{\min} + (i-1)h, c_{\min} + ih], \qquad h = \frac{c_{\max} - c_{\min}}{N_{\mathcal{C}}}, \tag{29}$$

where $c_{\min} = \min(\mathcal{C})$, $c_{\max} = \max(\mathcal{C})$, and $i = 1, \ldots, N_{\mathcal{C}}$. Weighted majority voting is defined as

$$f_{\text{WMaj}}(\mathcal{V}, \mathcal{C}) = \arg\max_{\text{ans}} \sum_{V_r \in \mathcal{V}} \mathbb{I}(A_r = \text{ans}) C_r, \tag{30}$$

where A_r is the answer extracted from V_r. For every nonempty interval, let $\mathcal{V}^i$ and $\mathcal{C}^i_{\mathcal{V}}$ denote its trajectories and corresponding confidences. Its winner and winner-specific mean confidence are

$$A^i_{\text{sub}} = f_{\text{WMaj}}(\mathcal{V}^i, \mathcal{C}^i_{\mathcal{V}}), \qquad \bar{C}^i_{\text{sub}} = \frac{\sum_{V_r \in \mathcal{V}^i} \mathbb{I}(A_r = A^i_{\text{sub}}) C_r}{\sum_{V_r \in \mathcal{V}^i} \mathbb{I}(A_r = A^i_{\text{sub}})}. \tag{31}$$

The interval winners are then aggregated by another weighted-majority vote:

$$f_{\text{HierV}}(\mathcal{V}, \mathcal{C}) = \arg\max_{\text{ans}} \sum_{i: \mathcal{V}^i \neq \emptyset} \mathbb{I}(A^i_{\text{sub}} = \text{ans}) \bar{C}^i_{\text{sub}}. \tag{32}$$

This two-level aggregation prevents a densely populated confidence interval from dominating solely through sample count. Algorithms 2 and 3 give the corresponding procedure.

Prompt Template. In our experiments, we used two different prompt templates to handle different benchmark. For GPQA-D, the format of the prompt is:

```
{
    "role": "user",
    "content": "Return your final response within \\boxed{} and only include
        the letter choice (A, B, C, or D) as your final response. {Question}"
}
```

For HMMT2025, AIME2024, AIME2025, and BRUMO2025, the specific format is:

```
{
    "role": "user",
    "content": "{Question}\nPlease reason step by step,
                and put your final answer within \\boxed{}."
}
```

Evaluation. For the main experimental setup, we employ the Qwen3 (Yang et al., 2025) series models ranging from 0.6B to 32B (operating in thinking mode by defaul, and models marked with * use non-thinking mode), along with DeepSeek-R1-0528-Qwen3-8B and DeepSeek-R1-Distill-Qwen-7B models (Guo et al., 2025) (referred to as DeepSeek-R1-8B and DeepSeek-R1-7B, respectively). In addition, we also used the Qwen2.5-Math-7B (Yang et al., 2024a) and Llama-3.1-8B-Instruct (Dubey et al., 2024) in analysis. We evaluate our approach on five mathematical reasoning benchmarks: HMMT2025 (Balunović et al., 2025), GPQA-D (Rein et al., 2023), AIME2024 (AIME, 2025), AIME2025 (AIME, 2025), and BRUMO2025 (Balunović et al., 2025). Following baseline configurations, we set the context length to 32k for Qwen3 series models and 64k for DeepSeek-Distill models. Unless otherwise specified, the temperature is configured as 0.7 for Qwen3 non-thinking mode and 0.6 for all other settings. All pass@1 metrics are computed by averaging results across 64 independent evaluations, while voting experiments use a Budget of 128 trajectories by default. In the experiments, all average results are computed using weighted averages based on the number of questions in each benchmark.

D PSEUDO CODE

Algorithm 1 SelfStepConf with Voting

Require: Question q, Budget B, Reflection tokens T_r, Threshold δ, EMA factor α
Ensure: Final answer A_{final}

Initialize $\mathcal{V} \leftarrow \emptyset, \mathcal{C} \leftarrow \emptyset$
for $b = 1$ to B **do**
  Initialize $\tau_{\text{conf}} \leftarrow 0, m \leftarrow 0$
  while not end of generation **do**
    Generate token t_i and compute $C_{\text{token}-i}$
    if step boundary detected **then**
      $m \leftarrow m + 1$, compute C_{G_m}
      Compute Δ_{conf} and $\mathbb{I}_R$ using Eq. 13 and Eq. 14
      if $\mathbb{I}_R = 1$ **then**
        Inject reflection tokens using Eq. 16
      else
        Update τ_{conf} using Eq. 15
      end if
    end if
  end while
  Compute trajectory confidence and add to $\mathcal{V}, \mathcal{C}$
end for
Fit GMM and map to correct/incorrect using Eq. 4
Filter $\mathcal{V}_{\text{pos}}$ and $\mathcal{V}_{\text{neg}}$ using Eq. 6
$A_{\text{neg}} \leftarrow f_{\text{vote}}(\mathcal{V}_{\text{neg}}, -\mathcal{C}_{\text{neg}})$
$A_{\text{pos}} \leftarrow f_{\text{vote}}(\mathcal{V}_{\text{pos}}, \mathcal{C}_{\text{pos}})$
if $A_{\text{pos}} \neq A_{\text{neg}}$ **then**
  Reject Filter trajectories and re-fit GMM to get $\hat{\mathcal{V}}_{\text{pos}}$ using Eq. 8
  $A_{\text{final}} \leftarrow f_{\text{vote}}(\hat{\mathcal{V}}_{\text{pos}}, \hat{\mathcal{C}}_{\text{pos}})$
else
  $A_{\text{final}} \leftarrow A_{\text{pos}}$
end if
return A_{final}

Algorithm 2 Base Voting with Hierarchical Strategy

Require: Trajectory set $\mathcal{V}$, Confidence set $\mathcal{C}$, Number of intervals $N_{\mathcal{C}}$
Ensure: Voting result A
1: Divide confidence range into $N_{\mathcal{C}}$ intervals using Eq. 29
2: Initialize $\mathcal{A}_{\text{sub}} \leftarrow \emptyset$
3: **for** $i = 1$ to $N_{\mathcal{C}}$ **do**
4: Get $\mathcal{V}^i$ for interval $\mathcal{C}^i$
5: **if** $\mathcal{V}^i \neq \emptyset$ **then**
6: $A^i_{\text{sub}} \leftarrow f_{\text{WMaj}}(\mathcal{V}^i, \mathcal{C}^i)$ using Eq. 31
7: Add $(A^i_{\text{sub}}, \bar{\mathcal{C}}^i_{A_{\text{traj}}=A^i_{\text{sub}}})$ to $\mathcal{A}_{\text{sub}}$
8: **end if**
9: **end for**
10: $A \leftarrow f_{\text{WMaj}}(\mathcal{A}_{\text{sub}})$ using Eq. 32
11: **return** A

Algorithm 3 Weighted Majority Voting

Require: Trajectory set $\mathcal{V}$, Weight set $\mathcal{W}$
Ensure: Selected answer A
1: Extract unique answers $\mathcal{A} \leftarrow \{A_{\text{traj}} \mid \text{traj} \in \mathcal{V}\}$
2: **for** each ans $\in \mathcal{A}$ **do**
3: Compute scores[ans] using Eq. 30
4: **end for**
5: $A \leftarrow \arg\max_{\text{ans} \in \mathcal{A}} \text{scores[ans]}$
6: **return** A

E Supplementary Experiment

E.1 Reflection Information Ablation Study

For *SelfStepConf*'s operation of injecting additional reflection information at steps where confidence significantly decreases during inference, we primarily use "wait" as the critical fork token to trigger model reflection for simplicity. Additionally, we experimented with other reflection information as shown in Table 6. It can be observed that under the condition of not introducing additional parameters (i.e., without using other models for judgment or learning how to reflect through training), the performance differences across different tokens are not significant when only injecting reflection prompt tokens.

Table 6: Ablation study results of different reflection tokens. Using DeepSeek-R1-8B to generate 64 trajectories for each question across 5 benchmarks, computing average pass@1.

Reflection Token	HMMT2025	GPQA-D	AIME2024	AIME2025	BRUMO2025	Avg.
"wait"	59.22	62.71	82.86	75.07	79.84	67.06
"Wait"	58.91	62.34	85.45	73.80	79.27	66.87
"Hmm"	59.53	62.46	84.85	73.00	78.23	66.78
"Alternatively"	58.75	62.98	83.02	72.97	78.18	66.85

E.2 Step Split Ablation Study

A key challenge in implementing *SelfStepConf* lies in how to reasonably partition steps. Previous works have proposed partitioning by fixed-size windows (Wang et al., 2025a; Fu et al., 2025; Wang et al., 2025d), by "\n" (Yu et al., 2025), and by complete Thought-Action-Observation cycles in multi-turn dialogues (Ji et al., 2025). Unlike these approaches, we need to dynamically adjust steps during test-time inference, thus we believe using "\n\n" to maintain paragraph-level logical integrity is more appropriate in reasoning tasks.

Additionally, we refer to (Wang et al., 2025c) and believe that high entropy tokens in trajectories may be potential fork tokens, thus conducting step checks at these critical positions provides decisive assistance for the overall reasoning direction. Specifically, we adopt the high entropy threshold

Table 7: Ablation study results of different step splitting methods. Using DeepSeek-R1-8B to generate 64 trajectories for each question across 5 benchmarks, computing the **average pass@1**. HET denotes High-Entropy Token.

Benchmark	"\n\n"	"\n"	HET	Fixed Window			
				256	512	1024	2048
HMMT2025	**59.22**	58.85	58.91	58.39	58.75	58.06	57.76
GPQA-D	**62.71**	61.43	57.37	56.69	56.32	60.11	57.07
AIME2024	82.86	82.76	81.61	82.24	**82.97**	82.76	81.41
AIME2025	**75.07**	73.23	70.68	73.70	73.54	73.02	70.83
BRUMO2025	**79.84**	78.75	78.59	79.01	77.76	78.28	77.97
Avg.	**67.06**	65.95	63.06	62.97	62.71	64.99	62.70

of 0.672 from (Wang et al., 2025c), performing confidence checks when token entropy exceeds this value. However, during experiments, we found that directly splitting steps only according to high-entropy tokens results in excessively short step lengths. Therefore, we further set a lower bound of 200 for step length. As shown in Table 7 and Figure 5, Similar to the evaluation methods of §F.2 and §F.3, we compared six indicators such as Avg Pass@1, Steps Count, Tokens Count, Confidence, Reflections Count and Time. Compared with fixed-size window-level (256/512/1024/2048), dynamic sentence-level ("\n"), and entropy-level approaches (HET, High-Entropy Token), paragraph-level partitioning ("\n\n") is more suitable for SSC. Furthermore, in terms of efficiency, it can be seen that the influence of different step splits on inference time is relatively obvious. Combining steps count, tokens count, confidence, and reflection count, it can be known that this influence is mainly related to the number of times reflection is triggered, and is also affected by the other several indicators.

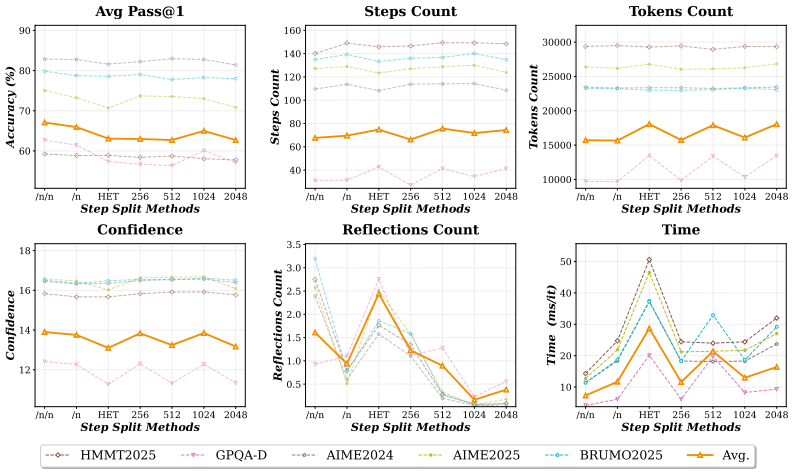


Figure 5: Ablation study results of different step splitting methods. Using DeepSeek-R1-8B to generate 64 trajectories for each question across 5 benchmarks. HET denotes High-Entropy Token. Complete results are provided in Table 25 of §G.16.

E.3 SSC's Impact on Model Sampling Behavior

This section examines SSC's effect on trajectory-level confidence changes to understand its influence on sampling distribution. First, we calculate the trajectory-level confidence of results generated by *SelfStepConf* (SSC) and BasicInference. As shown in Figure 6, SSC significantly improves confidence across all benchmarks, generating trajectories with higher correctness probability. And combined with the analysis in §5.2, it can be known that this improvement mainly comes from the correct samples.

The representative Pass@K curves are shown in Figure 7, with complete results in Figure 16. At $K = 1$, SSC improves over BasicInference, while the curves narrow as K increases. This pattern is consistent with improved sampling efficiency rather than an expanded reasoning limit (Yue et al., 2025).

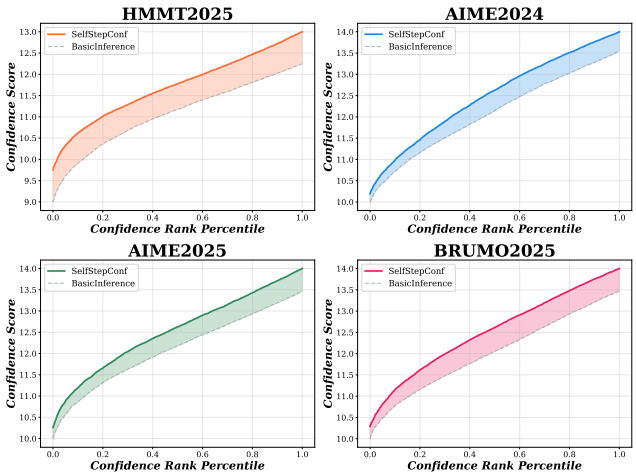


Figure 6: Comparing confidence between SSC and BasicInference using Qwen3-14B-NoThinking, sampling 512 traj/query. Complete results provided in Figure 15 of §G.8.

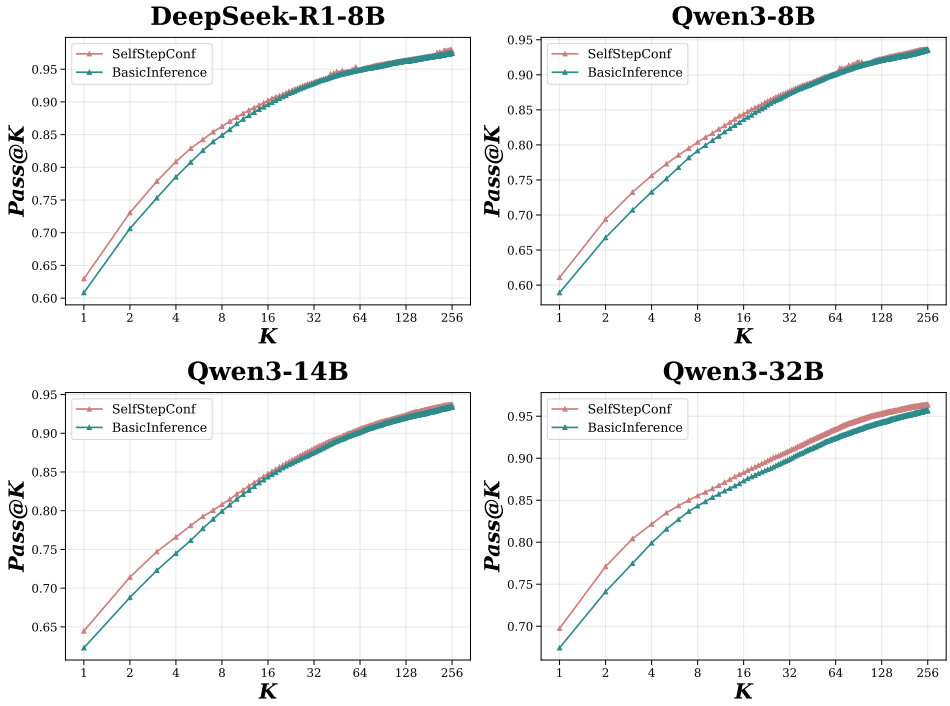


Figure 7: Pass@K on GPQA-D with 256 trajectories/query (64 repeats). SSC improves finite-budget accuracy, while the curves narrow as K increases.

Regarding SSC's improvement in sampling efficiency, Figure 8 shows that the enhancement primarily appears in moderately performing base models, exhibiting an arched growth pattern. For underperforming models with weaker reasoning capabilities, reflection injection is less effective. For high-performing models, the improvement is relatively weaker due to diminishing returns.

E.4 SelfStepConf Effects on Inference Dynamics

During inference, SSC monitors step confidence and intervenes when reflection is triggered. The main-paper example in Figure 4 uses temperature zero so SSC and BasicInference follow the same trajectory before the first intervention.

Two reflection triggers occur at steps 21 and 55. The trajectories subsequently differ in confidence and response length, and SSC maintains higher confidence after the second reflection before reaching

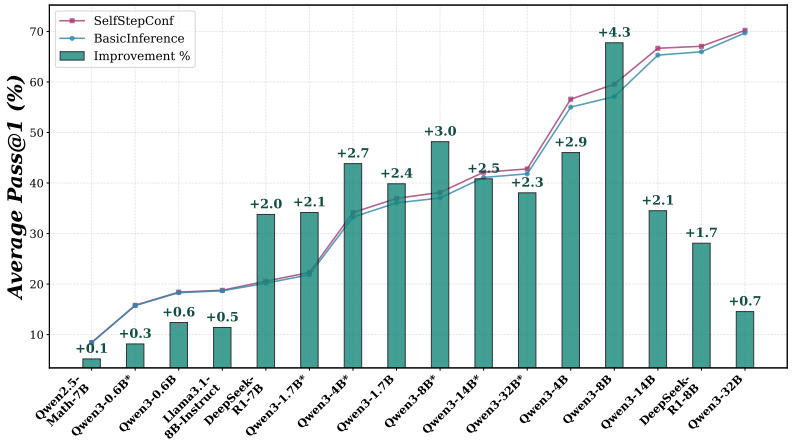


Figure 8: Pass@1 comparison between SSC and BasicInference using different models, sampling 1 response/query (64 repeats). Complete results are provided in Table 17 of §G.10.

the correct answer. BasicInference produces 91 steps and 9,472 tokens (104.1 tokens/step), whereas SSC produces 86 steps and 7,750 tokens (87.8 tokens/step). This is a single trajectory rather than evidence of uniform savings. Complete per-benchmark results are in Table 18. SSC also performs confidence checks during generation, with their frequency depending on step splitting (§E.2).

F Parameters Analysis

According to the implementation logic of *SelfStepConf* and *DistriVoting*, the main parameters are **(1)** top-k for computing token-level confidence, **(2)** EMA factor α for dynamically updating the confidence threshold in SSC, **(3)** δ for determining reflection trigger conditions in SSC, and **(4)** the number of intervals $N_{\mathcal{C}}$ specified in *HierVoting*.

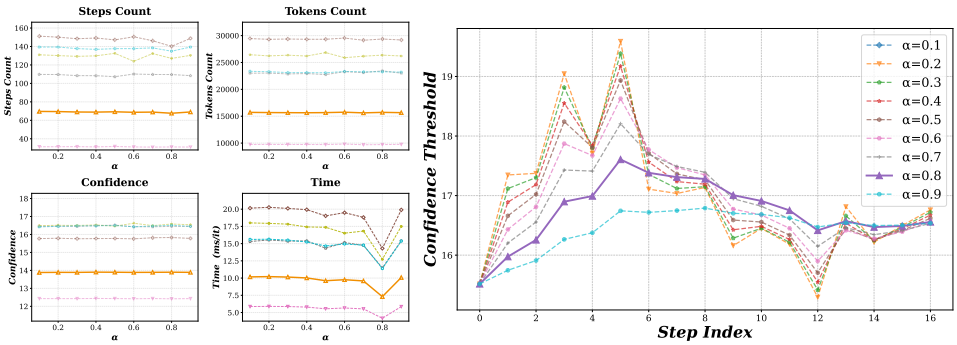
F.1 Analysis of top-k

First, for top-k, we follow the setting from prior work (Fu et al., 2025), setting $k = 20$. This choice is empirically justified by the probability distribution characteristics of modern LLMs. Through analysis of randomly sampled inference traces, we observe that at over 99.7% of token positions, the top-20 tokens account for more than 99% of the cumulative probability mass. This demonstrates that $K = 20$ effectively captures nearly all meaningful probability signal while maintaining computational efficiency. Additionally, this setting aligns well with practical deployment considerations, as standard inference engines like vLLM typically provide at most 20 log-probabilities by default.

F.2 Analysis of α

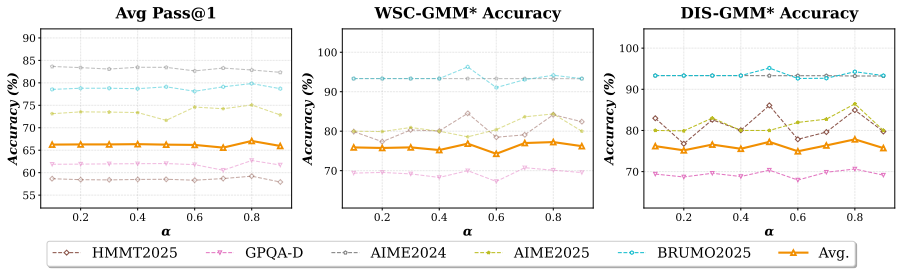
For α, δ, and $N_{\mathcal{C}}$, we primarily analyze their robustness through experiments. The EMA factor α mainly determines the proportion of the previous threshold when updating the threshold, corresponding to the case where $\mathbb{I}_R = 0$ in Equation 15. A larger α gives more weight to the confidence from previous steps, resulting in a smoother threshold curve, while a smaller α makes the threshold tend toward the step confidence G_m.

We analyzed the sensitivity of the parameter α by examining both Analysis Metrics (Steps Count, Tokens Count, Confidence, and Time) and Accuracy Metrics (Avg Pass@1, WSC-GMM* Accuracy, and DIS-GMM* Accuracy). Note that there are some discrepancies between these Analysis Metrics and the temperature-zero trajectory discussed in §5.6, as we used a temperature of 0.6 and repeated the process 64 times here (NVIDIA H-Series GPU), whereas that trajectory used a temperature of 0 (NVIDIA A-Series GPU). As shown in Figure 9 (a) and (c), within the range of α from 0.1 to 0.9, all metrics except time show minimal variation. This indicates that our proposed *SelfStepConf* and *DistriVoting* are not sensitive to the EMA update factor of the confidence threshold. Therefore, considering both efficiency and effectiveness, we opted for a setting of $\alpha = 0.8$.



(a) Analysis Metrics

(b) Variation of confidence threshold τ_{conf} with α



(c) Accuracy Metrics

Figure 9: Parameter sensitivity analysis of α under different metrics. Using DeepSeek-R1-8B to generate 128 trajectories per query, with experiments repeated 64 times. Complete results are provided in Table 19 of §G.12.

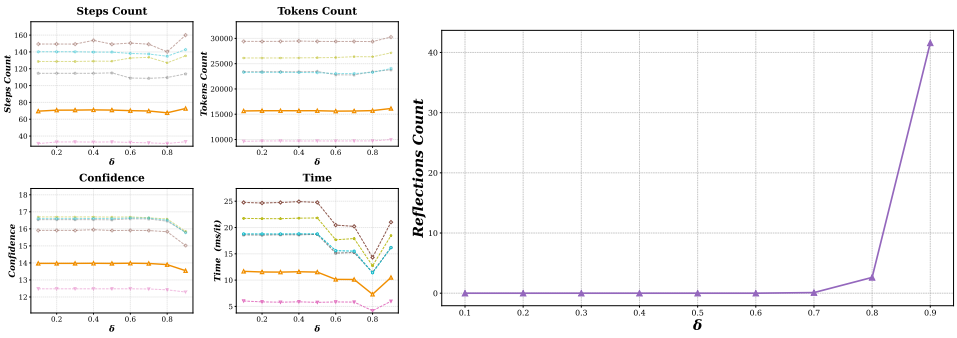
Additionally, to visually demonstrate the impact of different α values on *SelfStepConf*, we plotted the change in confidence threshold over steps for the same query under different α values in Figure 9 (b) (with temperature set to 0 and sample 1 trajectory for Question 7 of HMMT2025). It is evident that as α increases, the variation in the confidence threshold decreases, which aligns with the role of α as described in Eq 15.

F.3 Analysis of δ

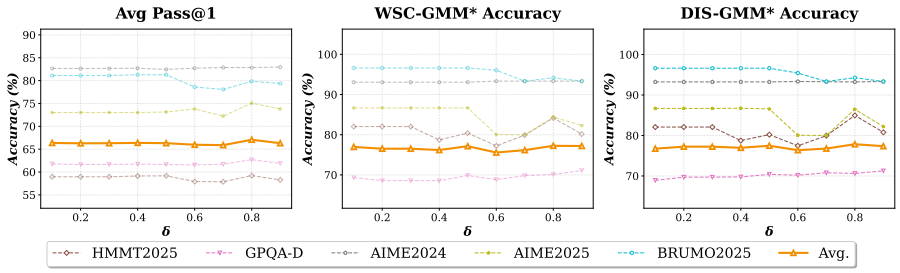
For δ in Equation 14, it determines how much the current step confidence needs to drop relative to the confidence threshold to trigger reflection. This value affects the strictness of the confidence checking mechanism: a larger δ means that even a slight drop in step confidence relative to the threshold will trigger reflection, while a smaller δ shows greater tolerance for drops in step confidence.

Similar to the analysis of parameter α, we examined the sensitivity of parameter δ from the perspectives of Analysis Metrics (Steps Count, Tokens Count, Confidence, and Time) and Accuracy Metrics (Avg Pass@1, WSC-GMM* Accuracy, and DIS-GMM* Accuracy), maintaining the same experimental setup. As shown in Figure 10 (a) and (c), parameter δ exhibits similar sensitivity to α, being insensitive to all metrics except time. Therefore, considering both efficiency and effectiveness, we chose $\delta = 0.8$.

Additionally, the role of δ primarily involves controlling the strictness of reflection triggers in Eq 14. To visually demonstrate its impact on *SelfStepConf*, we calculated the average number of reflection triggers for all questions in HMMT2025 under different δ settings with temperature set to 0. As shown in Figure 10 (b), the number of reflection triggers increases with δ, indicating that δ is directly proportional to the strictness of the reflection trigger conditions. Furthermore, when $\delta \leq 0.7$, the number of reflections drops to zero, causing *SelfStepConf* to degrade to BasicInference. Thus, from the perspective of reflection trigger conditions, $\delta = 0.8$ is also an appropriate choice.



(a) Analysis Metrics (b) Variation of reflection triggers counts with δ



(c) Accuracy Metrics

Figure 10: Parameter sensitivity analysis of δ under different metrics. Using DeepSeek-R1-8B to generate 128 trajectories per query, with experiments repeated 64 times. Complete results are provided in Table 20 of §G.13.

F.3.1 Analysis of $\alpha \times \delta$

To further analyze the sensitivity of parameters α and δ in SelfStepConf, we conducted a joint analysis using HMMT2025. Specifically, we set the range of α and β to $[0.1, 0.9]$, and then calculated the Analysis Metrics (Steps Count, Tokens Count, Confidence, Reflections Count, and Time) and Accuracy Metrics (Avg Pass@1, WSC-GMM* Accuracy, and DIS-GMM* Accuracy) under the group parameters. As shown in Figure 11, it can be observed that under joint analysis, $\alpha = 0.8$ and $\delta = 0.8$ remain a preferable choice. Furthermore, when δ is small (< 0.2), it can be observed that the various metrics show almost no change, which further proves that SelfStepConf degrades to Basic Inference in this case.

F.4 Analysis of $N_{\mathcal{C}}$

Finally, $N_{\mathcal{C}}$ in Equation 29 determines how many sub-intervals the confidence should be divided into during the BaseVoting process for Weighted Majority Voting. We evaluated three BaseVoting-based methods (DIS-Top50, DIS-GMM, and DIS-GMM*) across $N_{\mathcal{C}} \in [1, 20]$, as shown in Figure 12. The results show that except for DIS-Top50 at $N_{\mathcal{C}} = 1$ which exhibits notably different performance, all methods achieve similar results across different values of $N_{\mathcal{C}}$. This indicates that BaseVoting demonstrates considerable robustness for $N_{\mathcal{C}} > 1$. For simplicity, we adopt the default setting of $N_{\mathcal{C}} = 10$ throughout all experiments.

Based on our experimental results, we observe an interesting phenomenon: **stratified voting strategies ($N_{\mathcal{C}} > 1$) significantly improve performance only for the DIS-Top50 method, while showing limited effectiveness for *GMM Filter* approaches**. The fundamental reason why *GMM Filter* is insensitive to stratification operations lies in its powerful quality filtering capability, which has already addressed the problems that stratification strategies attempt to solve. Specifically, *GMM Filter* uses probabilistic modeling to precisely identify high-quality trajectories, resulting in relatively uniform quality across different confidence intervals after filtering, thereby reducing the necessity for stratification. Unlike DIS-Top50's simple threshold selection, *GMM Filter* has already removed most low-quality trajectories globally, making the voting results of remaining samples inherently stable.

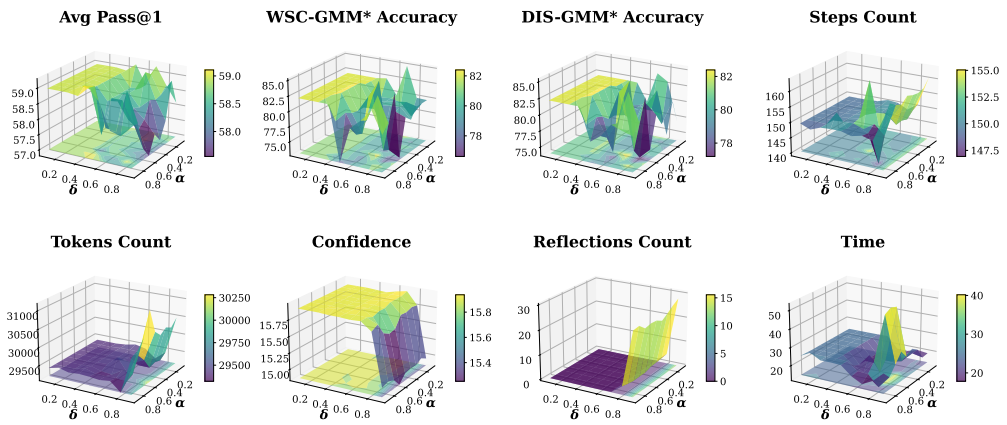
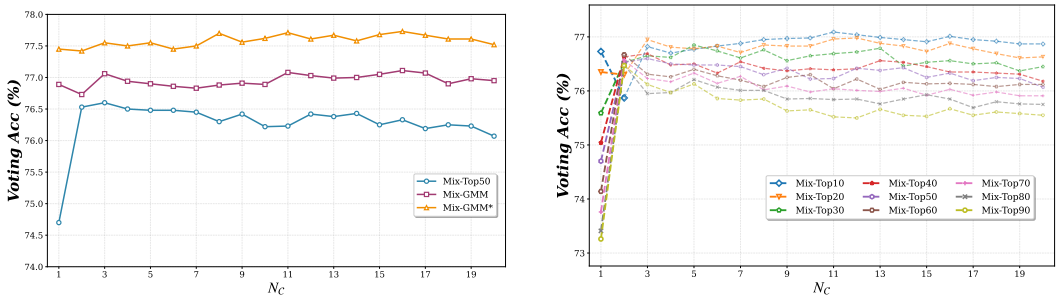


Figure 11: Parameter sensitivity analysis of $\alpha \times \delta$ under different metrics. Using DeepSeek-R1-8B to generate 128 trajectories per query of HMMT2025, with experiments repeated 64 times. Complete results are provided in Table 21 and Table 22 of §G.14.

When trajectory quality has already reached a high level through *GMM Filter*, further stratification operations provide limited performance gains, explaining why GMM and GMM* maintain relatively stable performance across different N_C settings. This observation is further supported by the trend shown in Figure 12 (b), where the performance difference between $N_c = 1$ and $N_C = 2$ decreases as the top-threshold increases. This finding demonstrates that **high-quality pre-filtering mechanisms can significantly reduce dependence on complex voting strategies**, which is consistent with the ablation analysis of *HierVoting* discussed in Table 10 of §G.1.



(a) N_C sensitivity under different voting methods

(b) N_C sensitivity under different Top-Threshold

Figure 12: Parameter sensitivity analysis of N_C under different voting methods. Using DeepSeek-R1-8B to generate 128 trajectories per query for voting, with experiments repeated 64 times. Complete results are provided in Table 16 of §G.6 and Table 23, Table 24 of §G.15.

G COMPLETE EXPERIMENTAL RESULTS

G.1 MAIN EXPERIMENTS AND ABLATION RESULTS ON MORE MODELS

More Main Results. In addition to the main experiments in the text involving the DeepSeek-R1 and Qwen3-32B models (Table 1 in §4.2), We also tested the voting performance of *SelfStepConf* and *DistriVoting* on Qwen3 models ranging from Qwen3-0.6B to Qwen3-32B, including both thinking mode and nonthinking mode. The results shown in Table 8 and Table 9 are consistent with the analysis in the main text, showing that *DistriVoting* consistently outperforms WSCVoting under different filter strategies. Moreover, both *DistriVoting* and WSCVoting demonstrate that the *GMM Filter* significantly outperforms the Top50 filter. The *SelfStepConf* generation method, indicated by *, further amplifies this improvement. Notably, the voting performance improvement of SSC on models with weaker and stronger reasoning abilities is not as pronounced, this observation aligns with our pass@1 analysis results of Figure 8 in §E.3 of the main text.

Table 8: All main results of *SelfStepConf* (SSC) and *DistriVoting* on various benchmarks using **Thinking Mode** of Qwen3-Series models. Set Budget to 128 and repeat 64 times. SC denotes Self-Consistency (i.e., Majority Voting), WSC represents Weighted Self-Consistency (i.e., Weighted Majority Voting), BoN represents Best of N, and *** indicates answers generated using the SSC approach**.

Model	Benchmark	Pass@1	SC	WSC	BoN	MoB	WSC			DIS		
							Top50	GMM	GMM*	Top50	GMM	GMM*
Qwen3-0.6B	HMMT2025	8.89	18.97	20.37	17.24	17.22	17.54	14.51	16.65	17.98	20.74	14.21
	GPQA-D	21.31	26.62	26.61	28.78	27.72	24.09	28.52	27.83	26.73	27.54	29.21
	AIME2024	10.73	17.08	17.55	13.44	23.95	29.58	23.84	29.74	26.20	28.45	27.22
	AIME2025	16.78	30.00	30.00	29.95	33.45	33.39	33.47	33.33	30.24	33.49	35.21
	BRUMO2025	16.93	23.13	23.33	23.39	30.25	29.79	28.83	28.96	30.45	29.66	27.11
	Avg.	18.30	24.99	25.18	25.85	27.16	25.40	27.25	27.58	26.54	27.75	27.97
Qwen3-1.7B	HMMT2025	22.29	29.64	29.48	36.61	26.82	28.12	32.30	32.93	27.14	32.81	35.77
	GPQA-D	34.91	43.22	43.94	42.91	43.51	43.60	44.74	48.36	45.51	45.85	48.26
	AIME2024	47.60	73.33	73.33	70.00	76.51	76.27	75.42	76.98	76.97	77.27	78.05
	AIME2025	34.48	46.67	47.29	39.95	50.00	50.08	55.25	53.07	52.19	54.85	51.94
	BRUMO2025	47.60	60.00	60.00	56.46	63.91	63.70	66.17	67.18	67.15	61.84	70.38
	Avg.	36.07	46.69	47.18	45.87	47.59	47.73	49.48	51.82	49.42	49.94	52.33
Qwen3-4B	HMMT2025	42.34	53.33	53.33	43.39	53.78	53.42	53.33	56.57	56.76	53.75	55.27
	GPQA-D	52.23	55.13	55.56	58.06	58.94	58.84	59.08	62.36	58.69	59.47	62.82
	AIME2024	71.30	80.00	80.00	76.72	83.35	83.47	83.33	82.41	84.01	83.41	83.91
	AIME2025	62.24	73.02	76.46	66.77	75.78	76.79	76.51	76.89	77.18	76.38	77.09
	BRUMO2025	62.60	67.55	70.00	73.39	76.83	75.07	76.04	77.10	77.13	76.22	76.67
	Avg.	55.02	60.17	60.99	60.70	64.03	63.88	64.07	66.47	64.38	64.36	66.75
Qwen3-8B	HMMT2025	41.46	59.95	60.00	54.27	61.72	60.16	60.99	63.75	61.20	62.34	62.60
	GPQA-D	54.70	63.83	63.80	63.22	64.11	64.35	64.48	65.37	64.29	64.43	66.31
	AIME2024	73.02	80.68	81.15	84.69	85.63	84.22	87.08	86.51	87.66	87.08	86.61
	AIME2025	62.55	77.19	77.55	64.79	74.32	74.95	74.22	72.45	73.65	73.59	74.11
	BRUMO2025	67.19	80.21	80.57	76.15	81.93	82.97	82.71	81.88	82.40	82.55	82.97
	Avg.	57.10	67.86	67.96	65.77	68.56	68.59	68.92	69.44	68.79	68.94	70.18
Qwen3-14B	HMMT2025	49.32	61.67	62.14	52.92	63.23	63.33	63.70	64.43	63.75	63.96	64.74
	GPQA-D	63.68	65.25	65.52	66.75	66.83	66.60	66.99	67.32	66.86	67.08	67.26
	AIME2024	78.59	85.00	85.42	85.99	86.20	84.74	87.76	85.68	88.39	88.28	86.82
	AIME2025	68.91	76.72	76.67	72.40	77.08	77.03	77.50	77.50	77.14	77.34	77.45
	BRUMO2025	75.16	80.52	80.83	83.70	86.41	85.26	86.25	85.31	86.41	86.61	85.89
	Avg.	65.31	69.30	69.58	69.39	71.13	70.74	71.44	71.45	71.41	71.59	71.60

Table 9: All main results of *SelfStepConf* (SSC) and *DistriVoting* on various benchmarks using **NonThinking Mode** of Qwen3-Series models. Set Budget to 128 and repeat 64 times. SC denotes Self-Consistency (i.e., Majority Voting), WSC represents Weighted SC (i.e., Weighted Majority Voting), BoN represents Best of N, and *** indicates answers generated using the SSC approach**.

Model	Benchmark	Pass@1	SC	WSC	BoN	MoB	WSC			DIS		
							Top50	GMM	GMM*	Top50	GMM	GMM*
Qwen3-0.6B	HMMT2025	0.97	3.34	3.83	3.50	6.91	6.95	4.94	6.94	5.27	6.90	5.70
	GPQA-D	23.14	23.39	23.17	23.79	22.12	20.45	23.92	24.09	24.24	24.32	24.94
	AIME2024	2.66	3.33	3.33	3.39	3.34	3.41	4.67	6.77	6.62	3.33	5.36
	AIME2025	2.45	6.67	6.67	3.44	7.83	6.61	7.51	9.96	10.03	9.69	8.40
	BRUMO2025	8.12	16.51	16.67	16.46	16.59	17.32	20.53	21.34	16.96	19.74	20.06
	Avg.	15.75	17.38	17.30	17.34	17.05	15.97	18.45	19.25	18.76	18.88	19.26
Qwen3-1.7B	HMMT2025	5.68	6.67	6.67	6.56	10.40	11.34	10.08	13.13	11.80	10.23	11.92
	GPQA-D	28.23	37.19	36.34	31.83	37.78	33.06	37.56	37.82	32.63	37.95	38.73
	AIME2024	12.92	23.49	23.65	10.10	25.73	26.84	29.87	32.97	26.85	30.26	32.97
	AIME2025	9.32	16.67	16.67	23.13	16.81	17.75	16.80	16.67	20.30	16.69	16.80
	BRUMO2025	17.40	23.33	23.33	19.95	25.26	30.53	25.62	26.41	31.09	26.61	25.36
	Avg.	21.85	29.78	29.26	25.45	30.90	28.74	31.15	31.96	28.81	31.53	32.33
Qwen3-4B	HMMT2025	11.67	16.67	16.67	19.84	16.67	16.80	16.91	17.23	16.69	16.80	16.47
	GPQA-D	41.26	44.79	46.69	46.48	47.17	45.35	47.85	48.36	46.97	48.13	48.46
	AIME2024	22.45	36.41	37.03	30.10	40.26	46.78	45.17	37.61	43.41	47.03	41.43
	AIME2025	18.39	20.36	23.18	23.28	33.28	33.39	33.14	31.20	33.23	32.44	35.40
	BRUMO2025	27.60	36.72	37.19	29.95	39.06	41.95	40.60	51.05	39.70	41.73	51.29
	Avg.	33.25	38.28	39.83	38.67	41.57	41.34	42.60	43.04	41.80	42.98	43.81
Qwen3-8B	HMMT2025	10.36	13.33	13.33	13.44	16.85	16.75	14.64	19.70	13.62	22.60	16.48
	GPQA-D	46.32	56.00	55.97	45.53	54.71	54.53	55.29	54.64	55.43	55.59	57.08
	AIME2024	28.23	46.51	46.30	23.44	43.68	40.23	40.16	44.70	43.51	48.51	44.25
	AIME2025	20.16	30.00	30.00	23.39	27.77	30.67	33.49	35.35	32.92	34.61	33.39
	BRUMO2025	28.02	41.35	43.13	33.39	45.15	46.37	50.00	43.54	49.93	39.49	44.99
	Avg.	37.03	47.24	47.38	37.18	46.65	46.60	47.47	47.54	47.72	48.31	48.66
Qwen3-14B	HMMT2025	9.81	21.82	20.26	20.53	20.69	25.61	22.16	25.54	25.84	26.51	21.94
	GPQA-D	52.38	55.47	58.51	54.06	58.07	56.04	59.45	56.28	55.98	56.45	60.71
	AIME2024	27.45	47.34	40.00	29.95	40.10	48.91	40.31	50.68	48.96	50.05	43.63
	AIME2025	21.09	35.51	33.07	30.10	33.33	38.58	34.17	39.60	37.82	38.31	35.49
	BRUMO2025	31.67	49.79	43.70	36.51	49.58	49.32	46.51	52.10	50.26	51.89	46.94
	Avg.	41.11	49.11	49.36	44.71	49.71	50.22	50.52	50.88	50.22	50.88	51.77
Qwen3-32B	HMMT2025	12.18	19.34	24.14	21.37	25.50	20.72	25.39	20.18	24.28	17.17	25.84
	GPQA-D	51.55	58.74	57.07	52.92	56.57	59.07	56.45	60.36	58.93	60.84	56.92
	AIME2024	30.05	40.00	46.67	34.22	50.16	40.07	49.64	44.63	40.91	49.63	50.89
	AIME2025	24.95	31.04	36.30	35.94	36.64	33.39	39.12	33.51	34.98	33.09	46.61
	BRUMO2025	35.73	43.49	50.00	34.74	49.64	49.51	50.63	43.68	46.22	45.80	52.24
	Avg.	41.81	49.21	50.36	44.86	50.50	50.34	50.69	50.98	50.50	51.62	52.00

Methods Ablation. We conducted detailed ablation experiments on the key components of the proposed methods: *SelfStepConf* and *DistriVoting* (*GMM Filter* + *Reject Filter* + *HierVoting*). As shown in Table 10 and Table 11, *SelfStepConf* is marked with ✗ for basic inference and ✓ for SSC inference. *GMM Filter* is marked with ✓ for using the *GMM Filter* and ✗ for using the naive top50 filter, and "–" indicates no filtering, using all trajectories for voting. *Reject Filter* is marked with ✗ and ✓ for not performing and performing the *Reject Filter*, respectively. *HierVoting* is marked with ✗ and ✓ for using Weighted-SC Voting and *HierVoting*, respectively. From the results, it is evident that SSC inference consistently outperforms basic inference under any setting, indicating that SSC provides a consistent improvement in voting performance. Additionally, among the three components of *DistriVoting*, the *GMM Filter* is crucial in affecting performance, as its presence significantly impacts voting results. The benefit of the *Reject Filter* on voting performance relies on the foundation of the *GMM Filter*, meaning that only after a good distribution split can the positive and negative results of the distribution be reliably rejected. In contrast, *HierVoting* improves voting performance even without the *GMM Filter*, but its impact is not significant with the addition of the *GMM Filter* and *Reject Filter*. This is primarily due to the diminishing marginal returns of performance enhancement, where the advantages of more complex voting strategies are mainly evident on a weaker filtering basis, aligning with our analysis of Figure 13 in §F.4.

Table 10: Complete ablation experiments on the method components in the main experiments using **DeepSeek-R1-8B**, sampling 128 trajectories/query (64 repeats). The naive version of the *GMM Filter* is the Top50 Filter, the naive version of *HierVoting* is Weighted-SC, and the naive version of the *Reject Filter* is no filtering. – indicates no filtering at all, i.e., directly using all trajectories for voting.

SelfStepConf	DistriVoting			Voting Acc (%)					
	GMM Filter	*Reject Filter*	*HierVoting*	HMMT2025	GPQA-D	AIME2024	AIME2025	BRUMO2025	Avg.
✗	✗	✗	✗	73.80±0.44	68.58±0.04	90.00±0.18	82.55±0.34	93.33±0.00	74.75±0.07
	✓	✗	✗	82.50±0.49	69.58±0.13	93.13±0.15	83.91±0.99	93.59±0.08	76.64±0.11
	✗	✓	✗	73.91±0.39	68.47±0.15	89.90±0.08	82.86±0.31	93.33±0.00	74.71±0.11
	✗	✗	✓	79.95±0.23	69.36±0.07	93.07±0.05	84.38±0.16	93.44±0.03	76.28±0.08
	✓	✓	✗	83.07±0.26	69.31±0.23	93.28±0.05	86.77±0.29	93.75±0.13	76.82±0.14
	✓	✗	✓	82.03±0.68	69.67±0.13	93.28±0.03	85.89±0.31	93.85±0.08	76.87±0.18
	✗	✓	✓	79.27±0.65	69.52±0.19	93.18±0.13	84.43±0.23	93.28±0.18	76.32±0.05
	✓	✓	✓	82.55±0.31	69.82±0.09	93.13±0.18	85.52±0.60	93.70±0.10	76.95±0.10
	–	–	✗	69.69±0.18	67.65±0.04	86.67±0.00	80.78±0.13	93.33±0.00	73.30±0.03
	–	–	✓	76.67±0.24	68.26±0.07	92.92±0.13	84.58±0.45	93.33±0.05	75.28±0.05
✓	✗	✗	✗	75.31±0.00	69.11±0.28	89.69±0.19	80.00±0.10	93.33±0.00	74.95±0.16
	✓	✗	✗	84.17±0.54	70.11±0.07	93.33±0.36	84.38±0.59	94.17±0.44	77.24±0.14
	✗	✓	✗	75.36±0.28	69.07±0.15	89.69±0.08	80.05±0.18	93.33±0.00	74.94±0.13
	✗	✗	✓	81.30±0.50	70.08±0.12	93.13±0.24	82.76±0.21	93.39±0.03	76.71±0.11
	✓	✓	✗	85.21±0.52	70.16±0.04	93.18±0.05	84.90±0.23	94.74±0.16	77.46±0.07
	✓	✗	✓	83.75±0.60	70.11±0.48	93.13±0.03	84.74±0.31	94.64±0.11	77.26±0.29
	✗	✓	✓	81.20±0.52	70.13±0.17	93.13±0.24	82.71±0.21	93.33±0.03	76.72±0.13
	✓	✓	✓	84.95±0.86	70.63±0.17	93.23±0.05	86.46±0.55	94.27±0.17	77.84±0.28
	–	–	✗	70.57±0.13	67.76±0.02	86.67±0.00	79.32±0.21	93.18±0.16	73.30±0.03
	–	–	✓	78.96±0.52	68.81±0.11	92.76±0.16	82.86±0.21	93.23±0.03	75.66±0.08

Table 11: Complete ablation experiments on the method components in the main experiments using **Qwen3-32B**, sampling 128 trajectories/query (64 repeats). The naive version of the *GMM Filter* is the Top50 Filter, the naive version of *HierVoting* is Weighted-SC, and the naive version of the *Reject Filter* is no filtering. – indicates no filtering at all, i.e., directly using all trajectories for voting.

SelfStepConf	DistriVoting			Voting Acc (%)					
	GMM Filter	*Reject Filter*	*HierVoting*	HMMT2025	GPQA-D	AIME2024	AIME2025	BRUMO2025	Avg.
✗	✗	✗	✗	63.96±0.29	72.03±0.17	88.28±0.31	76.93±0.26	92.71±0.00	75.22±0.12
	✓	✗	✗	64.84±0.23	72.42±0.11	88.65±0.31	79.22±0.34	92.71±0.03	75.79±0.10
	✗	✓	✗	63.85±0.23	72.01±0.15	88.28±0.36	76.98±0.29	93.33±0.00	75.26±0.10
	✗	✗	✓	64.79±0.36	72.29±0.25	88.70±0.49	79.11±0.31	93.28±0.05	75.76±0.18
	✓	✓	✗	65.00±0.21	72.51±0.16	88.54±0.44	78.96±0.55	93.33±0.00	75.88±0.09
	✓	✗	✓	64.48±0.26	72.75±0.27	89.11±0.36	78.75±0.18	93.23±0.03	76.01±0.15
	✗	✓	✓	64.95±0.24	72.33±0.33	88.70±0.52	79.06±0.34	93.28±0.03	75.79±0.20
	✓	✓	✓	64.43±0.21	72.74±0.25	89.01±0.39	78.70±0.23	93.23±0.08	75.99±0.09
	–	–	✗	62.24±0.39	70.55±0.14	86.88±0.67	77.08±0.10	93.33±0.00	74.07±0.16
	–	–	✓	62.66±0.42	70.34±0.15	89.32±0.16	80.42±0.18	93.18±0.08	74.51±0.12
✓	✗	✗	✗	65.63±0.49	72.53±0.12	90.42±0.36	77.86±0.33	93.33±0.00	76.03±0.12
	✓	✗	✗	65.21±0.57	72.43±0.25	90.16±0.21	80.00±0.42	93.28±0.18	76.10±0.18
	✗	✓	✗	65.63±0.55	72.53±0.09	90.42±0.34	77.76±0.21	93.33±0.00	76.02±0.09
	✗	✗	✓	64.74±0.42	72.99±0.16	89.27±0.47	79.38±0.55	93.33±0.15	76.27±0.16
	✓	✓	✗	65.16±0.55	72.48±0.28	90.05±0.26	79.90±0.60	92.71±0.15	76.05±0.21
	✓	✗	✓	63.23±0.47	73.13±0.06	89.22±0.57	80.31±0.60	92.66±0.06	76.23±0.09
	✗	✓	✓	64.64±0.29	73.13±0.15	89.38±0.52	79.17±0.44	93.28±0.05	76.33±0.17
	✓	✓	✓	65.73±0.05	73.18±0.02	89.11±0.50	80.05±0.44	93.33±0.08	76.53±0.08
	–	–	✗	64.11±0.18	71.43±0.09	87.45±0.52	77.55±0.52	93.33±0.00	74.90±0.03
	–	–	✓	65.00±0.31	71.97±0.19	88.39±0.39	79.38±0.47	93.18±0.08	75.56±0.09

G.2 Detailed Ablation Results of Distribution Splitting Methods

We compare Top50, K-Means, MeanShift, and GMM as distribution splitting methods on five benchmarks. As shown in Table 12, Acc and WAcc are defined in §5.1, and higher AUROC indicates better separation in the positive pool. Voting Acc is calculated with *DistriVoting*, Predict Acc is the binary assignment accuracy of the splitting method, and Predict Time is its average runtime per trajectory.

Table 12: Complete ablation study results of different correctness prediction methods. Using DeepSeek-R1-8B to generate 128 trajectories for each question across 5 benchmarks with *DistriVoting*, repeated 64 times.

Method	Metric	HMMT2025	GPQA-D	AIME2024	AIME2025	BRUMO2025	Avg.
Top50	Acc (%)	71.94	68.65	91.24	84.40	88.63	74.47
	WAcc (%)	72.39	68.71	91.41	84.57	89.02	74.61
	AUROC (↑)	0.5234	0.4921	0.5358	0.5129	0.6037	0.5117
	Voting Acc (%)	78.28	67.85	93.18	83.44	93.33	75.10
	Predict Acc (%)	56.77	52.66	53.62	53.98	56.68	53.64
	Predict Time (ms/it)	0.0300	0.1110	0.0961	0.0884	0.0443	0.0935
K-Means	Acc (%)	76.59	68.07	94.00	87.50	90.76	75.29
	WAcc (%)	76.64	68.07	94.04	87.49	90.88	75.32
	AUROC (↑)	0.5306	0.5047	0.5478	0.4827	0.6242	0.5204
	Voting Acc (%)	82.03	67.12	93.33	85.16	93.54	75.19
	Predict Acc (%)	71.12	51.00	62.79	62.51	62.66	56.19
	Predict Time (ms/it)	0.6490	0.6080	0.5560	0.5690	0.5880	0.6014
MeanShift	Acc (%)	76.93	69.82	94.40	88.03	91.33	76.55
	WAcc (%)	76.95	69.82	94.41	88.00	91.43	76.57
	AUROC (↑)	0.5125	0.5047	0.5383	0.4806	0.6051	0.5158
	Voting Acc (%)	82.71	67.40	93.28	85.47	93.96	75.50
	Predict Acc (%)	76.08	53.01	68.53	68.44	66.08	59.34
	Predict Time (ms/it)	1.8210	1.8950	1.8030	1.7700	1.7000	1.8492
GMM	Acc (%)	76.71	71.69	93.90	87.79	91.06	77.60
	WAcc (%)	76.92	71.72	94.00	87.86	91.24	77.68
	AUROC (↑)	0.6032	0.5605	0.6441	0.5569	0.6776	0.5831
	Voting Acc (%)	82.55	69.82	93.13	85.52	93.70	76.95
	Predict Acc (%)	72.33	56.38	66.20	65.11	65.19	60.46
	Predict Time (ms/it)	0.2730	0.354	0.3300	0.3350	0.2970	0.3369

G.3 Detailed Ablation Results of Voting Budget

For the representative budget-scaling results in Table 3 of §4.3, we provide all six budgets, methods, and benchmark-level results in Table 13.

Table 13: Complete ablation study results of different Budget. Using DeepSeek-R1-8B to generate Budget trajectories for each question across 5 benchmarks with *DistriVoting*, repeated 64 times. * indicates answers generated using the *SelfStepConf* approach.

Budget	Method	HMMT2025	GPQA-D	AIME2024	AIME2025	BRUMO2025	Avg.
8	WSC-Top50	71.09±1.30	67.37±0.87	89.06±0.34	80.52±0.16	90.31±0.34	73.17±0.51
	WSC-GMM	72.71±1.98	66.99±0.26	89.53±0.52	81.25±0.49	90.10±0.52	73.18±0.25
	WSC-GMM*	73.07±0.65	67.18±0.89	89.38±0.42	80.31±0.42	90.00±0.49	73.22±0.56
	DIS-Top50	70.78±1.30	67.14±0.69	88.75±0.39	80.47±0.00	90.16±0.34	72.95±0.31
	DIS-GMM	72.71±2.11	66.94±0.41	89.27±0.57	81.20±0.37	90.05±0.42	73.12±0.24
	DIS-GMM*	72.97±0.60	67.16±1.22	89.32±0.44	80.21±0.34	89.90±0.50	73.18±0.78
16	WSC-Top50	72.55±0.70	67.86±0.10	89.48±0.13	81.25±0.76	91.77±0.13	73.86±0.09
	WSC-GMM	75.78±1.20	67.89±0.20	90.42±0.03	83.23±0.86	92.55±0.44	74.53±0.17
	WSC-GMM*	76.72±0.86	68.05±0.01	91.25±0.52	82.14±0.24	92.34±0.15	74.68±0.07
	DIS-Top50	72.08±1.22	67.83±0.26	89.22±0.08	80.94±0.96	91.51±0.42	73.72±0.16
	DIS-GMM	74.58±0.68	68.35±0.67	91.09±0.28	82.97±0.83	92.40±0.55	74.73±0.39
	DIS-GMM*	75.94±0.52	68.19±0.04	90.78±0.08	82.45±0.47	93.07±1.32	74.74±0.13
32	WSC-Top50	74.01±0.83	68.49±0.28	89.69±0.55	81.51±0.34	93.07±0.23	74.56±0.24
	WSC-GMM	79.95±1.56	68.68±0.28	92.40±0.55	84.53±0.68	93.65±0.29	75.83±1.94
	WSC-GMM*	78.23±0.26	69.13±0.31	92.03±0.21	83.49±1.09	93.85±0.13	75.84±2.22
	DIS-Top50	73.39±0.89	68.47±0.23	90.89±0.42	82.03±1.04	92.92±0.34	74.64±0.16
	DIS-GMM	79.43±1.59	68.69±0.47	92.34±0.49	83.80±0.47	93.59±0.34	75.71±0.53
	DIS-GMM*	80.10±1.43	68.96±0.27	91.88±0.11	83.65±1.31	93.85±0.38	75.90±0.23
64	WSC-Top50	74.48±0.63	68.43±0.16	89.64±0.26	81.98±0.42	93.28±0.08	74.62±0.13
	WSC-GMM	81.20±0.52	69.02±0.15	92.97±0.21	85.31±0.68	93.75±0.26	76.30±0.15
	WSC-GMM*	82.03±0.65	69.82±0.12	92.76±0.39	83.96±0.26	94.32±0.36	76.78±0.05
	DIS-Top50	77.45±0.78	69.06±0.09	92.50±0.44	83.54±0.44	93.23±0.03	75.71±0.16
	DIS-GMM	80.94±1.46	69.14±0.44	92.97±0.16	85.21±0.94	93.75±0.36	76.34±0.39
	DIS-GMM*	81.30±0.49	70.13±0.50	92.76±0.21	83.70±0.13	94.17±0.16	76.87±0.18
128	WSC-Top50	73.80±0.44	68.58±0.04	90.00±0.18	82.55±0.34	93.33±0.00	74.75±0.07
	WSC-GMM	82.50±0.49	69.58±0.13	93.13±0.15	83.91±0.99	93.59±0.08	76.64±0.11
	WSC-GMM*	84.17±0.54	70.11±0.07	93.33±0.36	84.38±0.59	94.17±0.44	77.24±0.14
	DIS-Top50	79.27±0.65	69.52±0.19	93.18±0.13	84.43±0.23	93.28±0.18	76.32±0.05
	DIS-GMM	82.55±0.31	69.82±0.09	93.13±0.18	85.52±0.60	93.70±0.10	76.95±0.10
	DIS-GMM*	84.95±0.86	70.63±0.17	93.23±0.05	86.46±0.55	94.27±0.17	77.84±0.28
256	WSC-Top50	73.96±0.34	68.49±0.11	89.79±0.00	83.65±0.78	93.33±0.00	74.79±0.04
	WSC-GMM	83.65±0.34	69.59±0.25	93.33±0.00	86.82±0.57	93.54±0.21	77.04±0.06
	WSC-GMM*	86.25±0.23	70.24±0.02	93.23±0.16	84.32±0.36	94.74±0.78	77.56±0.00
	DIS-Top50	81.30±0.73	69.80±0.19	93.33±0.00	84.79±0.39	93.49±0.03	76.75±0.22
	DIS-GMM	83.91±0.26	70.21±0.20	93.33±0.00	86.77±0.16	94.43±0.55	77.53±0.11
	DIS-GMM*	85.42±0.70	71.15±0.08	93.33±0.00	84.90±0.40	95.47±0.50	78.18±0.05

G.4 Detailed Results on the Impact of GMM Filter and Reject Filter on Voting Information Quality

For the analysis of the impact of key components *GMM Filter* and *Reject Filter* of *DistriVoting* on the confidence quality of trajectories involved in voting in the main text Table 4 of §5.1, we provide detailed results for each benchmark here. In addition to DeepSeek-R1-8B, we also provide results for Qwen3-8B, Qwen3-14B, Qwen3-14B-NonThinking, and Qwen3-32B, as shown in Table 14.

Table 14: Complete results of the proportion of correct samples across different stages of the voting process. Different models generate Budget=128 responses for each query during voting, with experiments repeated 64 times and results averaged.

Model	Metric	Stage	HMMT2025	GPQA-D	AIME2024	AIME2025	BRUMO2025	Avg.
DeepSeek-R1-8B	Acc	I	60.36	64.54	86.83	79.92	81.22	69.27
		II	76.71	71.69	93.90	87.79	91.06	77.60
		III	77.43	75.86	94.08	88.75	91.34	80.41
	WAcc	I	61.59	64.73	87.40	80.70	82.31	69.74
		II	76.92	71.72	94.00	87.86	91.24	77.68
		III	77.63	75.89	94.17	88.83	91.52	80.48
Qwen3-8B	Acc	I	58.75	65.16	82.66	73.89	73.85	67.85
		II	73.25	71.78	88.69	80.94	81.19	75.26
		III	76.10	78.89	89.00	82.69	82.16	80.25
	WAcc	I	59.66	65.32	83.08	74.43	74.35	68.17
		II	73.34	71.80	88.74	80.93	81.26	75.30
		III	76.18	78.91	89.05	82.69	82.22	80.27
Qwen3-14B	Acc	I	60.98	68.56	85.70	76.73	79.70	71.28
		II	76.48	75.40	90.38	86.40	85.46	78.90
		III	78.32	82.20	90.76	87.40	86.73	83.56
	WAcc	I	61.70	68.70	86.02	77.32	80.16	71.57
		II	76.49	75.43	90.44	86.42	85.55	78.94
		III	78.33	82.22	90.82	87.42	86.82	83.59
Qwen3-14B-NonThinking	Acc	I	24.42	56.21	37.93	37.95	43.06	48.52
		II	28.67	62.13	42.14	42.42	45.52	53.66
		III	29.81	69.34	43.90	43.29	47.13	58.66
	WAcc	I	25.20	56.56	38.97	38.55	43.55	49.01
		II	28.93	62.22	42.53	42.60	45.69	53.81
		III	30.08	69.41	44.28	43.45	47.28	58.79
Qwen3-32B	Acc	I	59.29	72.72	87.65	73.33	81.82	73.78
		II	61.27	79.30	89.27	73.95	84.00	78.48
		III	63.44	85.51	89.45	76.60	85.07	82.91
	WAcc	I	59.94	72.91	87.98	73.57	82.48	74.07
		II	61.51	79.28	89.36	74.03	84.20	78.52
		III	63.67	85.48	89.53	76.68	85.25	82.96

G.5 Detailed Results on the Impact of *SelfStepConf* on Confidence Distribution

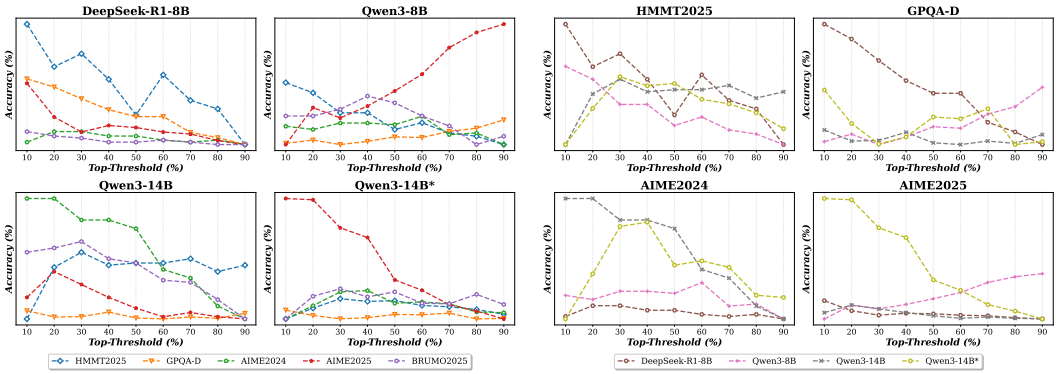
For the impact of *SelfStepConf* on confidence distribution under DeepSeek-R1-8B shown in Figure 2 of §5.2 in the main text, we provide additional quantitative experiments under more models (Qwen3-8B, Qwen3-14B, Qwen3-14B-NonThinking, and Qwen3-32B) here. As shown in Table 15, we primarily use two evaluation metrics: AUROC and WAcc. It can be observed that under different models, the confidence quality (AUROC) of SSC's generated results is higher compared to the BasicInference, and the average correct confidence (WAcc) is also higher.

Table 15: Complete results of the confidence separation of the content generated by BasicInference and *SelfStepConf* on different models.

Model	Benchmark	BasicInference		SelfStepConf		Δ (%)	
		AUROC	WAcc	AUROC	WAcc	AUROC	WAcc
DeepSeek-R1-8B	HMMT2025	0.9002	67.28	0.8957	67.72	-0.50	+0.65
	GPQA-D	0.7167	60.59	0.7699	61.29	+7.42	+1.16
	AIME2024	0.8833	86.56	0.8881	86.96	+0.54	+0.46
	AIME2025	0.8294	82.62	0.8478	82.87	+2.22	+0.30
	BRUMo2025	0.8198	84.88	0.8304	85.40	+1.29	+0.61
	Avg.	0.7701	68.04	0.8060	68.63	+4.66	+0.87
Qwen3-8B	HMMT2025	0.8424	48.39	0.9001	50.24	+6.85	+3.83
	GPQA-D	0.6518	56.06	0.7505	57.60	+15.14	+2.75
	AIME2024	0.8012	80.06	0.9099	80.25	+13.57	+0.24
	AIME2025	0.7390	66.43	0.8942	67.09	+21.00	+0.99
	BRUMo2025	0.7600	75.99	0.7992	75.82	+5.16	-0.22
	Avg.	0.7023	60.46	0.7978	61.66	+13.60	+1.98
Qwen3-14B	HMMT2025	0.8071	53.46	0.8979	53.90	+11.25	+0.82
	GPQA-D	0.6660	58.29	0.7447	60.12	+11.82	+3.14
	AIME2024	0.7745	83.93	0.9235	83.15	+19.24	-0.93
	AIME2025	0.6932	71.08	0.8849	70.48	+27.65	-0.84
	BRUMo2025	0.7566	79.38	0.7961	79.38	+5.22	+0.00
	Avg.	0.7007	63.45	0.7941	64.50	+13.33	+1.65
Qwen3-14B-NonThinking	HMMT2025	0.7499	19.40	0.8471	20.40	+12.96	+5.15
	GPQA-D	0.6229	25.96	0.6570	26.49	+5.47	+2.04
	AIME2024	0.7724	38.53	0.7945	43.42	+2.87	+12.69
	AIME2025	0.8535	32.53	0.9144	35.67	+7.14	+9.65
	BRUMo2025	0.6821	41.79	0.7131	45.99	+4.54	+10.05
	Avg.	0.6763	28.64	0.7175	30.22	+6.09	+5.52
Qwen3-32B	HMMT2025	0.7976	56.84	0.8868	57.76	+11.18	+1.62
	GPQA-D	0.6673	63.27	0.7763	66.04	+16.33	+4.38
	AIME2024	0.9538	83.93	0.8713	82.83	-8.65	-1.31
	AIME2025	0.6743	71.67	0.8371	71.88	+24.14	+0.29
	BRUMo2025	0.7273	83.29	0.7692	82.82	+5.76	-0.56
	Avg.	0.7129	67.29	0.8008	68.98	+12.33	+2.51

G.6 Top-Threshold Sensitivity Analysis on More Models and Benchmarks

For the results shown in Figure 3 of §5.3 in the main text, which display the optimal benchmark-level top-threshold applicable under different benchmarks using Qwen3-8B, we present here from both benchmark and model perspectives. As shown in Figure 13, we separately demonstrate that the applicable top-threshold varies for different models on the same benchmark, and also varies for different benchmarks on the same model. This further supports the rationale for using top50 to represent the fixed threshold filter method.



(a) Top-Threshold sensitivity across models

(b) Top-Threshold sensitivity across benchmark

Figure 13: Optimal fixed top-threshold traversal results for filter voting across different benchmarks. Using different models to generate 256 responses for each query, repeated 64 times. **Accuracy denotes WSC-TopK.** Complete results are provided in Table 16.

Table 16: Complete WSC-TopK results of optimal fixed top-threshold traversal results for filter voting across different benchmarks. Using different models to generate 256 responses for each query, repeated 64 times.

Model	Benchmark	Top-Threshold								
		10	20	30	40	50	60	70	80	90
DeepSeek-R1-8B	HMMT2025	81.88	80.83	81.15	80.52	79.64	80.63	80.00	79.79	78.91
	GPQA-D	70.24	70.04	69.75	69.48	69.31	69.31	68.92	68.79	68.62
	AIME2024	92.92	93.18	93.18	93.07	93.07	92.97	92.92	92.97	92.86
	AIME2025	85.47	84.64	84.27	84.43	84.38	84.27	84.22	84.06	83.96
	BRUMO2025	93.65	93.54	93.49	93.39	93.39	93.44	93.39	93.33	93.33
Qwen3-8B	HMMT2025	62.66	62.34	61.72	61.72	61.20	61.41	61.09	60.99	60.73
	GPQA-D	64.13	64.23	64.09	64.19	64.33	64.31	64.50	64.60	64.86
	AIME2024	86.98	86.88	87.08	87.08	87.03	87.29	86.72	86.77	86.41
	AIME2025	72.03	73.18	72.86	73.23	73.70	74.22	75.05	75.52	75.78
	BRUMO2025	82.45	82.45	82.66	83.07	82.86	82.45	82.14	81.56	81.82
Qwen3-14B	HMMT2025	62.71	63.96	64.32	64.01	64.06	64.06	64.17	63.85	64.01
	GPQA-D	67.10	66.95	66.96	67.07	66.93	66.90	66.95	66.92	67.04
	AIME2024	89.06	89.06	88.54	88.54	88.33	87.34	87.14	86.46	86.15
	AIME2025	77.55	78.18	77.86	77.55	77.29	77.08	77.19	77.08	77.03
	BRUMO2025	86.61	86.72	86.88	86.46	86.35	85.94	85.89	85.47	85.00
Qwen3-14B-NonThinking	HMMT2025	24.22	25.11	25.89	25.67	25.73	25.33	25.22	25.00	24.61
	GPQA-D	56.96	56.51	56.23	56.33	56.60	56.57	56.71	56.23	56.27
	AIME2024	48.39	49.48	50.63	50.73	49.69	49.79	49.64	48.96	48.91
	AIME2025	45.91	45.80	43.48	42.67	39.17	38.31	37.12	36.58	35.94
	BRUMO2025	48.91	50.78	51.41	50.73	51.15	50.21	50.16	50.94	50.10

G.7 Detailed Clustering Results of Answer Categories via GMM

§5.4 interprets GMM components as candidate quality pools rather than answer categories. Here, we visualize confidence distributions for selected frequent answers. As shown in Figure 14, the answer-conditioned distributions overlap substantially, so confidence alone does not cleanly identify answer categories.

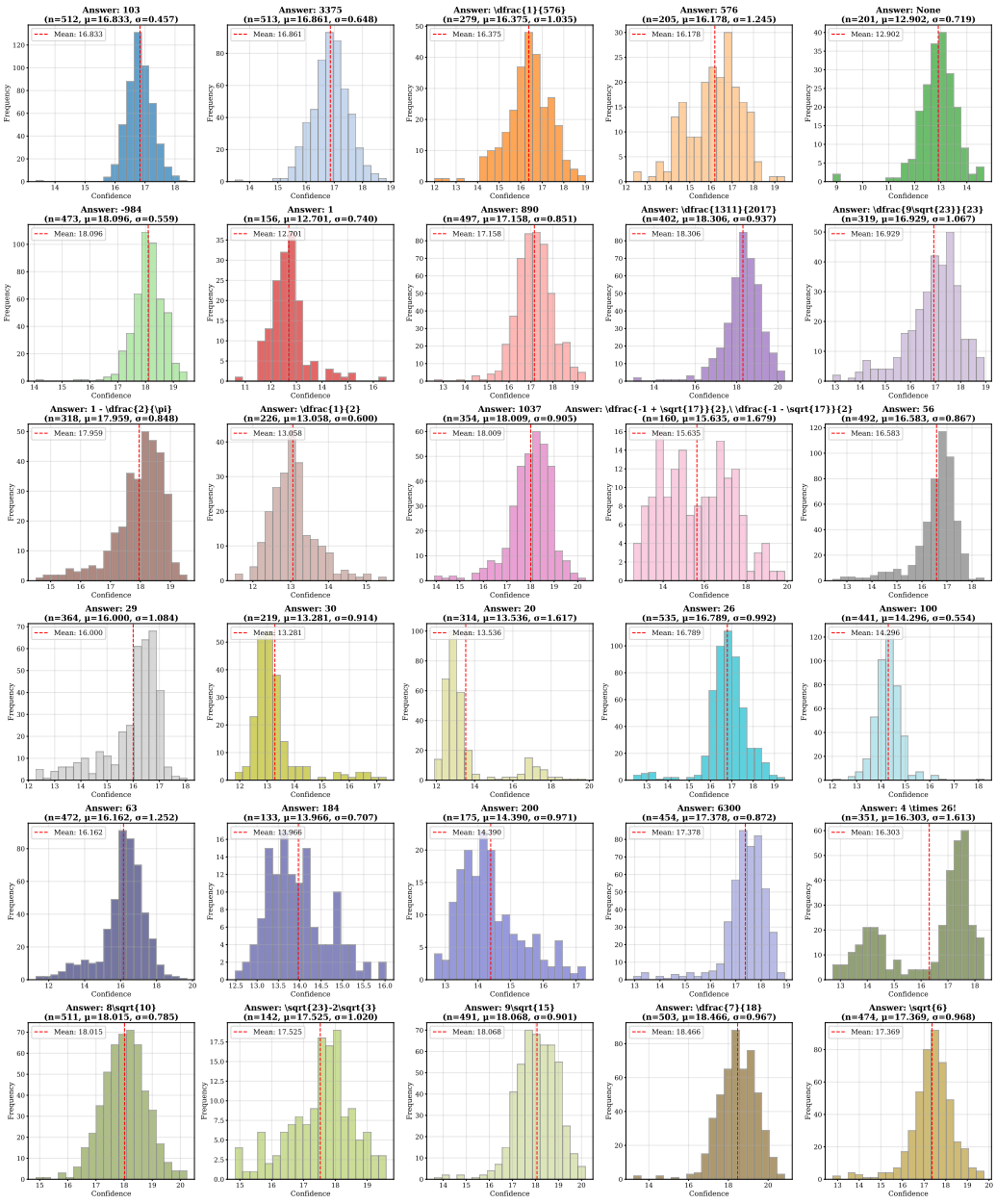
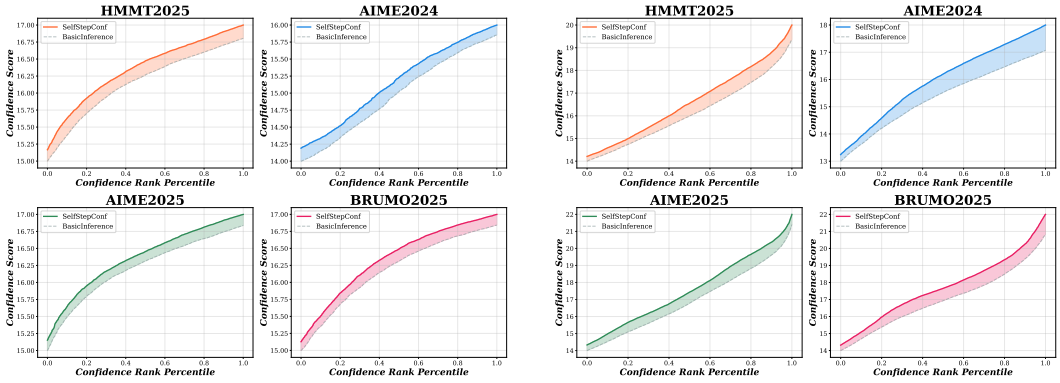


Figure 14: Distribution visualization of answers as Gaussian components in GMM using DeepSeek-R1-8B with 512 repeated sampling for each query on HMMT2025. Individual visualization of the top 30 most frequent answers.

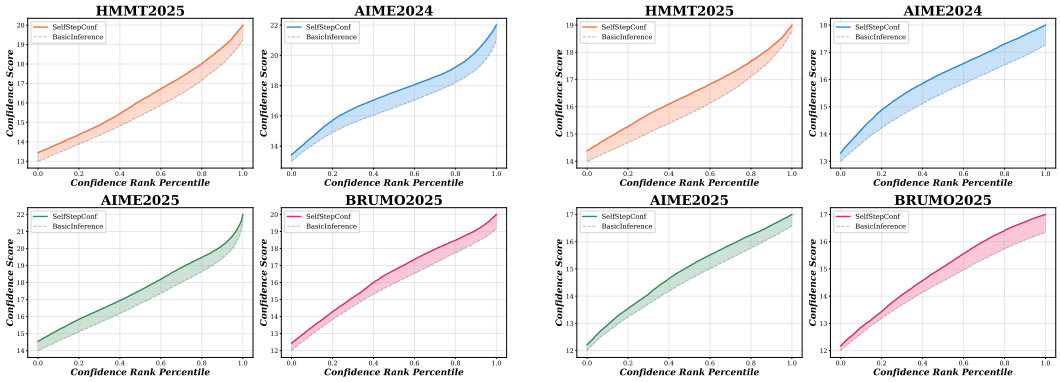
G.8 Additional Results on the Impact of SelfStepConf at the Trajectory-Level

For the visualization results in Figure 6 of §E.3 in the main text, which show the impact of *SelfStepConf* on trajectory-level confidence compared to the BasicInference using DeepSeek-R1-8B, we additionally provide results for Qwen3-8B, Qwen3-14B, Qwen3-14B-NonThinking, and Qwen3-32B here. As shown in Figure 15, it can be intuitively observed that our analysis conclusions remain consistent across different models.



(a) DeepSeek-R1-8B

(b) Qwen3-8B



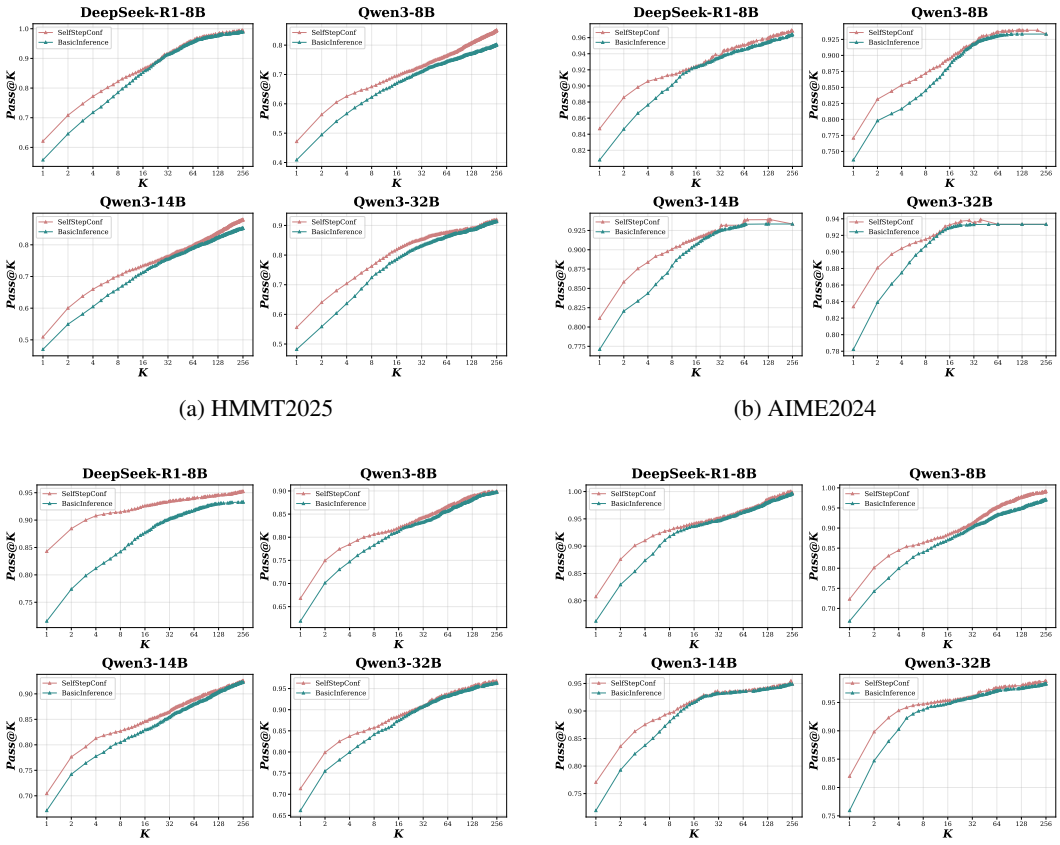
(c) Qwen3-14B

(d) Qwen3-32B

Figure 15: Complete results of trajectory-level confidence comparison between SSC and BasicInference across different benchmarks. Using different models to generate 512 responses for each query, and all responses across the entire benchmark are sorted and visualized by confidence scores.

G.9 PASS@K ANALYSIS OF SELFSTEPCONF ON MORE BENCHMARKS AND MODELS

For the analysis results shown in Figure 6 of §E.3 in the main text, which compare the pass@K performance of *SelfStepConf* to the BasicInference on GPQA-D under different models, we additionally provide results for four benchmarks: HMMT2025, AIME2024, AIME2025, and BRUMO2025. As shown in Figure 16, it can be intuitively observed that our analysis remains consistent across different benchmarks.



(a) HMMT2025

(b) AIME2024

(c) AIME2025

(d) BRUMO2025

Figure 16: Complete results of Pass@K comparison between SSC and BasicInference under different sampling numbers. Different models generate 256 responses on all Benchmark, repeated 64 times.

G.10 Detailed Pass@1 Results of SelfStepConf and BasicInference

For the average pass@1 results of *SelfStepConf* and BasicInference under different models shown in Figure 8 of §E.3 in the main text, we provide detailed data specific to each benchmark here, as shown in Table 17.

Table 17: Complete results of Pass@1 comparison between SSC and BasicInference across different models and benchmarks. Results averaged over 64 independent sampling/query.

Method	Model	HMMT2025	GPQA-D	AIME2024	AIME2025	BRUMO2025	Avg.
BaseModel	Qwen2.5-Math-7B	0.00	12.89	1.56	0.47	2.03	8.41
	Qwen3-0.6B-NonThinking	0.97	23.14	2.66	2.45	8.12	15.75
	Qwen3-0.6B	8.89	21.31	10.73	16.78	16.93	18.30
	Llama3.1-8B-Instruct	0.10	28.54	5.31	0.73	3.44	18.67
	DeepSeek-R1-7B	20.31	12.47	41.04	30.01	40.07	20.16
	Qwen3-1.7B-NonThinking	5.68	28.23	12.92	9.32	17.40	21.85
	Qwen3-4B-NonThinking	11.67	41.26	22.45	18.39	27.60	33.25
	Qwen3-1.7B	22.29	34.91	47.60	34.48	47.60	36.07
	Qwen3-8B-NonThinking	10.36	46.32	28.23	20.16	28.02	37.03
	Qwen3-14B-NonThinking	9.81	52.38	27.45	21.09	31.67	41.11
	Qwen3-32B-NonThinking	12.18	51.55	30.05	24.95	35.73	41.81
	Qwen3-4B	42.34	52.23	71.30	62.24	62.60	55.02
	Qwen3-8B	41.46	54.70	73.02	62.55	67.19	57.10
	Qwen3-14B	49.32	63.68	78.59	68.91	75.16	65.31
	DeepSeek-R1-8B	57.71	61.79	82.76	73.28	77.71	65.97
	Qwen3-32B	52.34	69.11	81.20	69.64	79.53	69.70
SelfStepConf	Qwen2.5-Math-7B	0.00	12.97	2.11	0.22	1.33	8.42
	Qwen3-0.6B-NonThinking	1.19	22.92	2.81	2.66	9.53	15.80
	Qwen3-0.6B	8.81	21.25	11.00	16.42	18.68	18.41
	Llama3.1-8B-Instruct	0.10	28.78	4.89	0.59	3.39	18.77
	DeepSeek-R1-7B	19.31	12.39	42.12	30.63	44.26	20.57
	Qwen3-1.7B-NonThinking	6.46	28.13	15.05	11.20	18.00	22.30
	Qwen3-4B-NonThinking	11.93	42.20	24.17	19.06	28.33	34.15
	Qwen3-1.7B	23.17	35.82	48.47	34.39	49.21	36.95
	Qwen3-8B-NonThinking	11.41	47.44	28.44	20.42	30.94	38.14
	Qwen3-14B-NonThinking	12.76	54.24	30.05	21.64	34.02	42.14
	Qwen3-32B-NonThinking	9.80	54.24	30.05	21.64	34.02	42.78
	Qwen3-4B	43.83	53.19	75.25	63.44	66.28	56.59
	Qwen3-8B	44.53	57.34	74.90	62.71	70.73	59.56
	Qwen3-14B	50.41	64.86	81.07	70.63	76.50	66.67
	DeepSeek-R1-8B	59.22	62.71	82.86	75.07	79.84	67.06
	Qwen3-32B	53.15	69.62	81.97	70.15	79.56	70.22

G.11 Detailed Results on the Dynamic Impact of SelfStepConf on the Inference Process

The main-text discussion in §5.6 reports representative trajectory behavior and average token and per-iteration timing changes for DeepSeek-R1-8B and Qwen3-8B. Table 18 provides the complete per-benchmark step, token, confidence, and timing results.

Table 18: Complete results of response length changes between SSC and BasicInference. Results are obtained at **temperature=0**, generating one response for each query in every benchmark, with values averaged across all queries within each benchmark (Time: ms/it).

Model	Benchmark	BasicInference				SelfStepConf			
		Step	Token	Confs.	Time	Step	Token	Confs.	Time
DeepSeek-R1-8B	HMMT2025	154.00	28266.73	17.03	396.07	154.40	28604.80	17.20	417.11
	GPQA-D	30.80	9560.65	13.31	124.27	29.27	9411.71	13.33	127.17
	AIME2024	88.63	21239.20	17.96	289.66	83.33	20733.97	17.92	295.24
	AIME2025	123.23	26673.87	17.87	374.35	128.10	26280.73	17.91	382.20
	BRUMO2025	135.40	23137.50	17.44	321.32	124.50	22291.60	17.45	318.73
	Avg.	66.47	15322.42	14.92	207.70	64.48	15097.02	14.95	212.51
Qwen3-8B	HMMT2025	50.03	22114.77	15.13	302.55	622.63	24983.03	15.08	357.86
	GPQA-D	18.68	9757.86	13.97	130.35	60.47	5243.37	14.89	76.72
	AIME2024	39.60	15649.53	16.59	230.83	78.73	19586.87	16.24	279.85
	AIME2025	58.60	19786.97	16.08	270.24	97.63	20736.83	15.93	294.75
	BRUMO2025	54.77	16939.90	16.50	233.05	183.00	20881.97	15.90	297.39
	Avg.	30.78	13103.12	14.76	178.96	130.29	11395.75	15.23	163.79

G.12 Detailed Results of Sensitivity Analysis for Parameter α

For the sensitivity analysis results of the parameter α shown in of §F.2, we provide the complete experimental data here. This mainly includes the voting results of Pass@1, WSC-GMM* and DIS-GMM* using DeepSeek-R1-8B with α ranging from 0.1 to 0.9, as shown in Table 19.

Table 19: Complete of sensitivity analysis of α using DeepSeek-R1-8B, sampling 128 trajectories/query (64 repeats), setting $\delta = 0.8$.

Metric	Benchmark	δ								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Accuracy Metric										
Pass@1	HMMT2025	58.65	58.44	58.39	58.49	58.53	58.32	58.69	59.22	57.92
	GPQA-D	61.87	61.90	61.96	62.00	62.03	61.79	60.54	62.71	61.70
	AIME2024	83.65	83.39	83.07	83.47	83.45	82.66	83.29	82.86	82.34
	AIME2025	73.13	73.54	73.49	73.39	71.65	74.61	74.23	75.07	72.86
	BRUMO2025	78.54	78.80	78.80	78.70	79.12	78.12	79.12	79.84	78.70
	Avg.	66.25	66.29	66.29	66.35	66.24	66.18	65.58	67.06	65.95
WSC-GMM*	HMMT2025	79.90	77.34	80.31	80.05	84.53	78.47	79.08	84.17	82.40
	GPQA-D	69.38	69.55	69.18	68.28	70.00	67.26	70.75	70.11	69.52
	AIME2024	93.33	93.33	93.33	93.33	93.32	93.32	93.32	93.33	93.28
	AIME2025	80.00	79.90	80.94	80.00	78.57	80.39	83.64	84.38	80.00
	BRUMO2025	93.33	93.33	93.33	93.33	96.29	91.05	93.07	94.17	93.33
	Avg.	75.89	75.75	75.90	75.22	76.84	74.27	77.00	77.24	76.21
DIS-GMM*	HMMT2025	82.97	76.77	82.60	80.05	86.09	77.85	79.61	84.95	79.74
	GPQA-D	69.40	68.70	69.60	68.81	70.35	67.95	69.85	70.63	69.14
	AIME2024	93.33	93.33	93.33	93.33	93.30	93.30	93.30	93.23	93.23
	AIME2025	80.00	79.90	83.02	80.00	79.99	81.95	82.75	86.46	80.00
	BRUMO2025	93.33	93.33	93.33	93.33	95.16	92.63	92.68	94.27	93.33
	Avg.	76.19	75.17	76.57	75.55	77.23	74.94	76.36	77.84	75.72
Analysis Metric										
Steps Count	HMMT2025	151.20	150.11	148.40	149.17	147.18	150.51	146.20	140.14	148.77
	GPQA-D	31.33	31.38	31.46	31.21	31.65	31.14	30.98	31.13	31.04
	AIME2024	109.74	109.63	108.43	108.38	107.26	110.16	109.68	109.60	108.42
	AIME2025	131.01	130.16	129.28	129.89	132.58	123.98	132.33	127.08	130.38
	BRUMO2025	139.42	139.39	137.71	136.89	137.66	137.69	138.50	134.88	139.41
	Avg.	69.64	69.47	69.01	68.90	69.24	68.70	68.92	67.66	69.04
Tokens Count	HMMT2025	29442.61	29314.99	29366.06	29318.86	29319.90	29541.56	29137.55	29368.05	29187.45
	GPQA-D	9745.97	9751.10	9741.49	9735.64	9722.62	9815.61	9689.64	9721.65	9759.88
	AIME2024	23005.22	23026.91	22870.45	22939.62	22745.89	23280.70	23119.20	23438.78	23010.60
	AIME2025	26444.03	26225.54	26356.69	26201.99	26843.64	25884.50	26175.51	26367.72	26211.33
	BRUMO2025	23355.66	23281.21	23119.51	23116.11	23081.49	23330.81	23262.30	23270.82	23205.88
	Avg.	15714.24	15679.80	15660.99	15644.51	15671.50	15745.95	15622.83	15717.76	15663.25
Confidence	HMMT2025	15.77	15.79	15.76	15.77	15.78	15.76	15.82	15.83	15.78
	GPQA-D	12.43	12.43	12.43	12.44	12.43	12.42	12.43	12.42	12.43
	AIME2024	16.45	16.46	16.45	16.49	16.52	16.43	16.43	16.45	16.45
	AIME2025	16.49	16.51	16.50	16.53	16.46	16.62	16.49	16.58	16.54
	BRUMO2025	16.42	16.45	16.48	16.48	16.50	16.43	16.47	16.51	16.45
	Avg.	13.88	13.89	13.89	13.90	13.90	13.89	13.89	13.90	13.89
Time (ms/it)	HMMT2025	20.15	20.27	20.11	19.93	19.03	19.50	18.81	14.31	19.90
	GPQA-D	5.87	5.89	5.87	5.79	5.54	5.66	5.53	4.17	5.86
	AIME2024	15.31	15.53	15.36	15.40	14.34	15.15	14.75	11.47	15.40
	AIME2025	18.01	17.94	17.83	17.42	17.37	16.52	16.82	12.73	17.50
	BRUMO2025	15.61	15.64	15.51	15.24	14.65	14.95	14.78	11.39	15.38
	Avg.	10.17	10.21	10.15	10.02	9.62	9.76	9.59	7.30	10.08

G.13 Detailed Results of Sensitivity Analysis for Parameter δ

For the sensitivity analysis results of the parameter δ shown in of §F.3, we provide the complete experimental data here. This mainly includes the voting results of Pass@1, WSC-GMM* and DIS-GMM* using DeepSeek-R1-8B with δ ranging from 0.1 to 0.9, as shown in Table 20.

Table 20: Complete of sensitivity analysis of δ using DeepSeek-R1-8B, sampling 128 trajectories/query(64 repeats), setting $\alpha = 0.8$.

Metric	Benchmark	δ								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Accuracy Metric										
Pass@1	HMMT2025	58.96	58.96	58.96	59.13	59.17	57.92	57.86	59.22	58.28
	GPQA-D	61.77	61.64	61.67	61.73	61.65	61.56	61.71	62.71	61.88
	AIME2024	82.66	82.66	82.66	82.71	82.45	82.71	82.86	82.86	82.97
	AIME2025	73.02	73.02	73.02	73.02	73.13	73.80	72.24	75.07	73.80
	BRUMO2025	81.11	81.11	81.11	81.28	81.28	78.59	78.07	79.84	79.38
	Avg.	66.36	66.28	66.30	66.38	66.31	65.97	65.88	67.06	66.31
WSC-GMM*	HMMT2025	82.03	82.03	82.03	78.70	80.42	77.24	79.90	84.17	80.16
	GPQA-D	69.32	68.59	68.59	68.57	69.86	68.84	69.83	70.11	71.09
	AIME2024	93.07	93.07	93.07	93.07	93.07	93.33	93.33	93.33	93.33
	AIME2025	86.67	86.67	86.67	86.67	86.67	80.05	80.00	84.38	82.29
	BRUMO2025	96.61	96.61	96.61	96.61	96.61	96.04	93.33	94.17	93.33
	Avg.	76.97	76.52	76.52	76.19	77.15	75.57	76.17	77.24	77.20
DIS-GMM*	HMMT2025	82.08	82.08	82.08	78.75	80.21	77.45	79.95	84.95	80.78
	GPQA-D	68.92	69.71	69.71	69.77	70.39	70.19	70.79	70.63	71.27
	AIME2024	93.23	93.23	93.23	93.23	93.23	93.33	93.28	93.23	93.33
	AIME2025	86.67	86.67	86.67	86.72	86.56	80.05	80.00	86.46	82.19
	BRUMO2025	96.61	96.61	96.61	96.61	96.61	95.42	93.33	94.27	93.33
	Avg.	76.74	77.24	77.24	76.96	77.47	76.37	76.77	77.84	77.36
Analysis Metric										
Steps Count	HMMT2025	149.39	149.39	149.33	153.73	149.22	150.52	149.19	140.14	159.92
	GPQA-D	31.04	32.94	33.05	32.83	33.05	32.42	31.93	31.13	33.24
	AIME2024	114.58	114.58	114.58	114.57	115.12	108.98	108.58	109.60	113.88
	AIME2025	128.66	128.66	128.66	129.03	128.94	132.75	133.61	127.08	135.34
	BRUMO2025	140.20	140.20	140.20	139.84	139.77	138.18	137.54	134.88	142.81
	Avg.	69.59	70.78	70.84	71.12	70.87	70.23	69.78	67.66	72.77
Tokens Count	HMMT2025	29410.30	29410.30	29407.64	29480.44	29403.09	29389.67	29401.74	29368.05	30271.32
	GPQA-D	9640.65	9710.41	9719.70	9698.47	9715.40	9704.60	9718.37	9721.65	9972.02
	AIME2024	23414.50	23414.50	23414.50	23406.22	23467.06	22798.41	22774.72	23438.78	23779.66
	AIME2025	26129.52	26129.52	26129.52	26151.06	26183.19	26215.12	26387.48	26367.72	27142.56
	BRUMO2025	23290.87	23290.87	23290.87	23260.09	23226.01	23043.78	23038.24	23270.82	24036.34
	Avg.	15648.44	15691.88	15697.41	15689.41	15698.21	15612.96	15636.17	15717.76	16136.34
Confidence	HMMT2025	15.91	15.91	15.91	15.95	15.90	15.91	15.90	15.83	15.02
	GPQA-D	12.48	12.48	12.48	12.48	12.48	12.48	12.47	12.42	12.29
	AIME2024	16.54	16.54	16.54	16.54	16.53	16.59	16.58	16.45	15.80
	AIME2025	16.70	16.70	16.70	16.70	16.69	16.70	16.65	16.58	15.85
	BRUMO2025	16.60	16.60	16.60	16.61	16.60	16.64	16.63	16.51	15.78
	Avg.	13.97	13.97	13.97	13.98	13.97	13.98	13.97	13.90	13.54
Time (ms/it)	HMMT2025	24.77	24.66	24.76	24.93	24.78	20.46	20.22	14.31	21.02
	GPQA-D	6.03	5.86	5.80	5.87	5.77	5.85	5.82	4.17	5.98
	AIME2024	18.61	18.58	18.63	18.62	18.69	15.14	15.24	11.47	16.18
	AIME2025	21.74	21.69	21.68	21.78	21.82	17.67	17.90	12.73	18.46
	BRUMO2025	18.79	18.78	18.78	18.80	18.79	15.60	15.51	11.39	16.18
	Avg.	11.67	11.55	11.52	11.59	11.52	10.14	10.12	7.30	10.50

G.14 Detailed Results of Sensitivity Analysis for Parameter $\alpha \times \delta$

For the sensitivity analysis results of the parameter $\alpha \times \delta$ shown in of §F.3, we provide the complete experimental data here. This mainly includes the voting results of Pass@1, WSC-GMM* and DIS-GMM* using DeepSeek-R1-8B on HMMT2025 with α and δ ranging from 0.1 to 0.9, as shown in Table 21 and Table 22.

Table 21: Complete results of parameter sensitivity analysis of $\alpha \times \delta$ on HMMT2025 using DeepSeek-R1-8B, sampling 128 trajectories/query (64 repeats). **Evaluate using Analysis Metric.**

Metric	α	δ								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Steps Count	0.1	149.39	149.39	149.39	149.34	150.66	150.36	148.73	151.20	160.66
	0.2	149.39	149.39	149.39	149.39	150.15	150.25	149.21	150.11	158.18
	0.3	149.39	149.39	149.39	149.39	150.21	150.61	149.34	148.40	157.68
	0.4	149.39	149.39	149.39	149.65	150.21	150.67	149.60	149.17	158.09
	0.5	149.39	149.39	149.39	149.65	150.21	151.46	151.98	148.97	158.97
	0.6	149.39	149.39	149.39	149.65	149.73	163.15	147.19	152.34	157.30
	0.7	149.39	149.39	149.39	149.65	149.22	148.59	150.93	147.97	158.29
	0.8	149.39	149.39	149.33	149.65	149.69	150.52	149.19	140.14	159.92
	0.9	149.39	148.07	149.39	153.73	149.69	148.81	149.64	148.77	156.52
Tokens Count	0.1	29410.30	29410.30	29410.30	29410.69	29496.48	29434.60	29210.27	29442.61	30491.65
	0.2	29410.30	29410.30	29410.30	29410.30	29441.36	29435.32	29317.60	29314.99	30306.47
	0.3	29410.30	29410.30	29410.30	29410.30	29447.65	29462.27	29310.82	29366.06	30401.81
	0.4	29410.30	29410.30	29410.30	29436.85	29447.65	29475.31	29319.77	29318.86	30287.35
	0.5	29410.30	29410.30	29410.30	29436.85	29447.65	29520.59	29497.41	29637.60	30471.49
	0.6	29410.30	29410.30	29410.30	29436.85	29355.62	29693.73	29325.25	29389.67	31158.71
	0.7	29410.30	29410.30	29410.30	29436.85	29403.09	29325.25	29500.52	29453.27	30498.60
	0.8	29410.30	29410.30	29407.64	29480.44	29423.84	29389.67	29401.74	29368.05	30271.32
	0.9	29410.30	29481.11	29410.30	29436.85	29423.84	29395.96	29435.42	29187.45	30187.49
Confidence	0.1	15.91	15.91	15.91	15.91	15.89	15.93	15.96	15.77	14.96
	0.2	15.91	15.91	15.91	15.91	15.92	15.93	15.93	15.79	15.02
	0.3	15.91	15.91	15.91	15.91	15.91	15.92	15.93	15.76	14.97
	0.4	15.91	15.91	15.91	15.90	15.91	15.93	15.93	15.77	14.98
	0.5	15.91	15.91	15.91	15.90	15.91	15.91	15.89	15.61	14.96
	0.6	15.91	15.91	15.91	15.90	15.90	15.91	15.89	15.59	14.85
	0.7	15.91	15.91	15.91	15.90	15.90	15.95	15.86	15.65	14.98
	0.8	15.91	15.91	15.91	15.95	15.90	15.91	15.90	15.83	15.02
	0.9	15.91	15.91	15.91	15.90	15.90	15.89	15.90	15.78	14.99
Reflections Count	0.1	0	0	0.000529	0.002092	0.010502	0.034449	0.321665	4.01803	28.00328
	0.2	0	0	0.000529	0.002092	0.007813	0.034449	0.302839	3.686321	26.48138
	0.3	0	0	0.000529	0.002092	0.007292	0.033929	0.271697	3.403557	24.61753
	0.4	0	0	0.000529	0.002092	0.007292	0.031845	0.25119	3.227719	22.9203
	0.5	0	0	0.000529	0.002092	0.007292	0.031845	0.225324	2.37307	24.35578
	0.6	0	0	0.000529	0.002092	0.007292	0.035053	0.184679	2.428287	24.59583
	0.7	0	0	0.000529	0.002612	0.008854	0.040833	0.163306	2.306881	24.03886
	0.8	0	0	0.000529	0.002092	0.010475	0.028646	0.195833	2.746875	26.40324
	0.9	0	0	0.000529	0.002092	0.010995	0.039674	0.185857	3.375634	26.08073
Time (ms/it)	0.1	24.67	24.73	24.67	24.70	24.79	18.61	18.45	20.15	19.45
	0.2	24.65	24.68	24.71	24.69	18.60	18.66	18.59	20.27	19.12
	0.3	24.65	24.70	24.66	24.68	18.60	18.56	18.41	20.11	25.58
	0.4	24.70	24.74	24.72	24.70	18.56	18.61	18.57	19.93	28.05
	0.5	24.68	24.68	24.68	24.65	18.61	18.55	21.54	51.62	25.64
	0.6	24.73	24.60	24.66	24.75	18.62	25.07	16.74	52.89	30.53
	0.7	24.63	24.67	24.61	24.76	20.12	17.78	24.72	51.04	25.59
	0.8	24.77	24.66	24.76	24.93	24.78	20.46	20.22	14.31	21.02
	0.9	24.94	27.79	24.85	24.93	24.69	19.90	16.94	19.90	25.21

Table 22: Complete results of parameter sensitivity analysis of $\alpha \times \delta$ on HMMT2025 using DeepSeek-R1-8B, sampling 128 trajectories/query (64 repeats). **Evaluate using Accuracy Metric.**

Metric	α	δ								
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Pass@1	0.1	58.96	58.96	58.96	58.96	59.17	58.54	59.22	58.65	58.13
	0.2	58.96	58.96	58.96	58.96	58.49	58.54	59.06	58.44	58.96
	0.3	58.96	58.96	58.96	58.96	58.49	58.59	59.22	58.39	57.76
	0.4	58.96	58.96	58.96	58.91	58.49	58.59	58.18	58.49	57.24
	0.5	58.96	58.96	58.96	58.91	58.49	58.70	58.28	57.97	57.55
	0.6	58.96	58.96	58.96	58.91	58.49	58.54	58.02	57.76	56.98
	0.7	58.96	58.96	58.96	58.91	58.49	58.54	58.75	58.13	57.34
	0.8	58.96	58.96	58.96	59.13	59.17	57.92	57.86	59.22	58.28
	0.9	58.96	58.96	58.96	58.91	58.49	58.54	58.02	57.92	58.28
WSC-GMM*	0.1	82.03	82.03	82.03	82.03	82.29	79.74	84.95	79.90	79.84
	0.2	82.03	82.03	82.03	82.03	79.58	79.74	79.27	77.34	79.84
	0.3	82.03	82.03	82.03	82.03	79.58	79.74	80.05	80.31	79.90
	0.4	82.03	82.03	82.03	82.76	79.58	79.74	79.90	80.05	78.28
	0.5	82.03	82.03	82.03	82.76	79.58	79.64	76.20	79.90	79.27
	0.6	82.03	82.03	82.03	82.76	79.58	82.19	85.05	74.17	73.28
	0.7	82.03	82.03	82.03	82.76	79.79	76.77	83.28	74.74	84.32
	0.8	82.03	82.03	82.03	78.70	80.42	77.24	79.90	84.17	80.16
	0.9	82.03	82.03	82.03	82.03	72.92	78.88	80.42	82.40	78.91
DIS-GMM*	0.1	82.08	82.08	82.08	82.08	81.98	79.22	84.01	82.97	76.88
	0.2	82.08	82.08	82.08	82.08	78.85	79.22	79.01	76.77	80.73
	0.3	82.08	82.08	82.08	82.08	78.85	79.22	80.05	82.60	75.94
	0.4	82.08	82.08	82.08	82.70	78.85	79.22	80.05	80.05	79.74
	0.5	82.08	82.08	82.08	82.70	78.85	79.22	78.49	81.67	80.21
	0.6	82.08	82.08	82.08	82.70	79.79	83.28	85.26	73.85	75.47
	0.7	82.08	82.08	82.08	82.70	80.21	74.64	83.23	75.52	83.96
	0.8	82.08	82.08	82.08	78.75	80.21	77.45	79.95	84.95	80.78
	0.9	82.08	82.08	82.08	82.08	77.08	78.29	80.31	79.74	79.27

G.15 Detailed Results of Sensitivity Analysis for Parameter $N_{\mathcal{C}}$

For the sensitivity analysis results of the parameter $N_{\mathcal{C}}$ shown in Figure 12 of §F.4, we provide the complete experimental data here. This mainly includes the voting results of *DistriVoting* under different filter methods on DeepSeek-R1-8B with $N_{\mathcal{C}}$ ranging from 1 to 20, as shown in Table 23 and Table 24.

Table 23: Complete results of parameter sensitivity analysis of $N_{\mathcal{C}}$ under different voting methods. Using DeepSeek-R1-8B to generate 128 trajectories per query for voting, with experiments repeated 64 times ($N_{\mathcal{C}} \in [1, 10]$).

Method	Benchmark	1	2	3	4	5	6	7	8	9	10
DIS-Top50	HMMT2025	74.17	78.39	80.63	80.31	80.36	79.95	81.09	79.53	80.89	79.64
	GPQA-D	68.55	70.08	69.74	69.65	69.55	69.59	69.44	69.48	69.41	69.31
	AIME2024	89.38	92.08	93.18	93.18	93.18	93.07	93.13	93.07	93.13	93.07
	AIME2025	82.45	84.27	84.27	84.06	84.43	84.79	84.38	84.22	84.38	84.38
	BRUMO2025	93.33	93.96	93.65	93.70	93.70	93.59	93.44	93.44	93.49	93.39
	Avg.	74.70	76.53	76.60	76.50	76.48	76.48	76.45	76.30	76.42	76.22
DIS-GMM	HMMT2025	83.02	80.42	81.61	81.88	82.08	82.08	82.81	82.55	82.66	82.55
	GPQA-D	69.44	69.84	70.06	69.79	69.67	69.66	69.48	69.57	69.58	69.63
	AIME2024	93.28	93.23	93.23	93.33	93.28	93.28	93.28	93.28	93.33	93.28
	AIME2025	86.72	84.74	85.63	85.78	86.09	85.78	85.68	85.83	86.04	85.73
	BRUMO2025	93.70	94.06	94.01	93.96	93.91	93.85	94.01	94.11	93.96	93.85
	Avg.	76.89	76.73	77.06	76.94	76.90	76.86	76.83	76.88	76.91	76.89
DIS-GMM*	HMMT2025	85.05	82.45	82.50	82.86	83.07	83.02	83.23	83.54	83.75	83.75
	GPQA-D	70.19	70.56	70.68	70.63	70.64	70.50	70.54	70.74	70.57	70.69
	AIME2024	92.86	92.97	93.02	93.07	93.13	93.23	93.18	93.18	93.18	93.18
	AIME2025	85.00	83.91	84.48	84.22	84.74	84.32	84.69	85.31	84.79	84.69
	BRUMO2025	94.74	95.68	95.57	95.21	94.84	95.05	94.79	94.74	94.69	94.64
	Avg.	77.45	77.42	77.55	77.50	77.55	77.45	77.50	77.70	77.56	77.62

Table 24: Complete results of parameter sensitivity analysis of $N_{\mathcal{C}}$ under different voting methods. Using DeepSeek-R1-8B to generate 128 trajectories per query for voting, with experiments repeated 64 times ($N_{\mathcal{C}} \in [11, 20]$).

Method	Benchmark	11	12	13	14	15	16	17	18	19	20
DIS-Top50	HMMT2025	80.05	80.42	80.31	80.63	80.36	80.31	79.64	80.31	80.05	79.64
	GPQA-D	69.28	69.55	69.47	69.54	69.33	69.53	69.36	69.35	69.44	69.28
	AIME2024	93.02	93.07	93.02	93.13	93.07	92.97	92.97	92.92	92.81	92.76
	AIME2025	84.32	84.22	84.43	84.11	83.91	83.59	83.96	84.01	83.54	83.39
	BRUMO2025	93.39	93.39	93.33	93.39	93.39	93.28	93.33	93.28	93.33	93.33
	Avg.	76.23	76.42	76.38	76.43	76.25	76.33	76.19	76.25	76.23	76.07
DIS-GMM	HMMT2025	83.59	83.02	83.02	83.28	83.39	83.18	83.49	83.13	83.23	82.86
	GPQA-D	69.66	69.70	69.67	69.65	69.76	69.87	69.74	69.71	69.67	69.67
	AIME2024	93.28	93.28	93.33	93.28	93.28	93.28	93.28	93.28	93.28	93.28
	AIME2025	86.51	86.30	85.94	86.15	85.83	85.94	85.99	85.78	85.89	85.94
	BRUMO2025	93.91	93.91	93.91	93.80	93.80	93.80	93.85	93.80	93.80	93.70
	Avg.	77.08	77.03	76.99	77.00	77.05	77.11	77.07	76.90	76.98	76.95
DIS-GMM*	HMMT2025	84.32	84.22	84.53	84.74	84.64	84.84	84.79	84.69	84.64	84.53
	GPQA-D	70.73	70.63	70.69	70.56	70.66	70.79	70.63	70.62	70.58	70.45
	AIME2024	93.18	93.13	93.18	93.18	93.13	93.13	93.18	93.18	93.18	93.18
	AIME2025	84.69	84.43	84.38	84.17	84.69	84.17	84.58	84.27	84.58	84.43
	BRUMO2025	94.69	94.74	94.69	94.58	94.58	94.58	94.58	94.48	94.48	94.58
	Avg.	77.71	77.61	77.67	77.58	77.68	77.73	77.67	77.61	77.61	77.52

G.16 Detailed Results of Step Split Ablation

For the metrics Avg Pass@1, Steps Count, Tokens Count, Confidence, Time, and Reflections Count under different step split methods shown in Figure 5 of §E.2, we have supplemented the complete experimental data here, as shown in Table 25.

Table 25: Complete ablation study results of different step splitting methods. Using DeepSeek-R1-8B to generate 64 trajectories for each question across 5 benchmarks, computing the **Analysis Metrics**. HET denotes High-Entropy Token.

Metric	Benchmark	"\n\n"	"\n"	HET	Fixed Window			
					256	512	1024	2048
Steps Count	HMMT2025	140.14	148.97	145.93	146.54	149.36	149.27	148.54
	GPQA-D	31.13	31.27	42.70	27.04	41.31	34.45	41.23
	AIME2024	109.60	113.68	108.24	113.79	114.13	114.28	108.59
	AIME2025	127.08	129.03	123.41	127.02	128.76	130.05	123.92
	BRUMO2025	134.88	139.24	133.23	136.11	136.73	140.19	134.80
	Avg.	67.66	69.56	74.78	66.22	75.63	71.81	74.34
Tokens Count	HMMT2025	29368.05	29487.89	29272.19	29448.24	28928.73	29365.49	29333.62
	GPQA-D	9721.65	9675.06	13493.27	9855.01	13386.14	10356.28	13442.01
	AIME2024	23438.78	23302.43	23356.76	23353.05	23244.43	23333.48	23458.67
	AIME2025	26367.72	26182.84	26782.66	26036.69	26090.98	26258.83	26818.51
	BRUMO2025	23270.82	23201.30	22994.43	22975.56	23063.84	23293.29	23064.78
	Avg.	15717.76	15663.19	18062.42	15741.19	17894.01	16094.58	18055.93
Confidence	HMMT2025	15.83	15.67	15.67	15.83	15.91	15.92	15.77
	GPQA-D	12.42	12.27	11.27	12.30	11.32	12.28	11.34
	AIME2024	16.45	16.33	16.35	16.49	16.56	16.56	16.39
	AIME2025	16.58	16.47	16.02	16.63	16.66	16.68	16.09
	BRUMO2025	16.51	16.35	16.48	16.55	16.52	16.61	16.51
	Avg.	13.90	13.75	13.10	13.84	13.24	13.85	13.17
Time (ms/it)	HMMT2025	14.31	24.74	50.62	24.43	23.98	24.41	31.99
	GPQA-D	4.17	6.12	20.10	6.15	19.74	8.25	9.36
	AIME2024	11.47	18.32	37.40	18.29	18.14	18.27	23.73
	AIME2025	12.73	21.87	46.51	21.21	21.40	21.73	27.06
	BRUMO2025	11.39	18.70	37.21	18.28	32.93	18.67	29.21
	Avg.	7.30	11.70	28.72	11.58	21.39	12.97	16.39
Reflections Count	HMMT2025	2.747	0.791	1.764	1.354	0.271	0.064	0.085
	GPQA-D	0.931	1.103	2.757	1.116	1.275	0.227	0.559
	AIME2024	2.393	0.597	1.576	1.105	0.196	0.040	0.078
	AIME2025	2.573	0.509	2.513	1.573	0.329	0.060	0.174
	BRUMO2025	3.197	0.772	1.862	1.584	0.284	0.062	0.017
	Avg.	1.609	0.939	2.445	1.225	0.896	0.163	0.381