Reliability-Aware Adaptive Self-Consistency for Efficient Sampling in LLM Reasoning

Anonymous ACL submission

Abstract

Self-Consistency improves reasoning reliability through multi-sample aggregation, but incurs substantial inference cost. Adaptive self-consistency methods mitigate this issue by adjusting the sampling budget; however, they rely on count-based stopping rules that treat all responses equally, often leading to unnecessary sampling. We propose **Re**liability-Aware **A**daptive **S**elf-**C**onsistency (ReASC), which addresses this limitation by reframing adaptive sampling from response counting to evidence sufficiency, leveraging response-level confidence for principled information aggregation. ReASC operates in two stages: a single-sample decision stage that resolves instances confidently answerable from a single response, and a reliability-aware accumulation stage that aggregates responses by jointly leveraging their frequency and confidence. Across five models and four datasets, ReASC consistently achieves the best accuracy-cost trade-off compared to existing baselines, yielding improved inference efficiency across model scales from 3B to 27B parameters. As a concrete example, ReASC reduces inference cost by up to 70% relative to self-consistency while preserving accuracy on GSM8K using Gemma-3-4B-it.

1 Introduction

Large language models (LLMs) have demonstrated strong performance on complex reasoning tasks, yet the inherent stochasticity of decoding introduces variability in intermediate reasoning trajectories, making it difficult to reliably obtain a correct reasoning trajectory from a single generation. Self-Consistency (SC) addresses this challenge by sampling multiple reasoning paths and aggregating their final answers, effectively accumulating evidence and yielding consistent performance gains (Wang et al., 2022). However, SC relies on a fixed sampling budget, applying the same number of samples to all inputs. As a result, some instances

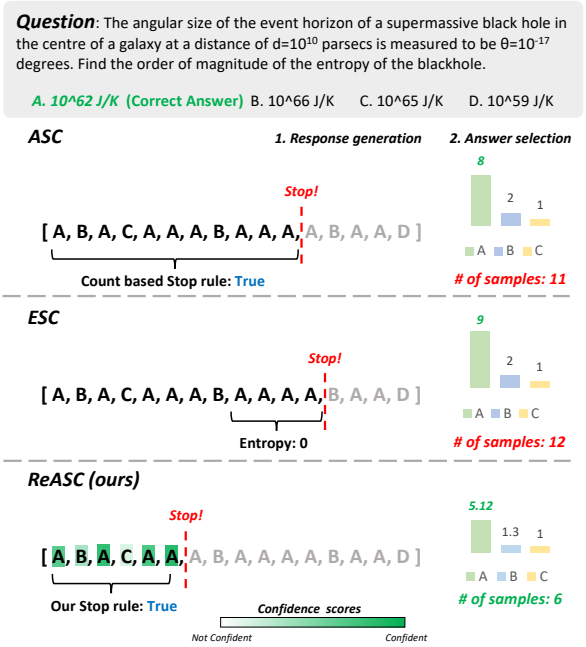

Figure 1: **Count-based stopping may lead to inefficient evidence accumulation.** Ignoring response reliability, count-based criteria may require unnecessary additional samples, while ReASC reaches the same decision with fewer samples.

continue sampling even after sufficient evidence has already been accumulated, while others remain unresolved even after the sampling budget is exhausted. To mitigate this inefficiency, adaptive self-consistency variants such as Adaptive Consistency (ASC) (Aggarwal et al., 2023) and Early-Stopping Self-Consistency (ESC) (Li et al., 2024) dynamically adjust the sampling budget based on observed responses. These methods primarily rely on count-based criteria to guide sampling decisions.

From an evidence accumulation perspective, this reliance on count-based aggregation treats all sampled responses as equally informative. By design, such aggregation does not account for differences in response reliability. However, reasoning trajectories generated under stochastic decoding can vary in reliability, with some responses providing strong

evidence while others being noisy or misleading. As a result, early reliable signals can be diluted by later unreliable responses, making it difficult to recognize when sufficient evidence has already been accumulated. This failure mode is illustrated in Figure 1, where ASC and ESC continue sampling even though the accumulated responses already provide sufficient evidence for a reliable decision.

These observations suggest that adaptive sampling decisions should be guided not only by how often an answer appears, but also by how much reliable evidence each individual response contributes to the final decision. Such response-level reliability can be captured from model confidence signals during generation, which provide instance-specific information about how strongly the model supports a given response (Wang and Zhou, 2024; Wang et al., 2024). Among various confidence signals, *self-certainty* has been shown to correlate with the reliability of reasoning trajectories, making it a suitable basis for guiding evidence accumulation without additional supervision (Kang et al., 2025).

Building on this insight, we propose **Re**liability-Aware **A**daptive **S**elf-**C**onsistency (ReASC), an adaptive self-consistency framework that incorporates a *self-certainty* variant as a response-level reliability signal to guide how evidence is accumulated at inference time. ReASC decomposes inference into two complementary stages, separating instances by evaluating the evidence sufficiency for each instance. In the first stage, the model evaluates the confidence of a single response to determine whether sufficient evidence is already available for a reliable decision. If additional evidence is required, the second stage performs reliability-aware evidence accumulation, allowing high-confidence responses to contribute more evidence than low-confidence ones. By guiding sampling decisions based on confidence-weighted evidence sufficiency rather than response counts alone, ReASC makes reliable decisions with fewer samples.

Empirically, ReASC consistently outperforms existing adaptive sampling baselines in the accuracy-cost trade-off across five models from three major families (LLaMA, Qwen, and Gemma) and four datasets. For instance, on GSM8K with Gemma-3-4B-it, ReASC reduces inference cost by approximately 70% relative to SC, while maintaining accuracy over prior adaptive baselines. Further analysis reveals that this efficiency gain arises from the complementary roles of ReASC's two stages. The first stage correctly resolves a substantial fraction of instances accurately with a single response. Notably, for instances that proceed beyond the first stage, the second stage still achieves substantial cost reductions without compromising accuracy. Together, these results show that ReASC establishes a principled framework for adaptive sampling by jointly modeling response counts and response reliability. Our contributions are as follows:

- We propose ReASC, a reliability-aware adaptive self-consistency framework that accumulates evidence by jointly considering response counts and response-level reliability.
- We empirically characterize the limitation of count-based stopping criteria, showing that treating all responses as equally informative can lead to unnecessary additional sampling.
- Through experiments and analyses, we show that ReASC consistently reduces inference cost while maintaining accuracy through the complementary roles of each stage.

2 Related Works

Adaptive Sampling for Self-Consistency. Self-Consistency (SC) improves reasoning reliability by sampling multiple reasoning trajectories and aggregating via majority voting, but relies on a fixed sampling budget, resulting in substantial inference cost (Wang et al., 2022). To address this limitation, adaptive self-consistency variants have adjusted the sampling budget based on observed responses (Aggarwal et al., 2023; Li et al., 2024; Wang et al., 2025). While these approaches reduce inference cost compared to SC, they rely on stopping criteria based on answer frequency or agreement patterns, implicitly treating all sampled responses as equally informative. As a result, they often lead to inefficient sampling even when sufficient evidence already exists to make a sampling decision.

Confidence Estimation. A growing body of work studies confidence and uncertainty estimation in large language models to assess the reliability of generated predictions. Prior work investigates response-level signals derived from model outputs, such as logit-based confidences and entropy-based uncertainty measures, showing that these signals correlate with prediction reliability (Kadavath et al., 2022; Kang et al., 2025; Zhang et al., 2023). Building on this line, several methods leverage confidence signals for post-hoc answer selection, reranking, and pruning of candidate responses (Wang

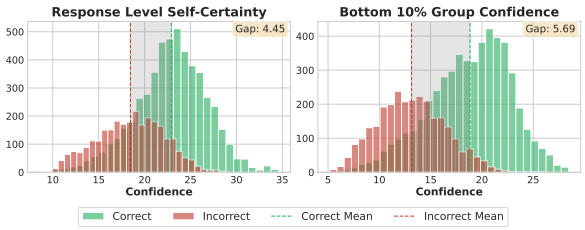

Figure 2: **Comparison of two confidence signals.** Using Gemma 3 4B-Instruct on MATH500, Bottom 10% Group Confidence shows a larger separation between correct and incorrect responses than Response-level Self-Certainty.

et al., 2024; Wang and Zhou, 2024; Taubenfeld et al., 2025; Fu et al., 2025). Our approach uses response-level confidence at inference time as a criterion for adaptive sampling. Specifically, ReASC interprets confidence as an estimate of response reliability and uses it to modulate how evidence is accumulated across sampled responses.

3 Preliminaries

Confidence as Evidence Strength. Recent work has shown that confidence signals derived from a model's token-level probability distribution correlate with the reliability of its reasoning trajectories (Kang et al., 2025; Fu et al., 2025). In this work, we interpret response-level confidence as an indication of how reliable evidence a generated response provides. Under this view, confidence naturally serves as a weighting signal in evidence accumulation, allowing more reliable responses to contribute more strongly. We adopt *self-certainty* as the underlying confidence signal (Kang et al., 2025). Given the model's token probability distribution at decoding step i, the token-level self-certainty is defined as

$$c_i = -\frac{1}{|\mathcal{V}|} \sum_{w \in \mathcal{V}} \log p(w \mid x, y_{\leq i}), \qquad (1)$$

where $\mathcal{V}$ denotes the vocabulary. Higher values correspond to a more concentrated probability distribution over tokens, reflecting greater model confidence during generation. Following Kang et al. (2025), response-level self-certainty is computed as the average of token-level self-certainties over the reasoning trajectory.

Bottom 10% Group Confidence. Response-level self-certainty summarizes the average confidence of a generated response, but can obscure localized uncertainty within a reasoning trace. To capture such localized uncertainty, we adopt the *Bottom 10% Group Confidence*. Specifically, given a response y, the token sequence is partitioned into sliding-window groups $\{G_1, G_2, \ldots, G_n\}$ and the group confidence C_{G_i} is computed by averaging the token-level self-certainties within the group G_i. The Bottom 10% Group Confidence is then defined as

$$C_{\text{bottom-10}}(y) = \frac{1}{|\mathcal{G}_{\text{b}}|} \sum_{G_j \in \mathcal{G}_{\text{b}}} C_{G_j}, \qquad (2)$$

where $\mathcal{G}_{\text{b}}$ denotes the set of groups with the lowest 10% group confidence. This metric emphasizes low-confidence segments indicative of unreliable reasoning, observed in prior work (Fu et al., 2025).

To determine which confidence metric reliably reflects response quality, we compare Bottom 10% Group Confidence with response-level self-certainty. As shown in Figure 2, Bottom 10% Group Confidence separates correct and incorrect responses more clearly than response-level self-certainty. Accordingly, we use this metric as the confidence signal in ReASC, with additional analyses provided in Appendix E.

4 Methods

ReASC is a reliability-aware adaptive self-consistency framework that decomposes inference into two stages to enhance efficiency. In **Stage 1 (Single-Sample Decision)**, the model evaluates the confidence of a single response to assess whether a reliable decision can be made without additional evidence, thereby avoiding unnecessary evidence accumulation. Inputs requiring additional evidence proceed to **Stage 2 (Reliability-Aware Accumulation)**, where the model accumulates confidence-weighted evidence from additional responses until a reliable decision can be made. An overview of ReASC is shown in Figure 3.

4.1 Stage 1: Single-Sample Decision

In Stage 1, ReASC assesses whether a reliable decision can be made from a single response. The model generates a response y and computes a response-level confidence score $S(y)$ using the Bottom 10% Group Confidence defined in Equation 2:

$$S(y) = C_{\text{bottom-10}}(y), \qquad (3)$$

which is interpreted as an estimate of response reliability and compared against a data-calibrated gating threshold τ_{gate}. If $S(y) \geq \tau_{\text{gate}}$, the response is accepted as providing sufficiently reliable evidence to determine the answer, and no further sampling is performed. Otherwise, the instance proceeds to

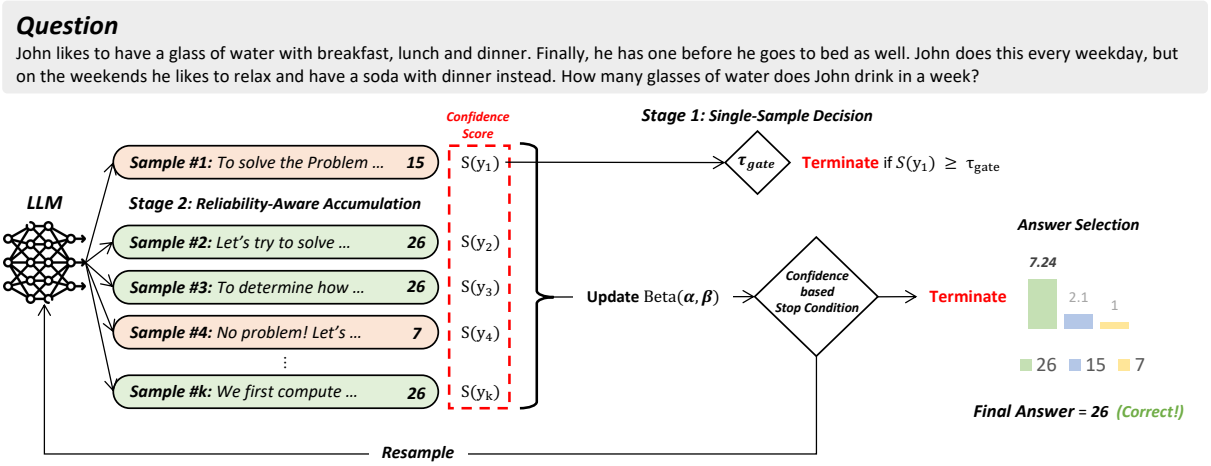

Figure 3: **Overview of ReASC.** The model first attempts a Single-Sample Decision (Stage 1) by evaluating whether the response reliability is sufficient. If not, it proceeds to Reliability-Aware Accumulation (Stage 2), where responses are adaptively sampled and aggregated via confidence-weighted Beta updates.

Stage 2 to accumulate additional evidence. The selection of τ_{gate} is detailed in Section 4.3.

4.2 Stage 2: Reliability-Aware Accumulation

When additional evidence is required, ReASC enters Stage 2 and accumulates confidence-weighted evidence from additional responses, interpreting confidence as an estimate of response reliability. Rather than relying on count-based agreement, this stage evaluates whether the accumulated evidence is sufficient to make a reliable decision.

ASC Beta Update. We first review the Beta-based stopping rule used in Adaptive Consistency (ASC) (Aggarwal et al., 2023). Let V denote the set of sampled responses, and let v_1 and v_2 be the counts of the most frequent and second most frequent candidates in V. The stopping decision is formulated as a binary comparison between these candidates, yielding a Beta posterior

$$p \sim \text{Beta}(\alpha, \beta), \qquad \alpha = v_1 + 1,\ \beta = v_2 + 1, \tag{4}$$

where p represents the probability that the most frequent candidate remains dominant as sampling continues. ASC stops sampling when the p exceeds a predefined threshold.

Confidence-Weighted Beta Update. The ASC formulation treats all sampled responses as equally informative, regardless of their reliability. Incorporating reliability into the aggregation allows more informative responses to exert greater influence, enabling sufficient evidence to be recognized earlier without being dominated by less informative ones. Building on this idea, we introduce a confidence-weighted variant of the Beta update, in which each sampled response contributes evidence that jointly accounts for its frequency and response reliability. Given a confidence score $S(y_i)$, we standardize it using statistics (μ, σ) estimated from a held-out calibration set, denote the standardized confidence as $z(y_i)$, and map it to an evidence weight via an exponential transformation:

$$v(y_i) \leftarrow v(y_i) + \max(1,\ \exp(\lambda z(y_i))) \tag{5}$$

where λ controls the sensitivity of the confidence-to-evidence mapping. This update can be interpreted as accumulating weighted pseudo-counts in the Beta posterior, allowing responses to contribute soft evidence proportional to their estimated reliability. The exponential form amplifies high-confidence responses, while the $\max(1, \cdot)$ operation ensures a minimum unit contribution, maintaining compatibility with the original count-based formulation. Among several reasonable designs, we find that this mapping yields stable stopping behavior across models and datasets (see Appendix D). This confidence-weighted formulation preserves the ASC Beta posterior structure, where (v_1, v_2) represent the accumulated confidence-weighted evidence of the two leading candidates.

Stopping Condition and Final Selection. Given the confidence-weighted Beta posterior, the stopping rule assesses the probability that the most frequent candidate remains dominant under further sampling. Accordingly, for a $\text{Beta}(\alpha, \beta)$ posterior, this probability admits the closed-form expression

$$P(p_1 > p_2 \mid V) = 1 - I_{1/2}(\alpha, \beta), \tag{6}$$

Algorithm 1 Offline Gating Threshold Calibration

Require: Calibration set $\{(x_i, y_i^*)\}_{i=1}^{k}$, target accuracy p_{target}
Ensure: Gating threshold τ_{gate}
// (1) Compute confidence for each instance
for $i = 1$ to k **do**
  Generate response y_i, then compute confidence $S(y_i)$
end for
// (2) Compute mean confidence of correct responses
$\mu_{\text{correct}} \leftarrow \mathbb{E}_{i:\, y_i = y_i^*}[\, S(y_i)\,]$
// (3) Compute accuracy-controlled threshold
Sort $\{S(\hat{y}_i)\}$ in ascending order as candidate thresholds
for each threshold t **do**
  $V(t) \leftarrow \{y_i : S(y_i) \geq t\}$
  Compute $\text{Acc}(t)$ over $V(t)$
  if $\text{Acc}(t) \geq p_{\text{target}}$ **then**
    $\tau_{\text{accuracy}} \leftarrow t$
    break
  end if
end for
$\tau_{\text{gate}} \leftarrow \max(\mu_{\text{correct}}, \tau_{\text{accuracy}})$

where $I_{1/2}(\alpha, \beta)$ denotes the regularized incomplete Beta function. A detailed derivation of this expression is provided in Appendix A. Evidence accumulation continues until $P(p_1 > p_2 \mid V) \geq C_{\text{threshold}}$, where $C_{\text{threshold}}$ is a predefined confidence threshold, or until a maximum sampling budget is reached. Once the stopping condition is met, the final answer is selected as $\hat{y} = \arg\max_y v(y)$, corresponding to the leading candidate supported by the accumulated confidence-weighted evidence.

4.3 Selecting Thresholds

ReASC leverages calibrated confidence statistics to support reliability-aware decision-making in both stages. Specifically, the framework requires (1) confidence statistics (μ, σ) to standardize response-level confidence scores for confidence-weighted Beta update in Stage 2, and (2) a decision threshold τ_{gate} that determines whether a reliable decision can be made from a single response in Stage 1. Depending on the availability of labeled validation data, we estimate these quantities using one of two calibration settings: *offline* or *online* calibration.

4.3.1 Offline Calibration

In the offline setting, we assume access to labeled validation data and use a subset of k labeled instances for calibration. We compute the mean and standard deviation (μ, σ) of response-level confidence score from single-sample responses, which are used to standardize confidence in the confidence-weighted Beta update.

Selecting the Offline Gating Threshold. The gating threshold τ_{gate} is designed to accept a single-sample response only when its reliability is sufficiently high. Specifically, we consider two complementary criteria. First, we compute the mean confidence of correctly solved instances, μ_{correct}, which reflects the typical confidence level of reliable single-sample decisions. Second, to complement μ_{correct}, we derive an accuracy-controlled threshold τ_{accuracy} that conservatively selects a high-confidence acceptance region by identifying the smallest confidence value t such that the accuracy of instances accepted with $S(y) \geq t$ meets a target accuracy score p_{target}. The gating threshold is then defined as

$$\tau_{\text{gate}} = \max\{\mu_{\text{correct}},\ \tau_{\text{accuracy}}\}. \tag{7}$$

The full offline calibration procedure is summarized in Algorithm 1.

4.3.2 Online Calibration

In the online setting, labeled validation data are unavailable. Accordingly, confidence statistics are estimated from the confidence scores obtained during the single-sample generation in Stage 1 across all test instances. Specifically, we compute the mean and standard deviation (μ, σ) of these confidence scores using the test instances as the calibration set, without requiring any additional inference.

Selecting the Online Gating Threshold. As in the offline setting, the gating threshold τ_{gate} is designed to accept a single response only when its reliability is sufficiently high. However, in the online setting, labeled validation data are unavailable, and the confidence distribution of correct responses cannot be directly observed. To approximate the role of correctness information used in the offline setting, we model the distribution of unlabeled confidence scores using a Gaussian Mixture Model (GMM), treating correct and incorrect responses as latent Gaussian components. This modeling choice is supported by model-selection criteria such as AIC and BIC (Akaike, 2003; Schwarz, 1978), which consistently favor a two-component mixture, indicating that the confidence distribution is well captured by the resulting bimodal fit (see Appendix B).

Given the fitted GMM, we estimate the gating threshold using two criteria. First, we approximate the mean confidence of correct responses using the mean of the higher-confidence GMM component, which serves as a surrogate for μ_{correct}. Second, to conservatively define an acceptance region in the absence of labels, we derive a posterior-based

Model	Method	GSM8K			MATH500			Omni-Math			GPQA-Diamond		
		Acc ↑	TF ↓	Acc/TF ↑	Acc ↑	TF ↓	Acc/TF ↑	Acc ↑	TF ↓	Acc/TF ↑	Acc ↑	TF ↓	Acc/TF ↑
LLaMA-3.2-3B	pass@1	73.09	-	-	41.4	-	-	11.7	-	-	17.26	-	-
	SC (k=16)	83.93	12.31	6.82	54.8	25.31	2.17	16.1	44.99	0.36	25.38	30.62	0.83
	ESC (w=4)	83.93	6.81 (-44.6%)	12.32	54.8	21.41 (-15.4%)	2.56	16.1	43.00 (-4.4%)	0.37	25.38	25.89 (-15.4%)	0.98
	ASC	83.85	6.27 (-49.0%)	13.37	55.0	20.13 (-20.4%)	2.73	15.8	41.84 (-7.0%)	0.38	25.38	25.27 (-17.5%)	1.00
	ReASC (offline)	83.85	4.38 (-64.4%)	**19.14**	55.0	18.27 (-27.8%)	3.01	-	-	-	-	-	-
	ReASC (online)	83.85	5.09 (-58.7%)	16.47	53.4	16.75 (-33.8%)	**3.19**	15.3	34.90 (-22.4%)	**0.44**	25.89	23.46 (-23.4%)	**1.10**
Qwen-2.5-3B	pass@1	83.32	-	-	61.8	-	-	21.3	-	-	24.37	-	-
	SC (k=16)	89.46	18.70	4.78	72.6	31.76	2.29	27.0	43.84	0.62	30.96	30.85	1.00
	ESC (w=4)	89.39	8.03 (-57.0%)	11.13	72.6	21.75 (-31.5%)	3.34	27.0	38.58 (-12.0%)	0.70	30.96	22.46 (-27.2%)	1.38
	ASC	89.39	7.57 (-59.5%)	11.80	72.6	20.84 (-34.4%)	3.48	27.2	36.69 (-16.3%)	0.74	30.96	21.26 (-31.1%)	1.46
	ReASC (offline)	89.46	5.70 (-69.5%)	**15.69**	71.8	16.80 (-47.1%)	**4.27**	-	-	-	-	-	-
	ReASC (online)	89.54	6.43 (-65.6%)	13.93	72.2	17.72 (-44.2%)	4.07	27.3	33.29 (-24.1%)	**0.82**	30.96	19.88 (-35.6%)	**1.56**
Gemma-3-4B	pass@1	88.40	-	-	63.2	-	-	25.2	-	-	21.32	-	-
	SC (k=16)	92.12	32.67	2.82	71.6	50.15	1.43	29.9	90.36	0.33	30.46	52.74	0.58
	ESC (w=4)	92.04	12.93 (-60.4%)	7.12	71.6	30.48 (-39.2%)	2.35	30.0	69.08 (-23.5%)	0.43	30.46	32.89 (-37.6%)	0.93
	ASC	92.12	12.26 (-62.5%)	7.52	71.6	28.68 (-42.8%)	2.50	30.2	65.90 (-27.1%)	0.46	30.46	31.73 (-39.8%)	0.96
	ReASC (offline)	92.04	9.45 (-71.1%)	**9.74**	71.4	25.59 (-49.0%)	**2.79**	-	-	-	-	-	-
	ReASC (online)	92.04	10.25 (-68.6%)	8.98	71.4	26.17 (-47.8%)	2.73	30.5	62.20 (-31.2%)	**0.49**	29.95	24.66 (-53.2%)	**1.21**
Qwen-2.5-7B	pass@1	90.90	-	-	71.2	-	-	27.6	-	-	27.92	-	-
	SC (k=16)	94.31	41.59	2.27	80.6	71.59	1.13	33.1	101.55	0.33	36.55	76.30	0.48
	ESC (w=4)	94.24	13.87 (-66.7%)	6.80	80.6	39.71 (-44.5%)	2.03	33.1	83.97 (-17.3%)	0.39	36.55	49.54 (-35.1%)	0.74
	ASC	94.24	13.40 (-67.8%)	7.04	80.8	37.25 (-48.0%)	2.17	33.3	80.24 (-21.0%)	0.41	36.55	45.63 (-40.2%)	0.80
	ReASC (offline)	94.09	10.43 (-74.9%)	**9.02**	81.2	29.26 (-59.1%)	**2.78**	-	-	-	-	-	-
	ReASC (online)	94.24	12.40 (-70.2%)	7.60	81.2	30.74 (-57.1%)	2.64	32.7	70.95 (-30.1%)	**0.46**	36.55	39.67 (-48.0%)	**0.92**
Gemma-3-27B	pass@1	95.60	-	-	77.6	-	-	37.5	-	-	35.53	-	-
	SC (k=16)	97.04	166.93	0.58	82.6	291.36	0.28	42.9	585.84	0.07	45.69	368.97	0.12
	ESC (w=4)	97.04	47.73 (-71.4%)	2.03	82.6	126.56 (-56.6%)	0.65	42.8	408.31 (-30.3%)	0.10	45.69	210.78 (-42.9%)	0.22
	ASC	97.04	47.71 (-71.4%)	2.03	83.8	121.29 (-58.4%)	0.69	43.0	384.90 (-34.3%)	0.11	45.69	199.62 (-45.9%)	0.23
	ReASC (offline)	96.89	29.36 (-82.4%)	**3.30**	83.6	101.32 (-65.2%)	**0.83**	-	-	-	-	-	-
	ReASC (online)	97.04	35.32 (-78.8%)	2.75	83.6	100.62 (-65.5%)	**0.83**	42.5	352.77 (-39.8%)	**0.12**	47.21	161.68 (-56.2%)	**0.29**

Table 1: Full comparison across mathematical and general reasoning benchmarks. Metrics include accuracy (Acc), average TFLOPs (TF), and compute efficiency (Acc/TF). The best Acc/TF is shown in bold, and the second best is underlined. TF reduction percentages (shown in red) are reported relative to SC.

threshold τ_{post} by identifying the smallest confidence value t for which the mixture posterior exceeds a target accuracy score p_{target}:

$$P(z = 1 \mid r = t) = \frac{\pi_1 \mathcal{N}(t; \mu_1, \sigma_1^2)}{\pi_1 \mathcal{N}(t; \mu_1, \sigma_1^2) + \pi_2 \mathcal{N}(t; \mu_2, \sigma_2^2)} \tag{8}$$

where π_j, μ_j, and σ_j^2 denote the weight, mean, and variance of component j, and $z = 1$ indicates membership in the higher-confidence component. The final gating threshold is defined as the maximum of the two estimates. The full online calibration procedure is summarized in Algorithm 2.

5 Experiments

5.1 Experimental Setup.

Datasets and Baselines. We evaluate ReASC on four reasoning benchmarks spanning mathematical and general-domain reasoning: GSM8K (Cobbe et al., 2021), MATH500 (Lightman et al., 2023), Omni-Math (Gao et al., 2024), and GPQA-Diamond (Rein et al., 2024), which requires expert-level knowledge and multi-step reasoning. We report reference results, including Pass@1 for single-sample performance and self-consistency (SC) (Wang et al., 2022) with a fixed budget of k=16. We further compare ReASC with several representative adaptive self-consistency baselines that dynamically adjust the sampling process. ASC (Aggarwal et al., 2023) uses a count-based Beta stopping rule. ESC (Li et al., 2024) performs early stopping when all responses within a fixed context window of size 4 converge to the same answer. For ReASC, we evaluate both offline and online settings, except for Omni-Math and GPQA-Diamond, where only online calibration is reported due to the absence of labeled validation sets. Additional details on datasets and baselines can be found in Appendix C.

Implementation Details. We conduct experiments using five instruction-tuned language models from multiple families and scales, including LLaMA-3.2 (3B) (Grattafiori et al., 2024), Qwen-2.5 (3B, 7B) (Yang et al., 2025), and Gemma-3 (4B, 27B) (Team et al., 2025), covering model sizes from 3B to 27B. For confidence calibration, we use a held-out set of k=128 instances, and both offline and online variants of ReASC adopt a target accuracy of $p_{\text{target}} = 0.9$, selected based on the accuracy-cost trade-off. Adaptive stopping uses a confidence threshold of $C_{\text{threshold}} = 0.95$ for the confidence-weighted Beta update, follow-

Model	GSM8K		MATH500	
	Accept Ratio	Accept Acc	Accept Ratio	Accept Acc
Offline				
LLaMA-3.2-3B	48.98	91.33	7.2	86.11
Gemma-3-4B	51.18	97.78	34.0	96.47
Qwen-2.5-7B	59.59	97.58	38.4	91.67
Gemma-3-27B	60.58	98.62	36.6	97.27
Online				
LLaMA-3.2-3B	28.13	93.53	17.6	88.28
Gemma-3-4B	33.28	98.41	28.2	98.58
Qwen-2.5-7B	37.91	97.40	31.8	93.08
Gemma-3-27B	40.63	99.44	36.2	97.31

Table 2: **Stage 1 acceptance ratio and accuracy.** Stage 1 resolves a substantial fraction of inputs, while preserving accuracy under both offline and online calibration.

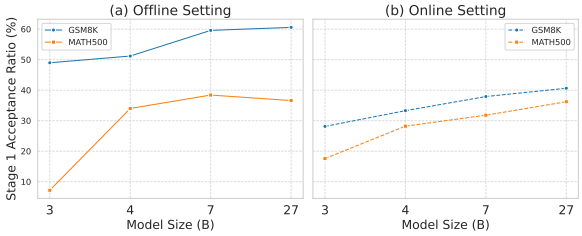

Figure 4: **Stage 1 acceptance ratio versus model size.** Acceptance increases with model scale across datasets.

ing the default ASC setting, with a scaling factor of $\lambda = 0.7$. We report accuracy, average inference cost (measured in TFLOPs), and their combined efficiency metric Acc/TF, which captures the trade-off between accuracy and computational cost, along with 95% confidence intervals for accuracy (Appendix F). TFLOPs are estimated following Kaplan et al. (2020) by approximating the compute required to generate $2N$ FLOPs per token, where N is the number of model parameters. Detailed hyperparameter settings are provided in Appendix G.

5.2 Main Result.

As shown in Table 1, ReASC achieves the strongest accuracy-cost trade-off across all five models and four benchmarks, as measured by Acc/TF. Specifically, ReASC consistently attains the lowest inference cost among self-consistency and adaptive baselines while preserving accuracy. For example, on GSM8K with Gemma-3-4B, ReASC reduces inference cost by approximately 70% relative to standard self-consistency, while consistently outperforming existing adaptive baselines in Acc/TF. These improvements are observed consistently across both offline and online settings, highlighting that ReASC remains effective for practical deployment even without offline calibration. Moreover, the same efficiency improvement persists across model scales ranging from 3B to 27B parameters, indicating the robustness and generalizability of ReASC. Together, these results show the effectiveness of reliability-aware evidence accumulation by enabling efficient sampling decisions.

Model	Method	GSM8K		MATH	
		Acc ↑	TF ↓	Acc ↑	TF ↓
LLaMA-3.2-3B	ASC (count)	84.99	5.99	55.42	19.46
	Ours (conf)	84.25	**3.95**	54.80	**14.75**
Qwen2.5-7B	ASC (count)	94.16	13.07	80.52	36.59
	Ours (conf)	94.04	**10.00**	**81.49**	**28.22**
Gemma3-27B	ASC (count)	96.92	47.76	85.48	123.63
	Ours (conf)	96.53	**28.02**	85.17	**101.25**

Table 3: **Stage 2 performance after Stage 1 filtering.** Reliability-aware accumulation reduces TF while preserving accuracy to count-based stopping.

5.3 Stage 1 Acceptance Analysis.

We begin by examining whether ReASC can reliably identify instances for which evidence accumulation is unnecessary. Specifically, we analyze the single-sample decision stage, which determines whether a reliable decision can be made from a single response. We measure the Stage 1 acceptance ratio, defined as the fraction of instances resolved after a single response, and the accuracy of accepted instances. As shown in Table 2, the acceptance ratio consistently increases with model size across datasets, while acceptance accuracy mostly remains above 90% under both offline and online calibration. Figure 4 further visualizes this trend, indicating that as model capability improves, Stage 1 reliably identifies instances that can be resolved from a single response.

5.4 Stage 2 Aggregation Analysis.

We next examine whether reliability-aware evidence accumulation can efficiently identify sufficient evidence when a single response is insufficient. To enable this analysis, we isolate the behavior of Stage 2 by reporting results only on instances not accepted at Stage 1 and comparing ReASC with ASC. As shown in Table 3, reliability-aware accumulation still consistently reduces inference cost relative to count-based stopping while preserving comparable accuracy. This suggests that while response counts provide a useful base signal, incorporating response-level confidence enables sufficient evidence to be identified with fewer samples when additional sampling is required.

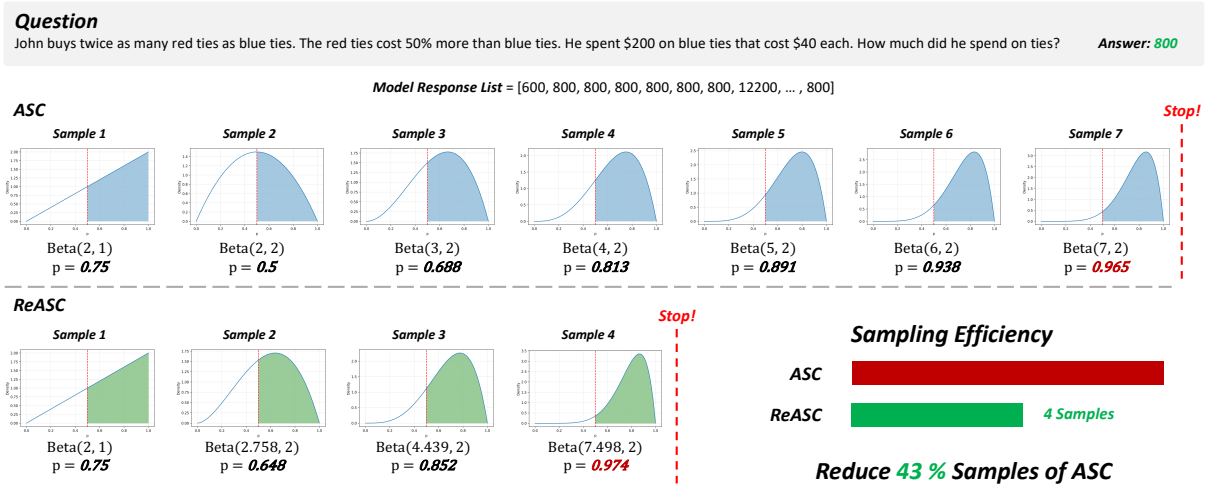

Figure 5: **Confidence-weighted update improves sampling efficiency.** Each sampled response updates a Beta posterior, shown as the shaded region in each plot. With sampling stopping at $p \geq 0.95$, ASC requires seven uniform updates, while ReASC reaches in four confidence-weighted updates, reducing sample cost by 43%.

5.5 Stage-wise Ablation Studies.

While the preceding analyses establish that each stage behaves as intended in isolation, they do not reveal how the two stages jointly contribute to the overall efficiency of ReASC. We therefore conduct a stage-wise ablation to disentangle the complementary contributions of the two stages. We compare three variants: **ASC**, which relies on count-based stopping; a **Stage 2 only** variant that applies confidence-weighted Beta update to all instances; and the full **ReASC** framework. As shown in Table 4, replacing count-based stopping with reliability-aware accumulation reduces inference cost relative to ASC by identifying sufficient evidence earlier. When comparing the Stage 2-only variant with ReASC, incorporating Stage 1 further reduces inference cost while preserving accuracy, indicating that evidence accumulation is unnecessary for a subset of instances in which a single response already provides reliable evidence. Together, these results demonstrate that the two stages play complementary roles: Stage 2 improves the efficiency of evidence accumulation when sampling is required, while Stage 1 avoids unnecessary sampling by identifying instances where a single response already provides sufficient evidence.

5.6 Confidence-Weighted Update Dynamics.

We present a qualitative analysis of a representative GSM8K instance with LLaMA-3.2-3B-Instruct to demonstrate how confidence-weighted Beta updates affect evidence accumulation. Figure 5 visualizes the Beta posteriors of Adaptive Consistency (ASC) and ReASC under the same sampled

Model	Method	GSM8K		MATH500	
		Acc ↑	TF ↓	Acc ↑	TF ↓
LLaMA-3.2-3B	ASC	83.85	6.27	55.00	20.13
	ReASC (Stage2 only)	**84.38**	5.33	**55.20**	18.76
	ReASC	83.85	**4.38**	55.00	**18.27**
Qwen2.5-7B	ASC	94.24	13.40	80.80	37.25
	ReASC (Stage2 only)	**94.24**	11.90	**81.20**	34.05
	ReASC	94.09	**10.43**	81.20	**29.26**

Table 4: **Stage-wise ablation of ReASC.** Stage 2 yields comparable accuracy than count-based stopping, while Stage 1 primarily reduces inference cost when applied.

responses. ASC aggregates evidence uniformly, resulting in a gradual shift of the Beta distribution and stopping after seven samples. In contrast, ReASC weights each update by response-level confidence, causing the posterior to concentrate more rapidly and reach the stopping threshold in four samples. As a result, ReASC reduces sample cost by 43% while converging to the same correct answer, enabling reliable decisions with fewer samples.

6 Conclusion

We present ReASC, a reliability-aware adaptive self-consistency framework that makes sampling decisions based on evidence sufficiency. By incorporating response-level reliability signals derived from model confidence, ReASC enables more effective evidence accumulation at test time. Notably, ReASC demonstrates superior accuracy-cost trade-offs over existing adaptive sampling methods across models and datasets. Our findings highlight the importance of incorporating response reliability into adaptive sampling and suggest a principled direction for future work on efficient test-time sampling.

Limitations

Our approach has several limitations. First, our current instantiation of ReASC estimates response-level reliability using model-derived confidence signals such as self-certainty. This choice is supported by prior findings and further validated by our experiments across multiple model families and datasets; however, the calibration of confidence signals may still vary across models and tasks. Second, as self-consistency is designed to elicit a model's existing knowledge through multiple samples, ReASC leverages response-level confidence to make more efficient sampling decisions, treating higher confidence as more reliable reasoning. While this assumption holds empirically across the benchmarks studied, it may be challenged in settings where models exhibit systematic overconfidence, suggesting that incorporating complementary reliability signals could further improve robustness. Finally, our work focuses on inference-time adaptation without additional training, prioritizing simplicity and broad applicability. While this design enables efficient deployment, incorporating learning-based approaches for reliability estimation could further improve accuracy and robustness, representing a promising direction for future work.

References

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and 1 others. 2023. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with llms. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12375–12396.

Hirotugu Akaike. 2003. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723.

Ding Chen, Qingchen Yu, Pengyuan Wang, Wentao Zhang, Bo Tang, Feiyu Xiong, Xinchi Li, Minchuan Yang, and Zhiyu Li. 2025. xverify: Efficient answer verifier for reasoning model evaluations. *arXiv preprint arXiv:2504.10481*.

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

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

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, and 1 others. 2024. Omni-math: A universal olympiad level mathematic benchmark for large language models. *arXiv preprint arXiv:2410.07985*.

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

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

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

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. 2024. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. *arXiv preprint arXiv:2401.10480*.

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

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Gideon Schwarz. 1978. Estimating the dimension of a model. *The annals of statistics*, pages 461–464.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. 2025. Confidence improves self-consistency in llms. *arXiv preprint arXiv:2502.06233*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2024. Soft self-consistency improves language model agents. *arXiv preprint arXiv:2402.13212*.

Xinglin Wang, Shaoxiong Feng, Yiwei Li, Peiwen Yuan, Yueqi Zhang, Chuyi Tan, Boyuan Pan, Yao Hu, and Kan Li. 2025. Make every penny count: Difficulty-adaptive self-consistency for cost-efficient reasoning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6904–6917.

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

Xuezhi Wang and Denny Zhou. 2024. Chain-of-thought reasoning without prompting. *Advances in Neural Information Processing Systems*, 37:66383–66409.

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

Tianhang Zhang, Lin Qiu, Qipeng Guo, Cheng Deng, Yue Zhang, Zheng Zhang, Chenghu Zhou, Xinbing Wang, and Luoyi Fu. 2023. Enhancing uncertainty-based hallucination detection with stronger focus. *arXiv preprint arXiv:2311.13230*.

A Derivation of the Beta-Based Stopping Criterion

We provide a brief derivation of the probability expression used in the Beta-based stopping rule. Suppose that V is the set of responses generated so far, and let v_1 and v_2 denote the number of samples assigned to the leading and second-best candidates.

Following the two-class reduction used in ASC, we consider the proportions (p_1, p_2) associated with these two candidates under continued sampling. Since $p_1 + p_2 = 1$, the event that the leading candidate remains dominant is equivalent to

$$p_1 > p_2 \quad \Longleftrightarrow \quad p_1 > \tfrac{1}{2}.$$

Under the Beta posterior induced by the accumulated counts in V, the distribution of p_1 is

$$p_1 \sim \mathrm{Beta}(\alpha, \beta), \qquad \alpha = v_1 + 1, \;\; \beta = v_2 + 1.$$

The dominance probability of interest is therefore

$$P(p_1 > p_2 \mid V) = P(p_1 > \tfrac{1}{2} \mid V).$$

The Beta(α, β) density is

$$f(t; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} t^{\alpha-1}(1-t)^{\beta-1},$$

yielding

$$P(p_1 > \tfrac{1}{2} \mid V) = \int_{1/2}^{1} \frac{1}{B(\alpha, \beta)} t^{\alpha-1}(1-t)^{\beta-1}\, dt.$$

The regularized incomplete Beta function is defined as

$$I_x(\alpha, \beta) = \frac{1}{B(\alpha, \beta)} \int_0^x t^{\alpha-1}(1-t)^{\beta-1}\, dt.$$

Applying this definition with $x = 1/2$ and the identity

$$\int_{1/2}^{1} f(t)\, dt = 1 - \int_0^{1/2} f(t)\, dt,$$

we obtain

$$P(p_1 > \tfrac{1}{2} \mid V) = 1 - I_{1/2}(\alpha, \beta).$$

which is the expression used in the stopping rule in Equation 6

Algorithm 2 Online Gating Threshold Calibration

Require: Unlabeled test set $\{x_i\}_{i=1}^n$, target accuracy p_{target}
Ensure: Gating threshold τ_{gate}

```
// (1) Compute confidence for each instance
for i = 1 to n do
    Generate response y_i
    Compute confidence S(y_i)
end for
// (2) Fit GMM and compute surrogate correct mean
Fit a 2-component GMM to {S(y_i)}
Let component c* be the one with the larger mean
μ_approx ← E_{r~c*}[r]   // surrogate correct mean
// (3) Posterior-based threshold search
Sort {S(y_i)} in ascending order as candidate thresholds
for each threshold t do
    Compute P(z = 1 | r = t) under the fitted GMM
    if P(z = 1 | r = t) ≥ p_target then
        τ_post ← t
        break
    end if
end for
τ_gate ← max(μ_approx, τ_post)
```

B AIC/BIC Analysis of Confidence Distributions

To justify the use of a two-component Gaussian Mixture Model (GMM) in the online calibration setting, we evaluate how well GMMs with $n \in \{1, 2, 3, 4\}$ components fit the unlabeled confidence-score distribution. For each dataset, we fit GMMs via the EM algorithm and compute the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC), which balance goodness of fit against model complexity (lower is better). As shown in Table 5, both AIC and BIC consistently select the two-component model across all settings, with clear improvements over a single-component model and no benefit from adding additional components. These results indicate that the confidence distribution is well captured by a bimodal mixture, supporting our use of a two-component GMM for online threshold estimation.

Model	Dataset	# Components k			
		1	2	3	4
Qwen2.5-3B	GSM8K	5051.12	**5004.18**	5010.70	5016.93
		5061.49	**5030.10**	5052.18	5073.96
	MATH500	2385.12	**2374.03**	2376.92	2383.82
		2398.55	**2395.10**	2410.64	2430.18
Gemma3-4B	GSM8K	7381.64	**7268.06**	7276.34	7280.79
		7392.01	**7305.21**	7309.53	7327.84
	MATH500	2962.88	**2950.26**	2954.60	2950.96
		2971.34	**2971.31**	2984.68	3005.46

Table 5: AIC/BIC scores for GMM with k components ($k = 1, 2, 3, 4$). Bold indicates the best (lowest) value.

Dataset	Answer Format	N_q	L_q	License
GSM8K	arabic number	1319	239.9	MIT License
MATH500	arabic number	500	195.9	MIT License
Omni-Math	arabic number	4428	270.8	Apache-2.0
GPQA-Diamond	option (A-D)	197	598.1	MIT License

Table 6: Relevant information of five datasets. N_q denotes the number of questions in each dataset. L_q denotes the average length of questions in each dataset.

C Datasets and Baselines

C.1 Datasets

We evaluate all methods on four reasoning benchmarks spanning mathematical and general-domain reasoning. Table 6 illustrates the statistics and the corresponding license information for each dataset. Below, we briefly describe each dataset and its evaluation protocol.

GSM8K. GSM8K is a grade-school-level mathematical reasoning benchmark consisting of 8,500 training and 1,319 test questions. Each question requires multi-step arithmetic reasoning expressed in natural language. Following standard practice, we evaluate accuracy based on the exact match of the final numerical answer extracted from the model output.

MATH500. MATH500 is a subset of the MATH benchmark designed to evaluate advanced mathematical problem-solving. It covers a diverse range of topics, including algebra, geometry, number theory, and calculus. We use the official test split of 500 problems and report the accuracy based on the verdict of the LLM Judge, namely xVerify (Chen et al., 2025).

Omni-Math. Omni-Math is a recently proposed large-scale benchmark for mathematical reasoning that emphasizes problem diversity and difficulty. It includes questions that require longer reasoning chains and compositional mathematical skills. We randomly sampled 1000 problems from the test set and report the accuracy based on the verdict of the LLM Judge, namely xVerify (Chen et al., 2025).

GPQA-Diamond. GPQA-Diamond is a general-domain reasoning benchmark curated to require expert-level knowledge and multi-step inference. Questions span scientific and technical domains and are intentionally designed to be difficult for non-expert models. We evaluate performance using exact-match accuracy against the answer choices.

C.2 LLM Inference Configuration.

For all experiments, we perform inference using the default generation configurations recommended for each model, without additional tuning. Specifically, LLaMA-3.2 is evaluated with temperature 0.6 and top-p 0.9; Qwen-2.5 uses temperature 0.7, top-p 0.8, and top-k 20; and Gemma-3 adopts temperature 1.0, top-p 0.95, and top-k 64. This design choice ensures that performance differences primarily reflect the behavior of adaptive sampling strategies rather than model-specific decoding heuristics.

C.3 Baselines

We compare ReASC against several representative inference-time baselines that differ in their sampling and stopping strategies.

Pass@1. Pass@1 reports the base model performance using a single sampled response. This serves as a lower-bound reference for sampling-based methods.

Self-Consistency (SC). Self-Consistency aggregates multiple independently sampled reasoning trajectories and selects the most frequent final answer. We use a fixed sampling budget of k=16 for all datasets.

Adaptive Consistency (ASC). ASC is an adaptive self-consistency method that dynamically determines when to stop sampling based on a count-based Beta stopping criterion. All sampled responses are treated as equally informative, and sampling terminates once the Beta posterior exceeds $C_{threshold} = 0.95$.

Early-Stopping Self-Consistency (ESC). ESC performs early stopping when the model produces identical answers within a fixed context window. We use a window size w of 4, as suggested in the original work.

D Analysis of Confidence Mapping Design

We study how different confidence mapping functions affect the accuracy-cost trade-off in our adaptive sampling framework. We consider the following alternative designs, each of which maps a normalized confidence score.

Mean-normalized aggregation. We consider a linear aggregation of normalized confidence scores,

$$v(y_i) \leftarrow v(y_i) + \frac{S(y_i)}{\mathbb{E}[\mathcal{D}_{val}]}.$$

Model	Mapping	GSM8K Acc/TF	MATH Acc/TF
LLaMA-3.2-3B	ASC (count-based)	13.37	2.73
	mean	13.92	2.86
	sigmoid	*10.32*	*2.60*
	exponential	14.74	2.84
	ours	**15.83**	**2.87**
Qwen-2.5-7B	ASC (count-based)	7.03	2.17
	mean	6.93	2.14
	sigmoid	*3.54*	*1.55*
	exponential	6.60	2.12
	ours	**7.60**	**2.38**
Gemma-3-27B	ASC (count-based)	2.03	0.69
	mean	2.04	0.67
	sigmoid	*1.10*	*0.42*
	exponential	2.04	0.67
	ours	**2.49**	**0.73**

Table 7: **Analysis of Confidence Mapping Design.** Results on LLaMA-3.2-3B, Qwen-2.5-7B, and Gemma-3-27B models. The proposed mapping consistently achieves the best accuracy-compute trade-off.

This mapping applies proportional vote increments without non-linear scaling.

Sigmoid-based mapping. Another alternative applies a sigmoid transformation,

$$v(y_i) \leftarrow v(y_i) + \sigma(\lambda z(y_i)),$$

which compresses confidence scores into a bounded range.

Unbounded exponential mapping. We also evaluate an exponential mapping without a lower bound,

$$v(y_i) \leftarrow v(y_i) + \exp(\lambda z(y_i)).$$

Proposed mapping. Finally, we propose an exponential mapping with a lower bound,

$$v(y_i) \leftarrow v(y_i) + \max(1, \exp(\lambda z(y_i))).$$

We also compared the alternative designs with ASC, which represents the count-based design. To isolate the effect of confidence mapping, we report Acc/TF as the primary metric. Results on three models (LLaMA-3.2-3B, Qwen-2.5-7B, and Gemma-3-27B) are shown in Table 7. While all variants achieve comparable accuracy, their efficiency differs substantially. Sigmoid-based mappings compress confidence scores to the range [0, 1], leading to slower vote accumulation and higher computational cost. In contrast, exponential mappings better differentiate high-confidence responses, enabling earlier stopping. Among them,

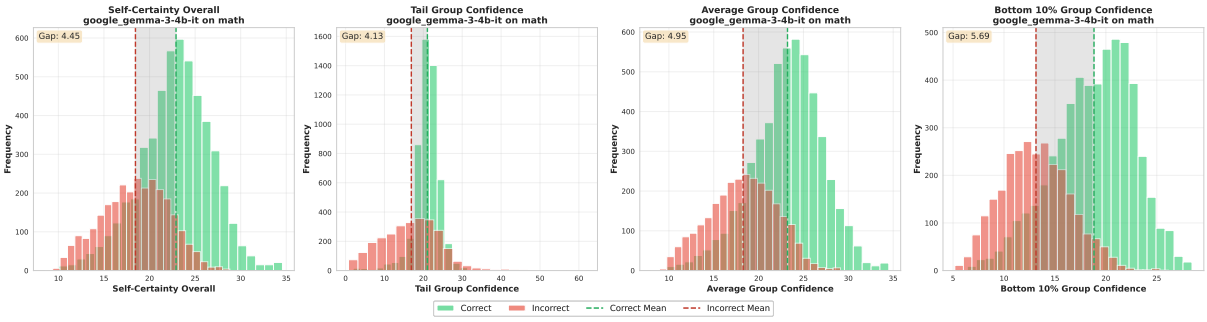

Figure 6: **Comparison of four self-certainty variants on a representative MATH instance using Gemma-3-4B-it.** Among the variants, Bottom 10% Group Confidence yields the largest gap between correct and incorrect responses, providing the clearest separation.

Confidence Metric	AUROC	Gap
Response-level Self-Certainty	0.801	4.45
Tail Group Confidence	0.699	4.13
Average Group Confidence	0.823	4.95
Bottom 10% Group Confidence	**0.860**	**5.69**

Table 8: AUROC and gap between correct and incorrect means comparison of confidence metrics for distinguishing correct and incorrect responses.

	AUROC	
Window Size	**Gemma-3-4B-it**	**Qwen-2.5-3B-it**
32	0.832	0.714
64	0.853	0.728
128	**0.874**	**0.744**
256	0.874	0.741
512	0.867	0.736
768	0.862	0.725

Table 9: AUROC sensitivity of Bottom 10% Group Confidence to sliding window group size on the MATH.

introducing a lower bound yields the most stable improvement across model scales, leading to the strongest accuracy-cost trade-off.

E Analysis on Confidence Score Design

E.1 Impact of Confidence Score

We analyze several confidence metrics to assess how reliably they distinguish correct from incorrect responses, including response-level self-certainty, tail-group confidence, average-group confidence, and Bottom 10% Group Confidence based on the confidence design choice from Fu et al. (2025). We evaluate each metric using two complementary criteria: (i) the separation gap between the mean confidence of correct and incorrect responses, and (ii) the area under the ROC curve (AUROC), which measures ranking-based discriminative performance. As shown in Figure 6, Bottom 10% Group Confidence exhibits the most significant separation gap, indicating clearer distributional separation between correct and incorrect responses. Consistent with this observation, AUROC results reported in Table 8 show that Bottom 10% Group Confidence achieves the strongest discriminative performance among the compared confidence metrics. Together, these results support our choice of Bottom 10% Group Confidence as the confidence signal in ReASC.

E.2 Sensitivity to Group Size.

We further analyze the sensitivity of the Bottom 10% Group Confidence to the sliding-window group size by evaluating its discriminative performance using AUROC. As shown in Table 9, AUROC remains relatively stable over a wide range of group sizes across both Gemma-3-4B-it and Qwen-2.5-3B-it on the MATH dataset, indicating that the metric is not overly sensitive to this hyperparameter. Among the tested values, a group size of 128 consistently yields the strongest or near-strongest separation between correct and incorrect responses. Based on this robustness-performance trade-off, we use a group size of 128 in all experiments.

F Confidence Interval Analysis

To assess the statistical robustness of the reported accuracy, we compute 95% confidence intervals using a non-parametric bootstrap over test instances. For each method, we collect the verdict for each test instance and generate 2,000 bootstrap resamples by sampling instances with replacement. Accuracy is computed for each resample, and the 95% confidence interval is obtained from the 2.5 and 97.5 percentiles of the resulting distribution.

Model	Method	GSM8K 95% CI	MATH500 95% CI	Omni-Math 95% CI	GPQA-Diamond 95% CI
LLaMA-3.2-3B-Instruct	Maj@k (k=16)	[81.96, 85.90]	[50.40, 59.20]	[13.81, 18.42]	[19.29, 31.47]
	ESC (w=4)	[81.96, 85.90]	[50.40, 59.20]	[13.81, 18.42]	[18.78, 30.46]
	ASC	[81.88, 85.82]	[50.60, 59.60]	[13.61, 18.12]	[19.29, 31.47]
	ReASC (offline)	[81.80, 85.82]	[50.60, 59.20]	–	–
	ReASC (online)	[81.73, 85.75]	[49.20, 58.20]	[13.01, 17.42]	[19.80, 31.98]
Qwen-2.5-3B-Instruct	Maj@k (k=16)	[87.87, 91.05]	[68.60, 76.20]	[24.42, 30.03]	[24.86, 37.56]
	ESC (w=4)	[87.72, 90.98]	[68.60, 76.20]	[24.42, 30.03]	[22.84, 36.04]
	ASC	[87.72, 90.98]	[68.60, 76.20]	[24.62, 30.13]	[24.86, 37.56]
	ReASC (offline)	[87.88, 90.75]	[68.00, 75.60]	–	–
	ReASC (online)	[87.72, 90.98]	[67.80, 75.40]	[24.72, 30.23]	[23.86, 37.06]
Gemma-3-4B-it	Maj@k (k=16)	[90.67, 93.48]	[67.60, 75.60]	[27.13, 32.83]	[23.86, 36.56]
	ESC (w=4)	[90.60, 93.48]	[67.60, 75.60]	[27.13, 32.93]	[23.86, 36.56]
	ASC	[90.67, 93.48]	[67.60, 75.40]	[27.43, 33.14]	[23.86, 36.56]
	ReASC (offline)	[90.52, 93.40]	[67.40, 75.40]	–	–
	ReASC (online)	[90.60, 93.48]	[67.20, 75.00]	[27.63, 33.33]	[23.35, 36.56]
Qwen-2.5-7B-Instruct	Maj@k (k=16)	[93.03, 95.53]	[77.20, 84.00]	[30.23, 36.24]	[29.44, 43.15]
	ESC (w=4)	[92.95, 95.38]	[77.20, 84.00]	[30.23, 36.24]	[28.43, 41.62]
	ASC	[92.95, 95.38]	[77.40, 84.20]	[30.43, 36.44]	[29.44, 43.15]
	ReASC (offline)	[92.80, 95.30]	[77.40, 84.20]	–	–
	ReASC (online)	[92.80, 95.30]	[78.00, 84.60]	[30.03, 36.24]	[29.44, 43.15]
Gemma-3-27B-it	Maj@k (k=16)	[96.06, 97.95]	[79.40, 85.80]	[40.04, 46.05]	[38.58, 52.79]
	ESC (w=4)	[96.06, 97.95]	[79.40, 85.80]	[39.94, 45.85]	[31.98, 45.69]
	ASC	[96.06, 97.95]	[80.60, 86.60]	[40.04, 46.05]	[38.58, 52.79]
	ReASC (offline)	[95.91, 97.80]	[80.20, 86.40]	–	–
	ReASC (online)	[96.06, 97.95]	[80.20, 86.40]	[40.04, 45.85]	[40.09, 54.31]

Table 10: 95% confidence intervals for accuracy computed via non-parametric bootstrap over test instances.

		Qwen2.5-3B			Gemma-3-4B		
Regime	p_{target}	**Acc**	**TF**	**Acc/TF**	**Acc**	**TF**	**Acc/TF**
Offline	0.9	72.0	17.19	**4.19**	71.8	27.18	**2.64**
	0.95	72.4	17.37	4.17	72.0	28.41	2.53
	0.99	72.8	20.01	3.64	72.0	29.41	2.45
Online	0.9	72.4	18.06	**4.01**	71.8	27.70	**2.59**
	0.95	72.0	18.79	3.83	71.8	27.94	2.57
	0.99	72.8	19.97	3.65	72.0	28.74	2.51

Table 11: **Sensitivity analysis of the target accuracy p_{target} for ReASC.** Results are reported in terms of accuracy (Acc), average TFLOPs (TF), and their efficiency ratio (Acc/TF) under both offline and online regimes.

G More Ablation Studies.

G.1 Selecting target accuracy

We conduct a hyperparameter sensitivity analysis on the target confidence threshold p_{target} using the Math500 dataset. Experiments are performed with two representative models, Qwen2.5-3B-Instruct and Gemma-3-4B-it, under both offline and online regimes of ReASC. We vary p_{target} across {0.9, 0.95, 0.99} and evaluate the resulting trade-off between accuracy and computational cost. As shown in Table 11, higher values of p_{target} generally lead to increased sampling and higher inference cost, while providing only marginal accuracy improvements. Across both models and regimes, $p_{\text{target}} = \mathbf{0.9}$ consistently achieves the best accuracy–compute trade-off, and we therefore adopt this setting in all main experiments.

G.2 Analysis of Calibration set size

We study the sensitivity of ReASC to the size of the calibration set used for threshold selection. Figure 7 shows accuracy and average TFLOPs as a function of calibration set size for LLaMA-3.2-3B-Instruct and Qwen-2.5-7B-Instruct. Across both models, accuracy improves with calibration size up to 128 examples and then saturates, showing negligible differences for larger sets. In contrast, the average TFLOPs increase monotonically as the calibration size grows, reflecting higher calibration overhead. These results indicate that a calibration size of 128 achieves the best trade-off between accuracy and inference cost, and we therefore use this value throughout our experiments.

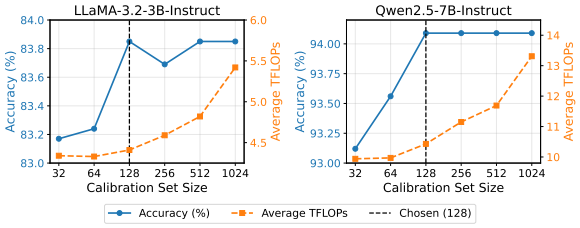

Figure 7: **Accuracy (left) and average TFLOPs (right) as a function of calibration set size.** Accuracy gains diminish beyond a calibration size of 128, whereas inference cost increases steadily with larger calibration sets. Based on this trade-off, we use a calibration size of 128 in all experiments.

G.3 Selecting λ for Confidence-Weighted Updates.

We conduct a sensitivity study on the confidence-weighting parameter λ using the GSM8K dataset with the LLaMA-3.2-3B-Instruct model. We evaluate $\lambda \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and report both accuracy and average TFLOPs. In Figure 8, the average TFLOPs consistently decrease as λ increases, indicating more aggressive evidence accumulation. Accuracy peaks at $\lambda = 0.7$, while larger values yield marginal cost reductions at the expense of accuracy. Considering both accuracy and computation, we select $\lambda = 0.7$ as it provides the most favorable accuracy–cost trade-off.

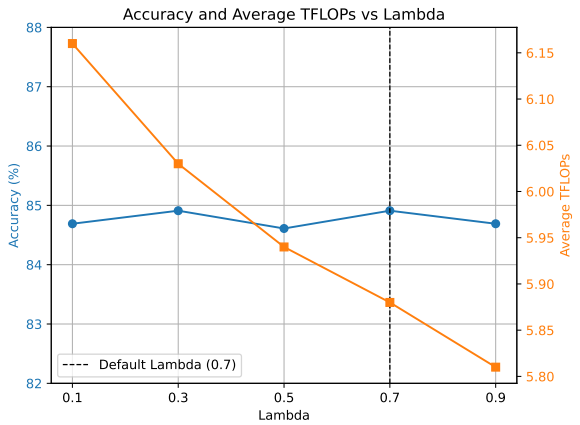

Figure 8: **Accuracy (left) and average TFLOPs (right) versus λ on GSM8K with LLaMA-3.2-3B-Instruct.** While TFLOPs decrease with larger λ, accuracy remains stable and peaks at $\lambda = 0.7$, which we set as the default.

H Use of AI Tools

During the preparation of this paper, AI tools (e.g., OpenAI's ChatGPT) were used in a limited, supporting capacity. Specifically, they assisted in enhancing the clarity and fluency of the text and in suggesting relevant keywords during the writing process. All conceptual ideas, experimental designs, implementations, analyses, and final interpretations were developed entirely by the authors. The authors independently verified all cited references, and no citation was included solely based on AI-generated content. No private, unpublished, or sensitive information was shared with AI tools beyond what is explicitly described in this paper.