

000 001 002 003 004 005 CGES: CONFIDENCE-GUIDED EARLY STOPPING FOR 006 EFFICIENT AND ACCURATE SELF-CONSISTENCY 007 008 009

010 **Anonymous authors**
011 Paper under double-blind review
012
013
014
015
016
017
018
019
020
021

ABSTRACT

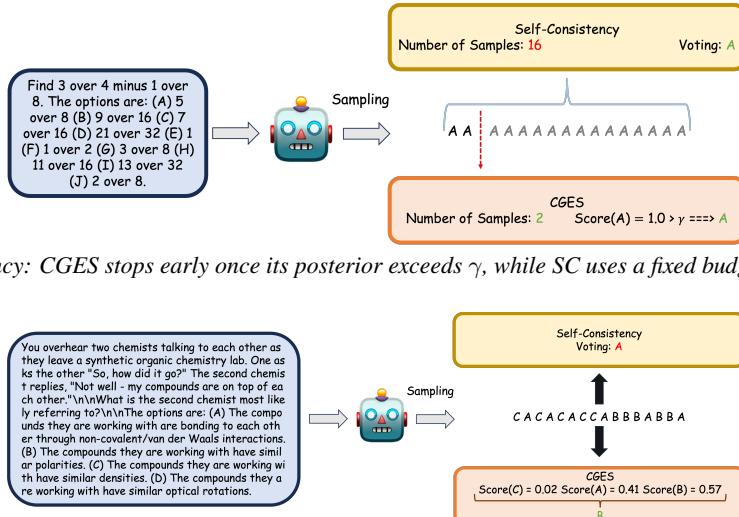
022 Large language models (LLMs) are often queried multiple times at test time,
023 with predictions aggregated by majority vote. While effective, this *self-*
024 *consistency* (Wang et al., 2023) strategy requires a fixed number of calls and fails
025 when the correct answer is infrequent. We introduce ***Confidence-Guided Early***
026 ***Stopping (CGES)***, a Bayesian framework that forms posteriors over candidate an-
027 swers from scalar confidence signals—derived from token probabilities or reward
028 models—and adaptively halts sampling once posterior mass exceeds a threshold.
029 We provide theoretical guarantees in both the ideal case of perfectly calibrated
030 confidences and the realistic regime with noisy confidences. Averaged over five
031 reasoning benchmarks, CGES reduces the average number of calls by **69.4%** (e.g.,
032 from 16.0 to 4.9) while maintaining accuracy within **0.06 percentage points** of
033 self-consistency.
034

035 1 INTRODUCTION 036

037 Large language models (LLMs) have achieved strong progress across reasoning, problem solving,
038 and open-domain tasks. A common way to improve reliability is *test-time scaling* (Snell et al., 2025),
039 a family of methods that allocate additional inference-time computation to improve performance.
040 One subset of these methods samples multiple responses and aggregates them into a final prediction.
041 Among the most widely used methods, self-consistency (SC) (Wang et al., 2023) aggregates outputs
042 by majority vote, leveraging the intuition that the most frequent answer across diverse generations is
043 likely to be correct. While simple and effective in many settings, majority-based aggregation suffers
044 from two major shortcomings. First, it assumes that response frequency is a faithful proxy for
045 correctness, which fails in cases where the correct answer appears infrequently. Second, it requires
046 a fixed number of model calls regardless of confidence, leading to substantial inefficiency.
047

048 Confidence signals offer an alternative perspective. Instead of depending solely on frequency, one
049 can incorporate confidence scores that capture the model’s belief in each response. These scores
050 may be derived from different sources. *Token probabilities* are taken directly from the model’s
051 output distribution and reflect how certain the model is about generating each token. *Calibration*
052 *schemes* adjust these raw probabilities so that they better match the actual likelihood of correctness,
053 turning overconfident or underconfident estimates into more reliable signals. *External reward models*
054 are trained separately, often with human feedback or domain-specific supervision, and provide
055 an independent measure of response quality beyond the model’s own probabilities. *Such signals can*
056 *distinguish between frequent but uncertain answers and rare yet confident ones.*
057

058 We propose a *confidence-based Bayesian framework* for test-time scaling. Our framework builds
059 on the idea that incorporating confidence enables robust aggregation and adaptive stopping, where
060 sampling stops once sufficient certainty is reached, reducing cost without sacrificing accuracy. Our
061 approach computes posterior probabilities over candidate answers by treating each response and its
062 associated confidence as probabilistic evidence. This yields two key advantages over majority voting:
063 (1) when the correct answer is frequent, our method reaches the same conclusion but often stops
064 earlier by exploiting high-confidence signals, improving efficiency; (2) when the correct answer is a
065 minority, our method can still recover it by amplifying the influence of confident predictions, where
066 majority voting fails. Figure 1 illustrates both phenomena on real items: our framework (a) halts
067 after only a few calls when confidence concentrates, and (b) outperforms majority vote when the
068 correct answer is a minority but highly confident. We further formalize this intuition by proving the-
069 oretical guarantees under ideal conditions where confidence scores are faithful to the true likelihood
070

(a) *Efficiency: CGES stops early once its posterior exceeds γ , while SC uses a fixed budget ($B=16$).*(b) *Accuracy: SC's majority vote is wrong, but CGES aggregates confidences and selects the correct answer.*Figure 1: **Examples of CGES vs. SC.** Top: early stopping with high confidence; Bottom: recovering a minority-but-confident answer.

of correctness. We then extend the analysis to the more practical case of noisy confidence estimates, where scores may be imperfect reflections of true correctness.

To operationalize this framework, we introduce *Confidence-Guided Early Stopping (CGES)*, which integrates Bayesian scoring with an adaptive stopping rule. CGES enables accuracy-efficiency trade-offs by halting once posterior concentration exceeds a threshold or a budget is reached. We evaluate CGES on AIME24, MATH500 (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), GPQA (Rein et al., 2024), and MMLU_Pro (Wang et al., 2024), comparing against self-consistency (Wang et al., 2023) and early-stopping self-consistency (Li et al., 2024). CGES consistently reduces LLM calls by large margins while maintaining or improving accuracy, demonstrating the value of calibrated confidence for test-time scaling and principled Bayesian aggregation. Our contributions are as follows:

- We propose a Bayesian framework that incorporates confidence estimates into self-consistency, enabling more accurate and theoretically grounded aggregation beyond majority voting.
- We design *Confidence-Guided Early Stopping (CGES)*, which adaptively halts sampling to trade off accuracy and efficiency.
- We establish theoretical guarantees of correctness under ideal conditions where confidence scores are perfectly calibrated to the true probabilities of correctness, and extend the analysis to the realistic setting with noisy confidence estimates.
- We empirically validate CGES across five reasoning benchmarks, showing substantial efficiency improvements while maintaining or improving accuracy.

2 RELATED WORK

A widely used approach for test-time scaling is *self-consistency*, introduced by Wang et al. (2023), which aggregates multiple reasoning paths by majority vote to improve chain-of-thought reliability. However, it requires a fixed number of model calls and can fail when the correct answer is infrequent. To reduce this cost, Li et al. (2024) proposed early-stopping self-consistency (ESC), which stops sampling once predictions agree. More recent extensions, such as Self-Calibration (Huang et al., 2025), use dynamic stopping rules or distill self-consistency signals into single-pass confidence estimates. In contrast, our work introduces a **Bayesian framework with theoretical guarantees for confidence-guided early stopping**.

Beyond self-consistency, several methods adaptively allocate test-time compute. Snell et al. (2025) study compute-optimal scaling strategies, and inference scaling laws characterize how performance improves with additional sampling, guiding compute-efficient inference (Wu et al., 2025). Wang

et al. (2025) pose test-time scaling as optimal resource allocation, showing trajectory-level allocation can be suboptimal and introducing direction-oriented rollouts to reduce redundancy across semantically similar samples. Muennighoff et al. (2025) propose s^1 , optimizing inference length under budget constraints. Other directions combine search and verification: self-enhanced tree search (Bi et al., 2025; Lample et al., 2022; Koh et al., 2024) expands multiple reasoning paths with sparse activation, while step-wise verifiers prune dynamically (Li et al., 2023; Lightman et al., 2024). Two-stage elimination approaches iteratively refine candidate answers (Chen et al., 2025), and query-variant ensembling improves robustness (Huang et al., 2024). These methods aim to balance accuracy and efficiency but differ in dependence on structured search, verifier signals, or semantic clustering. Our approach instead treats confidence as probabilistic evidence in a Bayesian model, enabling lightweight updates with formal consistency guarantees. Unlike methods focused on extracting a single answer or reasoning path, our framework samples multiple responses and then aggregates them via principled Bayesian inference.

A core component of our framework is *uncertainty estimation*, which has been studied extensively in generative LLMs. Probability-based methods such as length-normalized scoring (Malinin & Gales, 2021) mitigate biases toward shorter responses, while recent work incorporates meaning or trainable scoring. MARS (Bakman et al., 2024) adds semantics-aware token weighting, and LARS (Yaldiz et al., 2025) formulates supervised scoring over token-level features. Information-theoretic metrics distinguish epistemic and aleatoric uncertainty via iterative prompting (Abbasi-Yadkori et al., 2024). Similarity-based UQ methods such as SIMBA (Bhattacharjya et al., 2025) aggregate pairwise similarities for non-verbalized, largely black-box confidence estimates. Surveys (Geng et al., 2024) and unified taxonomies like the Uncertainty Estimation Codex (Xiao et al., 2025) provide overviews, and Bayesian or distillation-based approaches enable efficient estimation (Vejendla et al., 2025). Our goal is not to introduce new UQ methods; instead, we adapt existing techniques within the CGES framework, modifying them when necessary.

3 CONFIDENCE-BASED APPROACH

In this section, we propose a confidence-based method as an alternative to majority-vote self-consistency approach. We formalize the setting, introduce a Bayesian framework, provide theoretical guarantees under ideal conditions, and address realistic cases with noisy confidence scores.

3.1 PROBLEM SETTING AND NOTATIONS

Suppose we query a large language model (LLM) multiple times with a given query Q , whose true answer is A , thereby obtaining a sample set \mathcal{S} of responses. Let $\mathcal{U} = \{a_1, a_2, \dots, a_K\}$ denote the complete set of all possible distinct candidate *final* answers from the LLM, where K is the maximum number of such candidates. For the theoretical analysis here, we condition on the true answer A appearing in the candidate set, i.e., there exists $I \in [K]$ with $a_I = A$, and use I to denote the unknown correct index. If $A \notin \mathcal{U}$, no method operating only on $\{a_1, \dots, a_K\}$ can recover A : it must abstain or pick an incorrect candidate. In such cases, CGES still returns the highest-posterior candidate in \mathcal{U} , but the consistency guarantees in Sections 3.4–3.5 do not apply. Conditioned on (Q, A) and the identity of the correct candidate I , there exists a probability distribution over answers $P = (P_1, \dots, P_K) \sim f_P(Q, A, I)$, where $P_j = \mathbb{P}[R = a_j \mid Q, A, I]$. Intuitively, this captures the stochastic behavior of the LLM given the query: the likelihood of producing each candidate answer depends both on the query and on which candidate is correct. For each LLM call $t = 1, 2, \dots$, we draw a response–confidence pair (R_t, C_t) according to $R_t \sim P$ and $C_t \sim f_{C|P}(P)$. Here R_t denotes the *final textual answer string* produced by the LLM on call t (e.g., “12”, “ $x = 3$ ”, “yes”), after any post-processing or extraction step. Thus R_t takes values in the finite candidate set \mathcal{U} , and is not the full sequence of generated tokens. The confidence signal C_t is a scalar attached to R_t and serves as a (possibly noisy) proxy for its probability of correctness. This structure can be represented as a graphical model, illustrated in Fig. 2.

Idealistic vs. Realistic Assumptions. It is useful to distinguish two types of assumptions. *Idealistic assumptions* are simplifying conditions enabling a clean, efficient algorithm and optimality

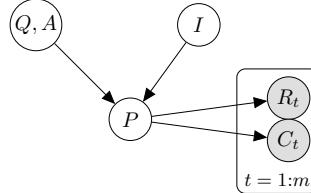


Figure 2: Graphical model for the sampling process in our theoretical analysis.

under ideal conditions (Theorem 1); these include a uniform error distribution and confidence scores independent of the true index. *Realistic assumptions* are expected to hold in practice and support both the CGES construction and guarantees under realistic conditions (Theorem 2); these include i.i.d. sampling assumption and a uniform prior on I . In the idealistic model, each C_t is treated as an *exact* conditional correctness probability for R_t , encoded in the likelihood (1). Under realistic conditions, C_t is only a *noisy estimator* of this probability, generated by some $f_{C|P}$ (e.g., from token probabilities or a reward model), and Theorem 2 characterizes when consistency still holds. Thus, while all four assumptions 1–4 hold in the idealized setting, only the weaker pair 1–2 is retained in realistic regimes.

The four assumptions are as follows:

1. Given a fixed query Q , with a fixed \mathcal{U} and P , the samples $\{(R_t, C_t)\}_{t=1}^m$ are i.i.d.
2. $I \sim \text{Uniform}(\{1, \dots, K\})$. This prior over the *index of the correct candidate* treats all K hypotheses symmetrically, reflecting arbitrary labeling of \mathcal{U} .
3. Under hypothesis $I = i$, the correct answer a_i is emitted with probability C_t , while each incorrect candidate shares the residual probability mass uniformly:

$$\mathbb{P}(R_t = a_i \mid C_t, I = i) = C_t, \quad \mathbb{P}(R_t = a_j \neq a_i \mid C_t, I = i) = \frac{1 - C_t}{K - 1}. \quad (1)$$

4. The confidence score C_t is independent of the index of the correct answer. Intuitively, the confidence attached to a sample should not depend on which candidate happens to be correct. Formally, $\mathbb{P}(C_t \mid I = i) = \mathbb{P}(C_t \mid I = j) \quad \forall i, j \in \{1, \dots, K\}$.

Remark 1 (Interpretation of Assumptions 1–4). *For a fixed question Q , the LLM with a chosen confidence-estimation method induces a joint distribution over answer strings $R_t \in \mathcal{U}$ and confidences $C_t \in (0, 1)$. Assumption 1 treats repeated calls as i.i.d. draws from this distribution, a standard call-level abstraction in self-consistency and test-time scaling analyses (Snell et al., 2025; Wu et al., 2025) enabling the law-of-large-numbers arguments in Theorems 1 and 2. Assumption 2 imposes a symmetric prior over the index I of the correct candidate, reflecting arbitrary labeling of \mathcal{U} . Assumption 3 specifies a one-versus-rest idealized likelihood parameterized by C_t , yielding a closed-form Bayesian update and the consistency result in Theorem 1. Assumption 4 requires that, before observing R_t , the marginal distribution of C_t is invariant to relabelings of \mathcal{U} ; it does not preclude higher C_t on correct answers. Section 3.5 and Theorem 2 analyze a more realistic regime where data may deviate from this idealized likelihood, while CGES still uses it for scoring. Together, Assumptions 1–4 are standard modeling abstractions enabling clean, closed-form guarantees; the algorithm itself and the realistic analysis in Theorem 2 do not rely on these idealized conditions being exactly true in practice.*

Given these assumptions, the objectives of our framework are twofold: (i) to *identify the most probable index $i \in [K]$ corresponding to the true answer*, and (ii) to *quantify the level of confidence in this selection*.

3.2 BAYESIAN CONFIDENCE-BASED FRAMEWORK

We apply Bayesian inference to compute posterior probabilities. Given an unknown index $I \in \{1, \dots, K\}$ and observed response–confidence pairs, the posterior distribution over I is

$$\mathbb{P}(I = i \mid \text{Obs}) = \frac{\mathbb{P}(\text{Obs} \mid I = i) \mathbb{P}(I = i)}{\sum_{k=1}^K \mathbb{P}(\text{Obs} \mid I = k) \mathbb{P}(I = k)}.$$

Here, the numerator combines the likelihood of the observations under hypothesis $I = i$ and the prior $\mathbb{P}(I = i)$, while the denominator normalizes across all K competing hypotheses. The set of observations is defined as $\text{Obs} = \{R_1, C_1, R_2, C_2, \dots, R_m, C_m\}$, where R_t denotes the response at step t and C_t its associated confidence score. This alternating structure reflects how, at each trial, both the raw prediction and its confidence are incorporated. The likelihood terms $\mathbb{P}(\text{Obs} \mid I = i)$ are therefore specified by our auxiliary generative model (cf. Section 3.1) and need not coincide with the true black-box LLM distribution $\mathbb{P}_{\text{LM}}(\cdot \mid Q)$. Since we assume a uniform prior over all hypotheses, i.e., $\mathbb{P}(I = i) = 1/K$, Bayes’ rule simplifies to $\mathbb{P}(I = i \mid \text{Obs}) = \frac{\prod_t \mathbb{P}(R_t, C_t \mid I = i)}{\sum_{k=1}^K \prod_t \mathbb{P}(R_t, C_t \mid I = k)}$. This prior is purely a symmetric choice over the K indices and is independent of the actual response distribution induced by the LLM. This formulation emphasizes that the posterior is built by multiplying the contributions of each observation and then renormalizing across all indices.

Algorithm 1 SCORE: Confidence-Based Bayesian Normalization

Require: Candidate set $\mathcal{U} = \{a_1, \dots, a_K\}$; samples $\mathcal{S} = \{(R_t, C_t)\}_{t=1}^m$ with $R_t \in \mathcal{U}$ and $C_t \in (0, 1)$

- 1: **for all** $a_i \in \mathcal{U}$ **do**
- 2: $s_{a_i} \leftarrow \prod_{t:R_t=a_i} C_t \times \prod_{t:R_t \neq a_i} \frac{1-C_t}{K-1}$
- 3: **end for**
- 4: **return** $\text{score}(a_i) = s_{a_i}/Z$ for all a_i

According to Assumption 4, we have $\mathbb{P}(C_t \mid I = i) = \mathbb{P}(C_t \mid I = j)$, $\forall i, j \in \{1, \dots, K\}$. As a result, the joint distribution factorizes as $\mathbb{P}(R_t, C_t \mid I = i) = \mathbb{P}(R_t \mid C_t, I = i) \mathbb{P}(C_t)$, and the marginal $\mathbb{P}(C_t)$ cancels out in the numerator and denominator of Bayes' rule. This leads to the simplified posterior form in (2), where X_i represents the posterior mass for hypothesis i .

$$\mathbb{P}(I = i \mid \text{Obs}) \propto \frac{\prod_t \mathbb{P}(R_t \mid C_t, I = i)}{\sum_{k=1}^K \prod_t \mathbb{P}(R_t \mid C_t, I = k)} \triangleq X_i, \quad (2)$$

In the ideal scenario, when the response R_t matches the true answer a_i , it is selected with probability equal to the reported confidence C_t , while the remaining $(K-1)$ incorrect answers share the residual probability mass uniformly. Formally:

$$\mathbb{P}(R_t \mid C_t, I = i) = \begin{cases} C_t & \text{if } R_t = a_i, \\ \frac{1-C_t}{K-1} & \text{otherwise.} \end{cases}$$

This ideal calibration assumption is used in Theorem 1; the realistic setting in Theorem 2 relaxes it and treats C_t as a noisy estimator of this probability.

3.3 ALGORITHM FOR CONFIDENCE-BASED SCORING

Given sampled answers and their confidences $\mathcal{S} = \{(R_t, C_t)\}_{t=1}^m$ for a single question, our goal is to convert them into calibrated, comparable probabilities over the unique answer set $\mathcal{U} = \{a_t\}_{t=1}^K$. The Bayesian posterior of Eq 2 factorizes into a product of per-sample terms: if the hypothesis (answer) is a_i , then a sample R_t that outputs a_i has likelihood C_t , and any other answer has likelihood $(1 - C_t)/(K - 1)$. We therefore form an unnormalized score s_{a_i} by multiplying these terms across all samples and then normalize across candidates. Algorithm 1 implements this computation.

Algorithm 2 wraps SCORE into an adaptive loop that allocates test-time compute per question. We begin with one sample per question, compute posteriors with SCORE, and maintain the set of unresolved questions D_{rem} whose current top posterior is below a confidence threshold γ . At round $t = 2, \dots, B$, we query the LLM *only* for $n \in D_{\text{rem}}$, append the new (R_t^n, C_t^n) , recompute SCORE on that question's t samples, and remove it from D_{rem} as soon as its top posterior exceeds γ . The process stops when all questions are confident or the budget B is reached, returning the argmax label per question and the average number of LLM calls. If no candidate ever exceeds the threshold γ for a given question before the budget B is exhausted, CGES still returns the current argmax posterior label for that question. Thus, γ controls test-time compute per question rather than enforcing abstention; extending the framework with a “no-answer” option is left for future work.

3.4 THEORETICAL ANALYSIS OF ALGORITHM PERFORMANCE (IDEAL SCENARIO)

Theorem 1. *Under Assumptions 1, 2, 3, and 4, and provided the confidences are informative i.e., $\mathbb{P}(C_t = 1/K) = 0$, the Bayesian confidence-based aggregator identifies the correct answer with probability tending to one as the number of samples m grows:*

$$\mathbb{P} \left(\arg \max_{i \in [K]} X_i = I \right) \longrightarrow 1 \quad \text{as } m \rightarrow \infty.$$

In fact, $X_I \rightarrow 1$ and $X_k \rightarrow 0$ for all $k \neq I$ almost surely.

Proof Sketch. Fix any wrong index $k \neq I$ and compare the (log) likelihood of the observed samples under the hypotheses I vs. k . Each sample contributes a log-likelihood ratio (LLR) increment whose *expected* value is strictly positive whenever the confidence C_t deviates from the uninformative value

270 **Algorithm 2** Confidence-Guided Early Stopping (CGES)

271 **Require:** N questions; threshold γ ; calls budget B ; scoring routine SCORE
272 1: Initialize $\text{scores}_n \leftarrow \text{SCORE}(\{(R_1^n, C_1^n)\})$ for all $n \in [N]$
273 2: $D_{\text{rem}} \leftarrow [N]$; calls $\leftarrow N$
274 3: **for** $t = 2$ to B **do**
275 4: $D_{\text{rem}} \leftarrow \{n \in D_{\text{rem}} : \max_i \text{scores}_n[i] < \gamma\}$
276 5: **if** $D_{\text{rem}} = \emptyset$ **then break**
277 6: **end if**
278 7: Query LLM for each $n \in D_{\text{rem}}$; calls $\leftarrow |D_{\text{rem}}|$
279 8: **for all** $n \in D_{\text{rem}}$ **do**
280 9: $\text{scores}_n \leftarrow \text{SCORE}(\{(R_1^n, C_1^n), \dots, (R_t^n, C_t^n)\})$
281 10: **end for**
282 11: **end for**
283 12: **return** $\hat{y}_n = a_{\arg \max_i \text{scores}_n[i]} \forall n$, usage $\leftarrow \text{calls}/N$

284
285 1/ K . By the Strong Law of Large Numbers (SLLN), these positive-drift increments accumulate
286 linearly, so the total LLR diverges to $+\infty$. Hence the likelihood under the true index dominates
287 every competitor, forcing the normalized posterior X_1 to 1 and all others to 0. A complete formal
288 proof is provided in Appendix A

290 3.5 THEORETICAL ANALYSIS OF ALGORITHM PERFORMANCE (REALISTIC SCENARIO)

291 In contrast to the ideal case, where the data are generated by the same likelihood used by the ag-
292 ggregator, the realistic setting permits model mismatch: the observed answers R_t are drawn from an
293 unknown but fixed (per question) distribution $\mathbf{P} = (P_1, \dots, P_K)$, while the confidence signal C_t
294 is a noisy proxy produced by an estimator (for example token probabilities). The aggregator itself
295 retains the same one-versus-rest likelihood as in the ideal model; under these conditions, consistency
296 reduces to the sign of the average LLR drift μ_k defined below.

297 **Theorem 2** (Consistency under realistic confidence noise). *Assume 1 and 2. For a fixed question,
298 let $R_t \sim \mathbf{P} = (P_1, \dots, P_K)$ i.i.d., and $C_t \mid \mathbf{P} \sim f_{C|\mathbf{P}}(\cdot \mid \mathbf{P})$ i.i.d., with $C_t \in (0, 1)$ a.s. and
299 $\mathbb{E}[|\log C_t| + |\log(1 - C_t)|] < \infty$. The aggregator uses the one-versus-rest likelihood from the
300 ideal model. Let $\theta_t = (1 - C_t)/(K - 1)$ and define*

$$302 \mu_k = \mathbb{E}_{\mathbf{P}, C} \left[(P_1 - P_k) \log \left(\frac{C_t}{\theta_t} \right) \right], \quad k \neq 1.$$

303 If $\mu_k > 0$ for all $k \neq 1$, then $\mathbb{P}(\arg \max_{i \in [K]} X_i = 1) \rightarrow 1$ as $m \rightarrow \infty$ and, in fact, $X_1 \rightarrow 1$ and
304 $X_k \rightarrow 0$ almost surely. If $\mu_{k^*} < 0$ for some $k^* \neq 1$, then $X_1 \rightarrow 0$ almost surely.

305 **Proof Sketch.** As in the ideal case, fix any $k \neq 1$ and consider the log-likelihood ratio (LLR)
306 between hypotheses $I = 1$ and $I = k$. Under the realistic generator, (R_t, C_t) are drawn with
307 $R_t \sim \mathbf{P}$ while the aggregator evaluates likelihoods using C_t and $\theta_t = (1 - C_t)/(K - 1)$. The
308 per-sample LLR increment has conditional mean $\mathbb{E}[Y_k^{(t)} \mid \mathbf{P}, C_t] = (P_1 - P_k)(\log C_t - \log \theta_t)$.
309 Averaging over (\mathbf{P}, C_t) gives the drift μ_k . If $\mu_k > 0$, the Strong Law of Large Numbers implies the
310 cumulative LLR grows linearly to $+\infty$, so the likelihood under $I = 1$ dominates and the posterior
311 concentrates on the truth. This condition also covers minority-correct regimes ($P_1 < P_k$) provided
312 C_t is systematically below $1/K$, which flips the sign of the log term and yields positive drift where
313 majority vote would fail. A full proof appears in Appendix B.

314 **Remark 2** (Missing ground-truth among candidates). *Both Theorem 1 and Theorem 2 analyze the
315 behavior of CGES under the event that the true answer A is contained in the candidate set \mathcal{U} , so
316 that I is well-defined. If $A \notin \mathcal{U}$ for a given question, then any aggregation procedure that operates
317 only on \mathcal{U} inevitably fails to return A : the error probability has an irreducible component $\mathbb{P}(A \notin \mathcal{U})$ that no self-consistency-style scheme can eliminate. In this misspecified regime, CGES behaves like
318 other sampling-based methods: it selects the candidate in \mathcal{U} with the highest posterior mass, and the
319 consistency guarantees above no longer hold. In practice, a high threshold γ can be used to signal
320 such high-uncertainty cases (e.g., when no candidate accumulates substantial posterior probability),
321 but formal guarantees on correctness are impossible without access to the true answer.*

324

4 EXPERIMENTS

325
 326 We evaluate CGES on five reasoning benchmarks using two 7B-class models and compare against
 327 self-consistency (SC) (Wang et al., 2023), early-stopping self-consistency (ESC) (Li et al., 2024),
 328 and **Adaptive-Consistency (ASC)** (Aggarwal et al., 2023). **SC** aggregates multiple samples by
 329 majority vote. **ESC** halts once recent predictions align, reducing calls. **ASC** stops dynamically when
 330 the estimated majority is stable. We report *accuracy* of the final answer and *efficiency* as the average
 331 number of LLM calls per question. Results are averaged over three seeds. Decoding and prompting
 332 follow prior work (Appendix D). Confidence signals C_t use the strategies in Section 4.2.
 333

334

4.1 DATASETS AND MODELS

335 We evaluate on five benchmarks spanning mathematics and broad knowledge: **AIME24**, 30 prob-
 336 lems from the 2024 American Invitational Mathematics Examination; **MATH500**, a 500-question
 337 subset of MATH (Hendrycks et al., 2021) targeting advanced reasoning; **GSM8K** (Cobbe et al.,
 338 2021), 8,500 grade-school math word problems; **MMLU_Pro** (Wang et al., 2024), 14 college-level
 339 subjects; and **GPQA Diamond** (Rein et al., 2024), expert-written science questions challenging
 340 even for skilled humans. For the harder datasets (AIME24, GPQA), we use *DeepSeek-R1-Distill-*
 341 *Qwen(7B)* (DeepSeek-AI et al., 2025), and for the easier ones (MATH500, GSM8K, MMLU_Pro)
 342 we use *Qwen2.5(7B)* (Yang et al., 2025).
 343

344

4.2 CONFIDENCE ESTIMATION STRATEGIES

345 We compare several strategies for estimating the *scalar* confidence $C_t \in (0, 1)$ of each sampled an-
 346 swer R_t . Each strategy maps model outputs (token probabilities or reward scores) to a scalar C_t that
 347 we interpret as an estimate of the probability that the corresponding response R_t is correct. These
 348 estimates need not be perfectly calibrated; they serve as noisy confidence signals in the sense for-
 349 malized in Section 3.1 and Theorem 2. Our framework operates at the response level, but confidence
 350 estimation relies on finer token-level granularities. Specifically, let a response R_t consist of a token
 351 sequence T_1, \dots, T_L with associated autoregressive probabilities p_1, \dots, p_L . We use lowercase p_ℓ
 352 to denote these token-level probabilities from the underlying language model, and uppercase $\mathbb{P}(\cdot)$
 353 for probabilities over candidate answers $a_j \in \mathcal{U}$ as in Section 3.1. The role of the present section is
 354 to map token-level scores (or reward-model outputs) into a single scalar confidence C_t . Building on
 355 this, we now describe three token-based approaches and one verifier-based alternative.
 356

357 **Length-Normalized Scoring (LNS)** (Malinin & Gales, 2021). A natural way to quantify the likeli-
 358 hood of a response is by averaging over token probabilities. The *geometric mean* yields the standard
 359 length-normalized score $\text{LNS}_{\text{geom}} = \exp\left(\frac{1}{L} \sum_{\ell=1}^L \log p_\ell\right)$ while the *arithmetic mean* provides a
 360 simpler length-insensitive proxy, $\text{LNS}_{\text{arith}} = \frac{1}{L} \sum_{\ell=1}^L p_\ell$. We set C_t to either of these values and
 361 denote them in results as *LNS [Geometric mean]* and *LNS [Arithmetic mean]*.
 362

363 **MARS (Step-Weighted Scoring)** (Bakman et al., 2024). The MARS method generalizes LNS by
 364 assigning different weights to different positions in the sequence. Each token T_ℓ receives an expo-
 365 nent $w(R, Q, L, \ell) \triangleq \frac{1}{2L} + \frac{u(R, Q, \ell)}{2}$, where **R** denotes the full textual response string (suppressing
 366 the time index **t** for brevity), so the overall score becomes $\bar{P}(R | Q, \theta) = \prod_{\ell=1}^L p_\ell^{w(R, Q, L, \ell)}$ where θ
 367 denotes the parameters of the language model generating token probabilities. Here $u(R, Q, \ell)$ is an
 368 importance score for token T_ℓ , such as the semantic change in the output when masking that token.
 369 While token-level weighting can be precise, for long reasoning responses most token importance
 370 scores become nearly uniform, and computing $u(\cdot)$ for every token is expensive (requiring L calls
 371 to a semantic extractor model such as a sentence transformer). To address this, we adopt a *step-wise*
 372 variant of MARS: instead of per-token weights, we assign weights at the granularity of reasoning
 373 steps or sentence segments. Although the step-importance score is recomputed each iteration, this
 374 overhead (6 layers, ~ 50 M parameters) is negligible compared to the 7B-parameter inference model
 375 we query for R_t . We denote the resulting confidence as $C_t = \text{MARS}$.
 376

377 **Reward Model Confidence.** In addition to the above token-level methods, we also consider a
 378 model-based approach that evaluates entire responses directly. A trained reward model assigns a
 379 quality score to each R_t , which we use as C_t . In our study, we use *Qwen2.5-Math-PRM-72B*
 380 process reward model (Zhang et al., 2025), which outputs a scalar in $(0, 1)$ that correlates with
 381

378 Table 1: Accuracy (%) and avg. #Calls across five reasoning tasks; Avg is mean over benchmarks. Parentheses
 379 show difference vs. SC (#Calls=16 or SC Acc). For CGES, the first row corresponds to the *Efficient* setting
 380 (lower threshold, fewer calls) and the second row to the *Conservative* setting (higher threshold, more calls).¹

	AIME24		MATH500		GSM8K		GPQA		MMLU_Pro		Avg.	
	#Calls	Acc										
SC	16.00	78.89	16.00	82.20	16.00	94.39	16.00	51.01	16.00	61.54	16.00	73.61
ESC (w=4)	11.02	78.89	7.88	82.07	5.22	94.37	11.90	51.18	9.04	61.43	9.01	73.59
	(4.98)	(+0.00)	(-8.12)	(-0.13)	(-10.78)	(-0.02)	(-4.10)	(+0.17)	(-6.96)	(-0.11)	(-6.99)	(-0.02)
ESC (w=8)	14.40	78.89	11.57	82.20	9.50	94.39	14.87	51.01	12.94	61.54	12.66	73.61
	(-1.60)	(+0.00)	(-4.43)	(+0.00)	(-6.50)	(+0.00)	(-1.13)	(+0.00)	(-3.06)	(+0.00)	(-3.34)	(+0.00)
Adaptive-Consistency (BETA)	10.25	78.89	7.36	82.07	5.05	94.37	11.43	50.51	8.58	61.46	8.53	73.46
	(5.75)	(+0.00)	(-8.64)	(-0.13)	(-10.95)	(-0.02)	(-4.57)	(+0.50)	(-7.42)	(-0.08)	(-7.47)	(-0.15)
Adaptive-Consistency (Dirichlet)	11.17	78.89	7.88	82.07	5.22	94.37	12.18	50.51	9.13	61.46	9.12	73.46
	(4.83)	(+0.00)	(-8.12)	(-0.13)	(-10.78)	(-0.02)	(-3.82)	(+0.50)	(-6.87)	(-0.08)	(-6.88)	(-0.15)
CGES (Ours)												
LNS[Arithmetic mean]	6.47	78.89	4.69	81.93	4.50	94.26	2.09	51.13	6.77	61.56	4.90	73.55
	(9.53)	(+0.00)	(-11.31)	(-0.27)	(-11.50)	(-0.13)	(-13.91)	(+0.12)	(-9.23)	(+0.02)	(-11.10)	(-0.06)
	9.59	78.89	5.81	81.87	4.50	94.26	10.48	51.52	8.29	61.58	7.73	73.62
	(6.41)	(+0.00)	(-10.19)	(-0.33)	(-11.50)	(-0.13)	(-5.52)	(+0.51)	(-7.71)	(+0.04)	(-8.27)	(+0.01)
LNS[Geometric mean]	7.27	78.44	6.64	82.00	5.36	94.36	4.44	51.35	6.78	61.63	6.88	73.56
	(8.73)	(-0.45)	(-9.36)	(-0.20)	(-10.64)	(-0.03)	(-11.56)	(+0.34)	(-9.22)	(+0.09)	(-9.12)	(-0.05)
	11.56	78.44	6.64	82.00	5.36	94.36	13.26	51.18	10.68	61.65	9.90	73.52
	(4.44)	(-0.45)	(-9.36)	(-0.20)	(-10.64)	(-0.03)	(-2.74)	(+0.17)	(-5.32)	(+0.11)	(-6.10)	(-0.09)
MARS	5.79	77.78	6.80	81.93	5.39	94.42	3.61	52.69	6.68	61.60	5.65	73.28
	(-10.21)	(-1.11)	(-9.20)	(-0.27)	(-10.61)	(+0.03)	(-12.39)	(+1.68)	(-9.32)	(+0.06)	(-10.35)	(-0.33)
	10.83	77.78	6.80	81.93	5.39	94.42	12.14	50.84	10.59	61.53	9.15	73.52
	(5.17)	(-1.11)	(-9.20)	(-0.27)	(-10.61)	(+0.03)	(-3.86)	(-0.17)	(-5.41)	(-0.01)	(-6.85)	(-0.09)

401 Table 2: **CGES-PRM (upper bound)**. Same conventions as Table 1. Confidences from a large PRM
 402 (Qwen2.5-Math-PRM-72B; scoring only).

	AIME24		MATH500		GSM8K		GPQA		MMLU_Pro		Avg.	
	#Calls	Acc	#Calls	Acc	#Calls	Acc	#Calls	Acc	#Calls	Acc	#Calls	Acc
RM Confidence	6.71	77.78	4.32	83.00	2.57	94.49	6.62	51.68	6.05	62.17	5.27	73.42
	(-9.29)	(-1.11)	(-11.68)	(+0.80)	(-13.43)	(+0.10)	(-9.38)	(+0.67)	(-9.95)	(+0.63)	(-10.73)	(-0.19)
	7.65	77.78	5.27	85.13	3.02	95.55	10.62	50.67	7.44	63.64	6.40	74.35
	(-8.36)	(-1.11)	(-10.73)	(+2.93)	(-12.98)	(+1.16)	(-5.38)	(-0.34)	(-8.57)	(+2.10)	(-9.60)	(+0.74)

411 alignment to the ground truth on math-style reasoning. Because it has 72B parameters (far larger
 412 than our 7B inference model), using it as the scorer is *impractical* for deployment; we include it as
 413 a near-optimal reference to approximate an upper bound on confidence quality, especially on tasks
 414 that are in-domain for this PRM. We denote this variant as *RM Confidence*.

4.3 RESULTS

415 Table 1 reports accuracy and average number of calls for CGES and baselines with a Self-
 416 Consistency budget of $B = 16$. We compare probability-based variants of CGES, which use token-
 417 level (LNS) and step-level (MARS) confidence scores. Across all benchmarks, CGES variants using
 418 token- and step-level confidence (LNS and MARS) markedly reduce model calls compared to SC
 419 and ESC while maintaining near-identical accuracy. With the strongest efficiency gains, arithmetic-
 420 mean LNS averages **4.90 calls** (**69.4%** fewer than SC’s 16) with only -0.06% accuracy change.
 421 MARS further improves efficiency on harder tasks such as GPQA, lowering calls from **16.00 to 3.61**
 422 with a **+1.68%** accuracy gain. On MMLU_Pro, CGES attains comparable or better accuracy with
 423 fewer than half the calls (e.g., 6.77 vs. 16.00 with $+0.02\%$). These results show CGES outper-
 424 forms local majority-vote stopping rules, enabling adaptive early stopping from confidence signals.
 425 For easier tasks (MATH500, GSM8K), probability-based variants show minor accuracy dips despite
 426 large call savings, as improvements then depend on *calibration* of C_t ; noisy probability proxies
 427 (LNS/MARS) can be slightly over- or under-confident, limiting accuracy gains. Even so, efficiency
 428 gains remain substantial (often $> 2 \times$ fewer calls) with accuracy effectively preserved.

431 ¹The *Efficient* setting corresponds to the smallest γ for which CGES matches or surpasses SC performance, while the *Conservative* setting
 corresponds to the largest γ considered. If CGES does not reach SC performance, both settings coincide at the largest γ .

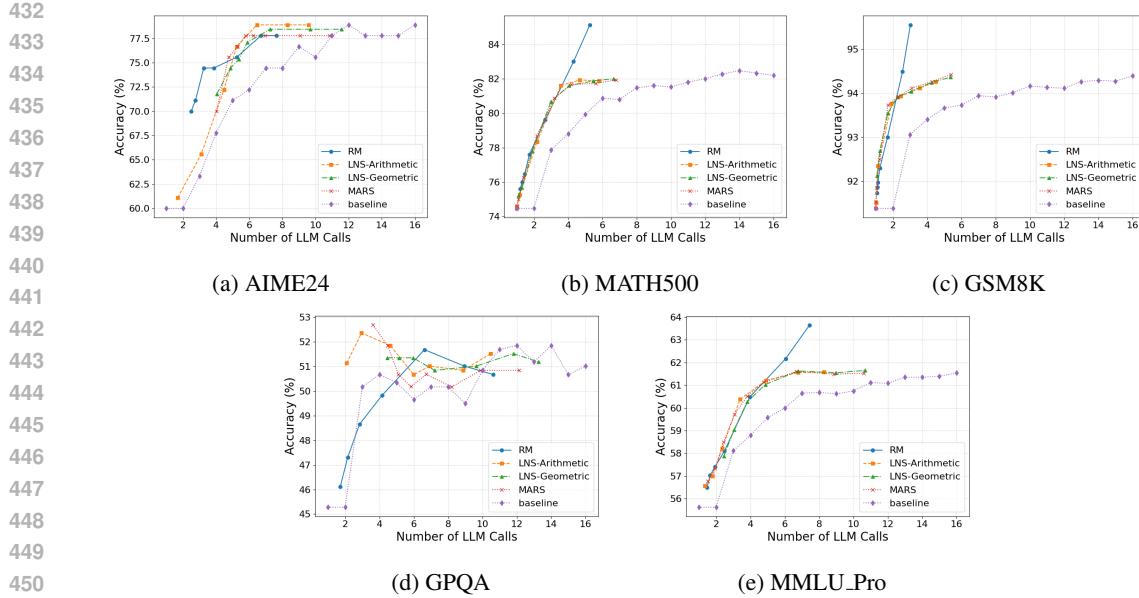


Figure 3: Accuracy vs. number of LLM calls ($B=16$) on AIME24 (a), MATH500 (b), GSM8K (c), GPQA (d), and MMLU_Pro (e). CGES achieves near-maximal accuracy with far fewer calls than self-consistency.

Reward-model-based CGES approximates a best-case scenario where confidence estimates are closer to ground truth than those from the 7B inference models. Since Qwen2.5-Math-PRM-72B is larger and trained for step-level scoring, its signals are stronger than what is practical at inference time. Improvements are most pronounced on math-centric datasets (MATH500, GSM8K), where the PRM’s training overlaps with the task: accuracy increases while calls drop sharply (Table 2). On AIME24 and GPQA, domain mismatch and higher difficulty limit PRM utility, yielding small accuracy drops despite fewer calls. On MMLU_Pro, the PRM still provides useful confidence signals and surpasses SC with substantially fewer calls. Appendix C reports additional results under smaller SC budgets ($B = 4, 8$), showing consistent trends, [as well as calibration analyses and large-model \(70B–72B\) experiments](#).

Figure 3 shows the accuracy–efficiency trade-off. Each CGES curve is obtained by sweeping the stopping threshold γ (from 0.7 to 0.9999); the *baseline* traces self-consistency (SC) at fixed budgets. On AIME24, GSM8K, and MATH500, CGES achieves near-maximal accuracy after only a few calls, whereas SC requires the full budget. **Reward-model (PRM) confidence** converges fastest (often within 3–4 calls), and is shown as a near-ideal reference rather than a deployable setting. Across datasets, curves flatten beyond ~ 6 calls, indicating diminishing returns and that SC’s $B=16$ is over-provisioned. These results confirm that confidence-guided stopping enables CGES to adaptively terminate sampling early without compromising accuracy.

5 CONCLUSIONS

We proposed CGES, a confidence-based Bayesian framework for test-time scaling of LLMs. By treating each response and its confidence as probabilistic evidence, CGES enables early stopping and more reliable aggregation than majority voting. Across five reasoning benchmarks, CGES substantially reduces model calls while maintaining or improving accuracy, outperforming Self-Consistency and early-stopping baselines. Our theoretical analysis further shows correctness under both ideal and noisy confidence assumptions. Overall, confidence-guided aggregation provides a principled solution within the family of test-time scaling methods that sample multiple responses and aggregate them into a single answer. Future work includes (i) developing more accurate confidence estimators to further enhance efficiency and accuracy, and (ii) predicting the required number of samples dynamically from confidence signals.

486

6 REPRODUCIBILITY STATEMENT

488 We have taken several steps to ensure the reproducibility of our work. All theoretical claims are
 489 stated under clearly enumerated assumptions (Section 3.1) with full proofs provided in Appen-
 490 dices A–B. The complete Bayesian formulation and algorithmic details of CGES are presented in
 491 Section 3.2–3.3, including pseudocode for both the scoring and stopping procedures (Algorithms 1–
 492 2). Experimental protocols are fully described in Section 4, covering datasets (Section 4.1), models,
 493 baselines, and confidence estimation strategies (Section 4.2). Detailed hyperparameters, decoding
 494 configurations, and additional results under varying budgets are included in Appendix C–D. To fur-
 495 ther facilitate verification, we provide an anonymized implementation and experiment scripts as
 496 supplementary material. Together, these resources ensure that both the theoretical and empirical
 497 results reported in this paper can be independently reproduced and validated.

498

499 REFERENCES

500 Yasin Abbasi-Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvari. To believe or not
 501 to believe your LLM: Iterativeprompting for estimating epistemic uncertainty. In *The Thirty-
 502 eighth Annual Conference on Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=k6iyUfwdI9>.

503

504 Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. Let’s sample step by step: Adaptive-
 505 consistency for efficient reasoning and coding with LLMs. In Houda Bouamor, Juan Pino, and
 506 Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Lan-
 507 guage Processing*, pp. 12375–12396, Singapore, December 2023. Association for Computational
 508 Linguistics. doi: 10.18653/v1/2023.emnlp-main.761. URL <https://aclanthology.org/2023.emnlp-main.761>.

509

510 Yavuz Faruk Bakman, Duygu Nur Yaldiz, Baturalp Buyukates, Chenyang Tao, Dimitrios Dimitri-
 511 adis, and Salman Avestimehr. MARS: Meaning-aware response scoring for uncertainty estima-
 512 tion in generative LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for
 513 Computational Linguistics (Volume 1: Long Papers)*, pp. 7752–7767, Bangkok, Thailand, August
 514 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.419. URL
 515 <https://aclanthology.org/2024.acl-long.419>.

516

517 Debarun Bhattacharjya, Balaji Ganesan, Junkyu Lee, Radu Marinescu, Katya Mirylenka, Michael
 518 Glass, and Xiao Shou. SIMBA UQ: Similarity-based aggregation for uncertainty quantifica-
 519 tion in large language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Car-
 520 olyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics:
 521 EMNLP 2025*, pp. 15880–15894, Suzhou, China, November 2025. Association for Compu-
 522 tational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.859. URL
 523 <https://aclanthology.org/2025.findings-emnlp.859>.

524

525 Zhenni Bi, Kai Han, Chuanjian Liu, Yehui Tang, and Yunhe Wang. Forest-of-thought: Scaling
 526 test-time compute for enhancing LLM reasoning. In *Forty-second International Conference on
 527 Machine Learning*, 2025. URL <https://openreview.net/forum?id=BMJ3pyYxu2>.

528

529 Yanxi Chen, Xuchen Pan, Yaliang Li, Bolin Ding, and Jingren Zhou. Simple and provable scaling
 530 laws for the test-time compute of large language models, 2025. URL <https://arxiv.org/abs/2411.19477>.

531

532 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,
 533 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John
 534 Schulman. Training verifiers to solve math word problems, 2021. URL <https://arxiv.org/abs/2110.14168>.

535

536 DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu,
 537 Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu,
 538 Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao
 539 Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan,
 Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao,
 Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding,

540 Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang
 541 Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai
 542 Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang,
 543 Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang,
 544 Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang,
 545 Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang,
 546 R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhua Chen, Shengfeng
 547 Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing
 548 Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen
 549 Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong
 550 Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu,
 551 Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xi-
 552 aoshua Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia
 553 Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng
 554 Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong
 555 Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong,
 556 Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou,
 557 Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying
 558 Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda
 559 Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu,
 560 Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu
 561 Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforce-
 562 ment learning, 2025. URL <https://arxiv.org/abs/2501.12948>.

563 Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koepll, Preslav Nakov, and Iryna Gurevych. A survey
 564 of confidence estimation and calibration in large language models. In Kevin Duh, Helena Gomez,
 565 and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter
 566 of the Association for Computational Linguistics: Human Language Technologies (Volume 1:
 567 Long Papers)*, pp. 6577–6595, Mexico City, Mexico, June 2024. Association for Computational
 568 Linguistics. doi: 10.18653/v1/2024.nacl-long.366. URL <https://aclanthology.org/2024.nacl-long.366/>.

569 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn
 570 Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset.
 571 In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks
 572 Track (Round 2)*, 2021. URL <https://openreview.net/forum?id=7Bywt2mQsCe>.

573 Chengsong Huang, Langlin Huang, and Jiaxin Huang. Divide, reweight, and conquer: A logit
 574 arithmetic approach for in-context learning, 2024. URL <https://arxiv.org/abs/2410.10074>.

575 Chengsong Huang, Langlin Huang, Jixuan Leng, Jiacheng Liu, and Jiaxin Huang. Efficient test-time
 576 scaling via self-calibration, 2025. URL <https://arxiv.org/abs/2503.00031>.

577 Jing Yu Koh, Stephen McAleer, Daniel Fried, and Ruslan Salakhutdinov. Tree search for language
 578 model agents, 2024. URL <https://arxiv.org/abs/2407.01476>.

579 Guillaume Lample, Marie-Anne Lachaux, Thibaut Lavril, Xavier Martinet, Amaury Hayat, Gabriel
 580 Ebner, Aurélien Rodriguez, and Timothée Lacroix. Hypertree proof search for neural theorem
 581 proving, 2022. URL <https://arxiv.org/abs/2205.11491>.

582 Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. Making
 583 language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual
 584 Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5315–
 585 5333, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/
 586 2023.acl-long.291. URL <https://aclanthology.org/2023.acl-long.291/>.

587 Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan
 588 Li. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. In *The Twelfth
 589 International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=ndR8Ytrzhh>.

594 Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan
 595 Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth*
 596 *International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=v8L0pN6EOi>.

597

598 Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction. In
 599 *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=jN5y-zb5Q7m>.

600

602 Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke
 603 Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time
 604 scaling, 2025. URL <https://arxiv.org/abs/2501.19393>.

605

606 David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien
 607 Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a
 608 benchmark. In *First Conference on Language Modeling*, 2024. URL <https://openreview.net/forum?id=Ti67584b98>.

609

610 Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute
 611 optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth Interna-*
 612 *tional Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=4FWAwZtd2n>.

613

614

615 Harshil Vejendla, Haizhou Shi, Yibin Wang, Tunyu Zhang, Huan Zhang, and Hao Wang. Efficient
 616 uncertainty estimation via distillation of bayesian large language models, 2025. URL <https://arxiv.org/abs/2505.11731>.

617

618 Xinglin Wang, Yiwei Li, Shaoxiong Feng, Peiwen Yuan, Yueqi Zhang, Jiayi Shi, Chuyi Tan, Boyuan
 619 Pan, Yao Hu, and Kan Li. Every rollout counts: Optimal resource allocation for efficient test-time
 620 scaling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.
 621 URL <https://openreview.net/forum?id=xSHqNf5Pdc>.

622

623 Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha
 624 Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language
 625 models. In *The Eleventh International Conference on Learning Representations*, 2023. URL
 626 <https://openreview.net/forum?id=1PL1NIMMrw>.

627

628 Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming
 629 Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang,
 630 Rongqi Fan, Xiang Yue, and Wenhui Chen. MMLU-pro: A more robust and challenging multi-
 631 task language understanding benchmark. In *The Thirty-eight Conference on Neural Information
 632 Processing Systems Datasets and Benchmarks Track*, 2024. URL <https://openreview.net/forum?id=y10DM6R2r3>.

633

634 Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws:
 635 An empirical analysis of compute-optimal inference for LLM problem-solving. In *The Thirteenth*
 636 *International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=VNckp7JEHn>.

637

638 Quan Xiao, Debarun Bhattacharjya, Balaji Ganesan, Radu Marinescu, Katsiaryna Mirylenka,
 639 Nhan H. Pham, Michael Glass, and Junkyu Lee. The consistency hypothesis in uncertainty quan-
 640 tification for large language models. In *Proceedings of the Forty-First Conference on Uncertainty
 641 in Artificial Intelligence*, UAI '25. JMLR.org, 2025.

642

643 Duygu Nur Yaldiz, Yavuz Faruk Bakman, Baturalp Buyukates, Chenyang Tao, Anil Ramakrishna,
 644 Dimitrios Dimitriadis, Jieyu Zhao, and Salman Avestimehr. Do not design, learn: A trainable
 645 scoring function for uncertainty estimation in generative LLMs. In *Findings of the Association for
 646 Computational Linguistics: NAACL 2025*, pp. 691–713, Albuquerque, New Mexico, April 2025.
 647 Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.
 findings-naacl.41. URL <https://aclanthology.org/2025.findings-naacl.41/>.

648 An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li,
 649 Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin
 650 Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Kebin Bao, Kexin Yang,
 651 Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang,
 652 Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan,
 653 Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL
 654 <https://arxiv.org/abs/2412.15115>.

655 Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu,
 656 Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical
 657 reasoning, 2025. URL <https://arxiv.org/abs/2501.07301>.

660 A FULL PROOF OF THEOREM 1

662 *Proof.* Without loss of generality, relabel so that the true index is $I = 1$. Define the unnormalized
 663 and normalized posteriors

$$664 \quad A_j := \prod_{t=1}^m \mathbb{P}(R_t | C_t, I = j), \quad X_j := \frac{A_j}{\sum_{k=1}^K A_k}, \quad j \in [K].$$

667 To prove the claim, it is enough to show that for every $k \neq 1$, $A_1/A_k \rightarrow \infty$ almost surely, which
 668 then implies $X_1 \rightarrow 1$ and $X_k \rightarrow 0$ almost surely.

670 **Step 1:** First, for a fixed $k \neq 1$, we define

$$672 \quad Y_k^{(t)} := \log \mathbb{P}(R_t | C_t, I = 1) - \log \mathbb{P}(R_t | C_t, I = k).$$

673 Under Assumption 3, conditioning on C_t we have

$$675 \quad \log \mathbb{P}(R_t | C_t, I = 1) = \begin{cases} \log C_t & \text{if } R_t = a_1, \\ \log(\frac{1-C_t}{K-1}) & \text{if } R_t \neq a_1, \end{cases}$$

678 and

$$680 \quad \log \mathbb{P}(R_t | C_t, I = k) = \begin{cases} \log C_t & \text{if } R_t = a_k, \\ \log(\frac{1-C_t}{K-1}) & \text{if } R_t \neq a_k. \end{cases}$$

682 Let $\theta_t := \frac{1-C_t}{K-1}$. Using Assumption 1 (i.i.d. across t given P and hence given C_t in this idealized
 683 parameterization), the conditional probabilities under the true model $I = 1$ are $\mathbb{P}(R_t = 1 | C_t, I = 1) = C_t$ and $\mathbb{P}(R_t = k | C_t, I = 1) = \theta_t$.

686 **Step 2:** In the next step, by taking the conditional expectation of $Y_k^{(t)}$ given C_t , it follows that

$$688 \quad \mathbb{E}[Y_k^{(t)} | C_t] = (C_t - \theta_t) \left(\log C_t - \log \theta_t \right) = (C_t - \theta_t) \log \left(\frac{C_t}{\theta_t} \right).$$

690 Because $x \mapsto \log x$ is strictly increasing, $(a - b) \log(a/b) > 0$ for $a \neq b$. Here $a = C_t$ and $b = \theta_t$,
 691 so the conditional mean is strictly positive whenever $C_t \neq \theta_t$, i.e., whenever $C_t \neq 1/K$. By the
 692 informativeness condition $\mathbb{P}(C_t = 1/K) = 0$,

$$693 \quad \mu_k := \mathbb{E}[Y_k^{(t)}] = \mathbb{E}[\mathbb{E}[Y_k^{(t)} | C_t]] > 0.$$

695 Moreover, $|Y_k^{(t)}|$ has finite expectation since $C_t \in (0, 1)$ a.s., making the logs finite.

698 **Step 3:** By Assumption 1, $\{Y_k^{(t)}\}_{t=1}^m$ are i.i.d. with $\mathbb{E}[|Y_k^{(t)}|] < \infty$ and $\mathbb{E}[Y_k^{(t)}] = \mu_k > 0$. Finally,
 699 the Strong Law of Large Numbers yields

$$700 \quad \frac{1}{m} \sum_{t=1}^m Y_k^{(t)} \xrightarrow{\text{a.s.}} \mu_k > 0 \implies \sum_{t=1}^m Y_k^{(t)} \xrightarrow{\text{a.s.}} +\infty.$$

702 Exponentiating both sides of $\log(A_1/A_k) = \sum_{t=1}^m Y_k^{(t)}$ gives
 703

$$704 \frac{A_1}{A_k} = \exp\left(\sum_{t=1}^m Y_k^{(t)}\right) \xrightarrow{\text{a.s.}} \infty. \\ 705 \\ 706$$

707 Since this holds for every $k \neq 1$, we have $A_k/A_1 \rightarrow 0$ almost surely for all $k \neq 1$, and therefore
 708

$$709 X_1 = \frac{1}{1 + \sum_{k \neq 1} A_k/A_1} \xrightarrow{\text{a.s.}} 1, \quad X_k = \frac{A_k/A_1}{1 + \sum_{j \neq 1} A_j/A_1} \xrightarrow{\text{a.s.}} 0. \\ 710$$

711 This completes the proof. \square
 712

713 B FULL PROOF OF THEOREM 2 714

715 *Proof.* Without loss of generality, let $I = 1$. For $j \in [K]$, define
 716

$$717 A_j := \prod_{t=1}^m \mathbb{P}(R_t \mid C_t, I = j), \quad X_j := \frac{A_j}{\sum_{k=1}^K A_k}. \\ 718 \\ 719$$

720 It suffices to show that for every $k \neq 1$, $A_1/A_k \xrightarrow{\text{a.s.}} \infty$, which implies $X_1 \rightarrow 1$ and $X_k \rightarrow 0$ almost
 721 surely.
 722

723 **Step 1:** In the first step, we fix $k \neq 1$ and set
 724

$$725 Y_k^{(t)} := \log \mathbb{P}(R_t \mid C_t, I = 1) - \log \mathbb{P}(R_t \mid C_t, I = k). \\ 726$$

726 With the model likelihood above and $\theta_t = \frac{1-C_t}{K-1}$,
 727

$$728 \log \mathbb{P}(R_t \mid C_t, I = 1) = \begin{cases} \log C_t & (R_t = a_1), \\ \log \theta_t & (R_t \neq a_1), \end{cases} \quad \log \mathbb{P}(R_t \mid C_t, I = k) = \begin{cases} \log C_t & (R_t = a_k), \\ \log \theta_t & (R_t \neq a_k). \end{cases} \\ 729 \\ 730$$

731 **Step 2:** Next by conditioning on (\mathbf{P}, C_t) and using $\mathbb{P}(R_t = a_r \mid \mathbf{P}) = P_r$, we have
 732

$$733 \mathbb{E}[Y_k^{(t)} \mid \mathbf{P}, C_t] = P_1(\log C_t - \log \theta_t) + P_k(\log \theta_t - \log C_t) = (P_1 - P_k)(\log C_t - \log \theta_t). \\ 734$$

735 Taking expectations over (\mathbf{P}, C_t) gives
 736

$$737 \mu_k := \mathbb{E}[Y_k^{(t)}] = \mathbb{E}_{\mathbf{P}, C} \left[(P_1 - P_k) \log \left(\frac{C_t}{\theta_t} \right) \right]. \\ 738$$

739 By assumption, $\mu_k > 0$ for all $k \neq 1$ and $\mathbb{E}[|Y_k^{(t)}|] < \infty$ (since $C_t \in (0, 1)$ a.s. and the logs are
 740 integrable).
 741

742 **Step 3:** By Assumption 1 and the i.i.d. generation of (R_t, C_t) given \mathbf{P} , the increments $\{Y_k^{(t)}\}_{t=1}^m$
 743 are i.i.d. with finite first moment and mean $\mu_k > 0$. The Strong Law of Large Numbers yields
 744

$$745 \frac{1}{m} \sum_{t=1}^m Y_k^{(t)} \rightarrow \mu_k > 0 \text{ almost surely} \implies \sum_{t=1}^m Y_k^{(t)} \rightarrow +\infty \text{ almost surely.} \\ 746$$

747 Hence
 748

$$749 \frac{A_1}{A_k} = \exp\left(\sum_{t=1}^m Y_k^{(t)}\right) \rightarrow \infty \text{ almost surely,} \\ 750$$

751 so $A_k/A_1 \rightarrow 0$ a.s. for all $k \neq 1$, and therefore $X_1 \rightarrow 1$ and $X_k \rightarrow 0$ almost surely.
 752

753 **Converse (necessity).** If $\mu_{k^*} < 0$ for some $k^* \neq 1$, then by the same SLLN argument,
 754 $\sum_{t=1}^m Y_{k^*}^{(t)} \rightarrow -\infty$ a.s., so $A_1/A_{k^*} \rightarrow 0$ almost surely and thus $X_1 \rightarrow 0$ almost surely. (When
 755 $\mu_k = 0$ for some k , the LLR has zero drift and the posterior need not concentrate; this is a boundary
 case.) \square

756 C ADDITIONAL RESULTS
757758 Additional results are provided in Table 3 and Table 4.
759

760

761

	AIME24		MATH500		GSM8K		GPQA		MMLU_Pro		Avg.	
	#Calls	Acc										
SC	4.00	67.78	4.00	78.80	4.00	93.40	4.00	50.67	4.00	58.79	4.00	69.89
CGES (Ours)												
LNS[Arithmetic mean]	3.03 (-0.97)	68.89 (+1.11)	2.59 (-1.41)	79.40 (+0.60)	1.84 (-2.16)	93.48 (+0.08)	1.52 (-2.48)	53.02 (+2.35)	2.76 (-1.24)	59.05 (+0.26)	2.35 (-1.65)	70.77 (+0.88)
LNS[Geometric mean]	4.00 (-0.83)	68.89 (+0.21)	3.75 (-1.07)	79.40 (+0.60)	3.86 (-1.86)	93.48 (+0.00)	4.00 (-0.68)	53.02 (+2.87)	3.98 (-0.98)	59.05 (+0.21)	3.92 (-1.08)	70.77 (+0.78)
MARS	2.48 (-1.52)	67.78 (+0.00)	2.93 (-1.00)	79.40 (+0.20)	2.14 (-1.85)	93.40 (+0.05)	3.33 (-1.00)	53.54 (+2.38)	3.02 (-0.95)	59.00 (+0.27)	2.92 (-1.26)	70.67 (+0.58)
RM Confidence	1.72 (-2.28)	68.89 (+1.11)	1.80 (-2.20)	78.93 (+0.13)	2.16 (-1.84)	94.16 (+0.76)	3.88 (-0.12)	50.51 (-0.16)	2.41 (-1.59)	58.98 (+0.19)	2.39 (-1.61)	70.29 (+0.40)
	3.68 (-0.32)	73.33 (+5.55)	2.74 (-1.26)	83.27 (+4.47)	2.43 (-1.57)	94.94 (+1.54)	3.97 (-0.03)	50.34 (-0.33)	3.44 (-0.56)	61.97 (+3.18)	3.25 (-0.75)	72.77 (+2.88)

780 Table 3: Accuracy (%) and avg. #Calls across five reasoning tasks; Avg is mean over benchmarks.
781 Parentheses show difference vs. SC (#Calls=4 or SC Acc). Efficient (first row) vs. Conservative
782 (second row) are two CGES settings.
783

784

785

786

	AIME24		MATH500		GSM8K		GPQA		MMLU_Pro		Avg.	
	#Calls	Acc										
SC	8.00	74.45	8.00	81.47	8.00	93.91	8.00	50.17	8.00	60.68	8.00	72.14
CGES (Ours)												
LNS[Arithmetic mean]	4.41 (-3.59)	74.45 (0.00)	4.08 (-3.92)	81.87 (+0.40)	3.41 (-4.59)	93.83 (-0.08)	2.08 (-5.92)	51.12 (+0.95)	4.42 (-3.58)	60.68 (0.00)	3.68 (-4.32)	72.39 (+0.25)
LNS[Geometric mean]	6.88 (-1.12)	74.45 (0.00)	4.88 (-3.12)	81.87 (+0.40)	4.26 (-3.74)	93.83 (-0.08)	7.56 (-0.44)	51.12 (+0.95)	6.59 (-1.41)	60.68 (0.00)	6.03 (-1.97)	72.39 (+0.25)
MARS	4.32 (-3.68)	75.19 (+0.74)	4.67 (-3.33)	81.73 (+0.26)	4.04 (-3.96)	93.78 (-0.13)	4.25 (-3.75)	51.68 (+1.51)	5.78 (-2.22)	60.89 (+0.21)	4.61 (-3.39)	72.65 (+0.51)
RM Confidence	7.71 (-0.29)	72.22 (-2.23)	5.52 (-2.48)	81.80 (+0.33)	5.05 (-2.95)	93.78 (-0.13)	7.99 (-0.01)	51.35 (+1.18)	7.40 (-0.60)	60.95 (+0.27)	6.74 (-1.26)	72.02 (-0.12)
	4.04 (-3.96)	74.45 (0.00)	3.69 (-4.30)	81.60 (+0.13)	5.08 (-2.92)	93.88 (-0.03)	3.56 (-4.44)	52.69 (+2.52)	4.43 (-3.57)	60.75 (+0.07)	4.16 (-3.84)	72.67 (+0.53)
	7.63 (-0.37)	75.56 (+1.11)	5.63 (-2.37)	81.87 (+0.40)	5.08 (-2.92)	93.88 (-0.03)	7.97 (-0.03)	51.68 (+1.51)	7.49 (-0.51)	61.03 (+0.35)	6.76 (-1.24)	72.80 (+0.66)
CGES - Near Ideal Scenario (Ours)												
RM Confidence	1.72 (-6.28)	75.56 (+1.11)	3.26 (-4.74)	82.67 (+1.20)	2.37 (-5.63)	94.44 (+0.53)	5.32 (-2.69)	50.34 (+0.17)	4.42 (-3.58)	61.68 (+1.00)	3.42 (-4.58)	72.94 (+0.80)

805 Table 4: Accuracy (%) and avg. #Calls across five reasoning tasks; Avg is mean over benchmarks.
806 Parentheses show difference vs. SC (#Calls=4 or SC Acc). Efficient (first row) vs. Conservative
807 (second row) are two CGES settings.
808

809

Table 6: Comparison of CGES and adaptive self-consistency baselines on 70B–72B models.

	AIME24		MATH500		MMLU_Pro		Avg.	
	#Calls	Acc	#Calls	Acc	#Calls	Acc	#Calls	Acc
SC	16.00	83.33	16.00	87.47	16.00	74.53	16.00	81.78
ESC (w=4)	8.98 (-4.98)	83.33 (+0.00)	6.30 (-8.12)	87.60 (-0.13)	6.67 (-10.78)	74.54 (-0.02)	7.32 (-4.10)	81.83 (+0.17)
ESC (w=8)	12.62 (-1.60)	83.33 (+0.00)	10.34 (-4.43)	87.47 (+0.00)	10.88 (-6.50)	74.52 (+0.00)	11.28 (-1.13)	81.77 (+0.00)
Adaptive-Consistency (BETA)	8.11 (-5.75)	83.33 (+0.00)	6.05 (-8.64)	87.60 (-0.13)	6.41 (-10.95)	74.54 (-0.02)	6.86 (-4.57)	81.83 (+0.50)
CGES (Ours)								
LNS[Arithmetic mean]	6.86 (-9.53)	83.33 (+0.00)	4.47 (-11.31)	87.20 (-0.27)	4.37 (-11.50)	74.54 (-0.13)	5.24 (-13.91)	81.69 (+0.12)
	<u>8.13</u> (-6.41)	84.44 (+0.00)	4.47 (-10.19)	87.20 (-0.33)	<u>5.54</u> (-11.50)	74.68 (-0.13)	<u>6.05</u> (-5.52)	82.11 (+0.51)

C.1 CONFIDENCE CALIBRATION ANALYSIS

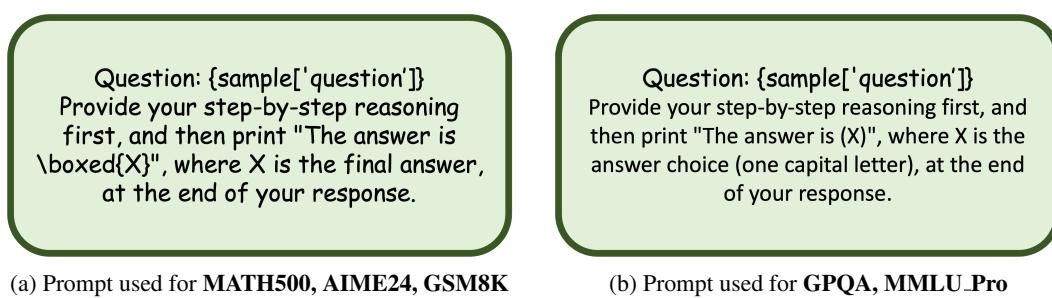
To evaluate how sensitive CGES is to miscalibrated confidence estimates, we measured two standard calibration metrics, Expected Calibration Error (ECE) and Maximum Calibration Error (MCE), for all confidence estimators (LNS variants, MARS, and reward-model scores) across all benchmarks. As summarized in Table 5, reward-model confidence is substantially better calibrated on datasets included in its training (e.g., MATH500 and GSM8K), and this corresponds to stronger CGES performance. Conversely, even with noisier estimators such as LNS (Arithmetic), CGES maintains near-identical accuracy and large call reductions, indicating robustness to *imperfect confidence*. *Overall, better calibration primarily improves efficiency (earlier stopping), whereas the final aggregated posterior remains stable, making CGES effective even under significant confidence noise.*

	AIME24		MATH500		GSM8K		GPQA		MMLU_Pro	
	ECE	MCE	ECE	MCE	ECE	MCE	ECE	MCE	ECE	MCE
LNS[Arithmetic mean]	0.295	0.682	0.196	0.555	0.022	0.572	0.336	0.437	0.291	0.377
LNS[Geometric mean]	0.195	0.610	0.153	0.375	0.036	0.451	0.205	0.359	0.189	0.302
MARS	0.176	0.526	0.139	0.446	0.038	0.418	0.202	0.394	0.189	0.206
RM Confidence	0.288	0.749	0.042	0.245	0.022	0.352	0.351	0.480	0.151	0.232

Table 5: Calibration scores (ECE and MCE) for different confidence estimators. Lower values indicate better calibration.

C.2 RESULTS WITH 70B–72B MODELS

We additionally evaluated CGES and adaptive SC baselines using large *DeepSeek-R1-Distill-Llama-70B* on AIME24 and *Qwen2.5-72B-Instruct* on MATH500 and MMLU-Pro. As shown in Table 6, the qualitative trends match those observed with 7B models: CGES consistently requires far fewer calls than Self-Consistency and ASC/ESC while achieving comparable or slightly improved accuracy. The Efficient configuration often reduces calls by 60–70% relative to SC, even at 70B scale.

864 **D DECODING HYPERPARAMETERS AND PROMPT TEMPLATES**
865866 For CGES, we sweep the stopping threshold γ over the grid
867 0.70, 0.75, 0.80, 0.85, 0.90, 0.95, 0.99, 0.999, 0.9999 to explore different accuracy-efficiency
868 trade-offs. All experiments, including those for SC, ESC, CGES, and confidence estimation, share
869 the same decoding setup. Specifically, we allow a maximum of 32,768 generation tokens, use
870 a temperature of 0.7, and apply nucleus sampling with top- $p = 1.0$ (i.e., no truncation). Top- k
871 sampling is disabled in all runs.872
873 **Prompt Templates.** We use dataset-specific answer-format constraints to simplify parsing.
874 Prompt templates are shown in Fig 4.885 Figure 4: Two prompt templates for evaluation.
886
887888 **E THE USE OF LARGE LANGUAGE MODELS (LLMs)**
889890 We used large language models (LLMs) solely to aid with polishing the writing and improving
891 clarity of exposition. No part of the research ideation, methodology, analysis, or experimental results
892 was generated by LLMs. The authors take full responsibility for the content of this paper.
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917