CGES: Confidence-Guided Early Stopping for Efficient and Accurate Self-Consistency

Anonymous ACL submission

Abstract

Large language models (LLMs) are often queried multiple times at test time, with predictions aggregated by majority vote. While effective, this *self-consistency* (Wang et al., 2023) strategy requires a fixed number of calls and fails when the correct answer is infrequent. We introduce ***Confidence-Guided Early Stopping (CGES)***, a Bayesian framework that forms posteriors over candidate answers and adaptively halts sampling once one answer accumulates enough posterior mass. We prove guarantees in both an ideal calibrated regime and a realistic noisy-confidence regime under a directional drift condition. Averaged over five reasoning benchmarks, CGES reduces the average number of calls by **58%** on average (from 16.0 to 6.7) while matching its accuracy within **0.4 percentage points** of self-consistency.

1 Introduction

Large language models (LLMs) have achieved strong progress across reasoning, problem solving, and open-domain tasks (Singh et al., 2026; Comanici et al., 2025; DeepSeek-AI et al., 2025; Yang et al., 2025). A common way to improve reliability is *test-time scaling* (Snell et al., 2025; Wu et al., 2025; Muennighoff et al., 2025), a family of methods that allocate additional inference-time computation to improve performance. One subset of these methods samples multiple responses and aggregates them into a final prediction (Wang et al., 2023; Li et al., 2024; Aggarwal et al., 2023). Among the most widely used methods, self-consistency (SC) (Wang et al., 2023) aggregates outputs by majority vote, leveraging the intuition that the most frequent answer across diverse generations is likely to be correct. While simple and effective in many settings, majority-based aggregation suffers from two major shortcomings. First, it assumes that response frequency is a faithful proxy for correctness, which fails in cases where the correct answer appears infrequently (Huang et al., 2024b; Taubenfeld et al., 2025). Second, it requires a fixed number of model calls regardless of confidence, leading to substantial inefficiency (Li et al., 2024; Aggarwal et al., 2023; Wang et al., 2025a).

Confidence signals offer an alternative perspective. Instead of depending solely on frequency, one can incorporate confidence scores that capture the model's belief in each response (Kadavath et al., 2022; Geng et al., 2024). These scores may be derived from different sources. *Token probabilities* are taken directly from the model's output distribution and reflect how certain the model is about generating each token (Malinin and Gales, 2021; Bakman et al., 2024). *Calibration schemes* adjust these raw probabilities so that they better match the actual likelihood of correctness, turning overconfident or underconfident estimates into more reliable signals (Kadavath et al., 2022; Yaldiz et al., 2025). *External reward models* are trained separately, often with human feedback or domain-specific supervision, and provide an independent measure of response quality beyond the model's own probabilities (Lightman et al., 2024; Zhang et al., 2025). Such signals can distinguish between frequent but uncertain answers and rare yet confident ones. This makes confidence a direct remedy for the central failure mode of self-consistency: majority frequency can dominate even when the minority answer carries stronger evidence.

We propose a *confidence-based Bayesian framework* for self-consistency. Rather than counting answer frequencies, our approach treats each sampled response and its confidence as probabilistic evidence, yielding posterior-style scores over candidate answers. This gives CGES two advantages over majority voting: (1) when the correct answer is frequent, CGES often reaches the same decision with fewer samples by stopping once posterior mass concentrates; and (2) when the correct answer is a minority, CGES can still recover it if correct

samples have stronger confidence than frequent but uncertain alternatives. Figure 1 illustrates both cases with real examples.

We formalize this idea under both an *ideal* calibrated-confidence regime and a *realistic* noisy-confidence regime that reflects practical LLM inference, where confidence estimates may be imperfect and only directionally informative. The realistic analysis shows that CGES remains consistent as long as confidence signals are even weakly informative about the correct answer, and our experiments validate this behavior with practical confidence estimators.

To operationalize this framework, we introduce *Confidence-Guided Early Stopping (CGES)*, which combines Bayesian scoring with adaptive stopping. CGES trades off accuracy and efficiency by stopping once posterior concentration exceeds a threshold or the 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, showing the value of calibrated confidence for test-time scaling and principled Bayesian aggregation.

Unlike prior self-consistency methods that separate **aggregation** (e.g., majority vote) from **stopping** (e.g., agreement-based heuristics), our approach casts both within a single probabilistic framework. We treat sampled responses together with their confidence scores as evidence for posterior inference over candidate answers, and use posterior concentration to drive adaptive stopping. This yields a unified view of test-time scaling in which confidence is not merely an auxiliary signal, but a mechanism for both more principled aggregation and more efficient inference.

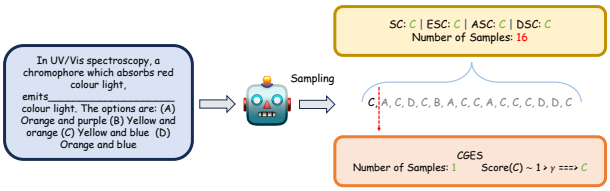
Our contributions are as follows:

- We reformulate self-consistency as *posterior inference over candidate answers*, replacing frequency-based majority voting with confidence-aware Bayesian aggregation.
- We introduce *Confidence-Guided Early Stopping (CGES)*, a unified framework in which the same posterior scores are used both to aggregate sampled responses and to decide when further sampling is unnecessary.
- We establish guarantees under an *ideal* calibrated-confidence setting in Theorem 3.2, and, more importantly, under a *realistic* noisy-confidence regime in Theorem 3.5, where ideal assumptions are relaxed. **We show that CGES remains effective beyond idealized assumptions.**
- We empirically validate CGES across five reasoning benchmarks, showing that confidence-aware aggregation can substantially reduce inference cost while maintaining or improving accuracy relative to standard self-consistency and heuristic early-stopping baselines.

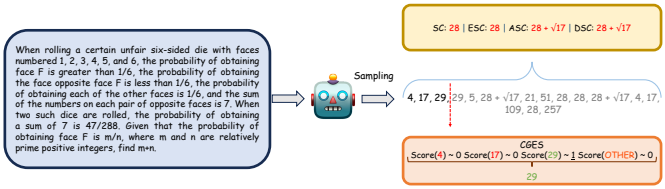
2 Related Work

A widely used test-time scaling method is *self-consistency* (Wang et al., 2023), which aggregates multiple reasoning paths by majority vote. While effective, it uses a fixed number of calls and can fail when the correct answer is infrequent (Huang et al., 2024b; Taubenfeld et al., 2025). Several variants improve efficiency via dynamic stopping rules (Li et al., 2024; Aggarwal et al., 2023; Wang et al., 2025a; Huang et al., 2025). Recent methods also use confidence-weighted aggregation (Taubenfeld et al., 2025) or exploit minority responses (Huang et al., 2024b). Unlike prior work, we introduce a **Bayesian framework** with **theoretical guarantees** for **confidence-guided early stopping**, unifying aggregation and adaptive stopping in one probabilistic model. Concurrent work also studies confidence in test-time scaling. DeepConf (Fu et al., 2026) uses token-level confidence within a single trajectory to halt low-confidence generations. AR-ROL (Xu et al., 2026) learns a quality head during RLVR training and reuses it as voting weights at inference. Both are complementary to CGES, a training-free Bayesian rule for aggregation and stopping over completed responses, which can directly use such learned confidence signals.

Beyond self-consistency, a broader literature studies adaptive allocation of test-time compute. One line examines compute-optimal scaling and inference scaling laws, proposing budget-aware sampling strategies (Snell et al., 2025; Wu et al., 2025; Wang et al., 2025b; Muennighoff et al., 2025). Another combines search with verification, using tree-search expansions, step-wise verifiers, iterative elimination, and query-variant ensembling to prune or refine candidates (Bi et al., 2025; Lample et al., 2022; Koh et al., 2024; Li et al., 2023; Lightman et al., 2024; Chen et al., 2026; Huang et al.,



(a) Efficiency: all baselines reach the correct answer (C) but use the full budget of 16 samples, while CGES stops after one sample once Score(C) $\sim 1 > \gamma$.



(b) Accuracy: SC/ESC return the plurality answer 28 and ASC/DSC return $28+\sqrt{17}$, both wrong; CGES aggregates confidences and recovers the correct minority answer 29.

Figure 1: **Examples of CGES vs. baselines.** Top: confidence-driven early stopping saves samples when the model is already certain. Bottom: confidence-weighted aggregation recovers a minority-but-confident answer that majority voting misses.

2024a). These methods typically rely on structured search, verifier signals, or semantic clustering to select a single best reasoning path. Our approach is complementary: it treats confidence as probabilistic evidence and *aggregates* sampled responses through a lightweight Bayesian update with formal consistency guarantees.

A core component of our framework is *confidence estimation*, studied extensively for generative LLMs. Unlike uncertainty quantification, which characterizes the predictive distribution, confidence estimation is response-specific, reflecting the model's belief in a generated output's correctness. (Lin et al., 2024; Kadavath et al., 2022). Existing methods derive confidence from token probabilities with length normalization or learned scoring (Malinin and Gales, 2021; Bakman et al., 2024; Yaldiz et al., 2025), information-theoretic or similarity-based black-box signals (Abbasi-Yadkori et al., 2024; Bhattacharjya et al., 2025), and Bayesian or distillation-based estimators (Vejendla et al., 2025), with surveys and taxonomies providing broader context (Geng et al., 2024; Xiao et al., 2025). Although CGES requires a confidence signal, our goal is not to introduce a new estimator. Instead, we adapt existing techniques within CGES, modifying them when needed; when response-level scores are unavailable, uncertainty quantification methods can serve as confidence proxies.

3 Confidence-Based Approach

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 $\mathcal{U} = \{a_1, a_2, \ldots, a_K\}$ denote the complete set of all possible distinct candidate final answers produced by the LLM, augmented with an explicit *OTHER/unseen* category. The parameter K is the maximum number of such candidates.[1]

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 $\{a_1, \ldots, a_K\}$ can recover A. 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, the LLM induces an unknown but fixed response distribution $\mathbf{P} = (P_1, \ldots, P_K)$ on $\mathcal{U}$, with $P_j = \mathbb{P}[R = a_j \mid Q, A, I]$. Intuitively, this captures the stochastic behavior of the LLM on the query: the likelihood of producing each candidate answer depends both on the query and on which candidate is correct. For each LLM call $t = 1, 2, \ldots$, we draw a response–confidence pair $(R_t, C_t) \sim f_{R,C|\mathbf{P}}(\cdot, \cdot \mid \mathbf{P})$ from a joint conditional distribution, with marginal $R_t \sim \mathbf{P}$. In general C_t may depend on R_t, since it is computed from the trajectory that produced it. The ideal Bayesian analysis in Theorem 3.2 uses only the conditional likelihood $\mathbb{P}(R_t \mid C_t, I)$ specified in Assumption 3, and therefore does not assume independence between R_t and C_t.Here R_t denotes the

[1]Since the answer space includes at least one concrete candidate and the *OTHER/unseen* category, we necessarily have $K \geq 2$.

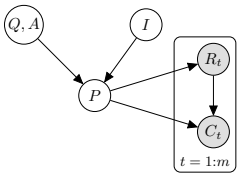


Figure 2: Graphical model for the LLM sampling process. The edge $R_t \to C_t$ reflects that the confidence score is computed from the trajectory that produced the response, e.g., token probabilities or a reward-model score on R_t.

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 (noisy) proxy for its probability of correctness $\mathbf{P}$. This structure can be represented as a graphical model, illustrated in Figure 2.

Idealistic vs. Realistic Assumptions. We distinguish two roles for our assumptions. *Idealistic* assumptions 3–4 are simplifying conditions used to *derive* the clean Bayesian scoring rule of Section 3.3 and the optimality result of Theorem 3.2; under 3, each C_t is treated as the *exact* conditional correctness probability of R_t, encoded in Eq. (3). *Realistic* assumptions 1–2 are mild and expected to hold in practice. The consistency guarantee under realistic data (Theorem 3.5) is proved using *only* 1 and 2; it does not require 3 or 4. In this regime C_t is a noisy estimator of correctness (e.g., a token-probability score or reward-model output) and may depend on R_t. Section 4 shows that CGES remains effective on real LLM data, where the idealistic assumptions do not hold.

The four assumptions are as follows:

1. Given a fixed query Q and correct index I (and hence fixed $\mathcal{U}$ and $\mathbf{P}$), the samples $\{(R_t, C_t)\}_{t=1}^m$ are i.i.d. This is standard in self-consistency and test-time scaling analyses (Snell et al., 2025; Wu et al., 2025).
2. $I \sim \text{Uniform}(\{1, \dots, K\})$. This prior over the correct candidate's *index* treats all K hypotheses symmetrically: the labeling of $\mathcal{U}$ is arbitrary, so the correct candidate's position carries no information and does not affect the proposed method's performance.
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:

$$\begin{aligned} \mathbb{P}(R_t = a_i \mid C_t, I = i) &= C_t, \\ \mathbb{P}(R_t = a_j \neq a_i \mid C_t, I = i) &= \frac{1 - C_t}{K - 1}. \end{aligned} \tag{1}$$

4. The confidence score C_t is independent of the index of the correct answer: $\mathbb{P}(C_t \mid I = i) = \mathbb{P}(C_t \mid I = j)$ for all $i, j \in \{1, \dots, K\}$. This is a labeling-symmetry condition on the *marginal* of C_t and does *not* rule out larger C_t on correct answers.

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 infer the (unknown) index of the correct candidate, denoted by $I \in \{1, \dots, K\}$, from observed response–confidence pairs $\text{Obs} \triangleq \{(R_t, C_t)\}_{t=1}^m$.

Lemma 3.1 (Posterior simplification under confidence-independence)**.** *Assume: (i) the samples $\{(R_t, C_t)\}_{t=1}^m$ are i.i.d. (Assumption 1), (ii) the prior over indices is uniform, $\mathbb{P}(I = i) = 1/K$ (Assumption 2), and (iii) the confidence score is independent of the correct index, i.e., $\mathbb{P}(C_t \mid I = i) = \mathbb{P}(C_t \mid I = j)$ for all i, j (Assumption 4). Then the posterior over I satisfies*

$$\mathbb{P}(I{=}i \mid \text{Obs}) \;=\; \frac{\prod_t \mathbb{P}(R_t \mid C_t, I{=}i)}{\sum_{k=1}^K \prod_t \mathbb{P}(R_t \mid C_t, I{=}k)}. \tag{2}$$

In particular, the marginal law of $\{C_t\}_{t=1}^m$ does not affect the posterior beyond the conditioning in $\mathbb{P}(R_t \mid C_t, I = i)$.

Proof Sketch. Bayes' rule combined with the uniform prior (Assumption 2) reduces the posterior to a normalized product of per-sample joint likelihoods; the chain rule decomposes each joint, and Assumption 4 cancels the resulting $\mathbb{P}(C_t)$ factor common to numerator and denominator, yielding (2). A full proof is provided in Appendix A.

Model instantiations. Lemma 3.1 reduces posterior inference to specifying the conditional model $\mathbb{P}(R_t \mid C_t, I = i)$, which under ideal calibration (Assumption 3) has the explicit form:

$$\mathbb{P}(R_t \mid C_t, I = i) = \begin{cases} C_t & \text{if } R_t = a_i, \\ \frac{1 - C_t}{K - 1} & \text{otherwise,} \end{cases} \tag{3}$$

which is the model used in Theorem 3.2. In the *realistic* setting of Theorem 3.5, Assumptions 3 and 4 are dropped: C_t is a noisy estimator of correctness that may depend on R_t, and the formula in Eq. (2) is used as a *scoring rule* rather than as a true posterior. Theorem 3.5 establishes consistency of this scoring rule under only Assumptions 1 and 2.

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 query, our goal is to convert them into normalized, confidence-weighted scores over the candidate set $\mathcal{U} = \{a_1, a_2, \ldots, a_K\}$. Under Assumptions 1– 4, these scores coincide with Bayesian posterior probabilities; otherwise, they are pseudo-posteriors with consistency characterized in Theorem 3.5. The scoring rule follows from the factorized posterior in Eq. (2): for each candidate a_i, we test the hypothesis that a_i is the correct answer. Under this hypothesis, each observation (R_t, C_t) contributes C_t when $R_t = a_i$, and $(1 - C_t)/(K - 1)$ otherwise. Thus, the unnormalized score for a_i is $s_{a_i} = \prod_{t:R_t=a_i} C_t \prod_{t:R_t\neq a_i}(1 - C_t)/(K - 1)$, which is normalized as $\mathrm{score}(a_i) = s_{a_i}/\sum_{j=1}^K s_{a_j}$. Lines 1–5 of Algorithm 1 implement this SCORE routine.

Lines 6 through 17 wrap SCORE into an adaptive loop that allocates test-time compute per question. We begin with one sample per question, compute posteriors via SCORE, and maintain the set of unresolved questions D_{rem} whose current top posterior is below a confidence threshold γ. At each subsequent round, we query the LLM only for questions in D_{rem}, append the new (R_t^n, C_t^n), recompute SCORE on that question's accumulated samples, and remove the question from D_{rem} once its top posterior exceeds γ. The process halts when all questions are confident or the budget B is reached, returning the argmax label per question along with the average number of LLM calls. If no candidate ever exceeds γ before the budget is exhausted, CGES still returns the current argmax label, so γ controls test-time compute rather than enforcing abstention. Extending the framework with a no-answer option is left for future work.

The theoretical analysis in Sections 3.4 and 3.5 treats $\mathcal{U}$ and $K \geq 2$ as fixed. In the online implementation, we instantiate $\mathcal{U}_t$ as the answers observed through round t together with an explicit OTHER or unseen bucket, ensuring $K_t \geq 2$ from the first round onward. Consistency extends to this dynamic-support case (Appendix D).

Algorithm 1 Confidence-Guided Early Stopping (CGES)

Require: Candidate set $\mathcal{U} = \{a_1, \ldots, a_K\}$; N questions; threshold γ; budget B

1: **function** SCORE($\mathcal{S} = \{(R_t, C_t)\}_{t=1}^m$)
2: **for** $a_i \in \mathcal{U}$ **do** $s_{a_i} \leftarrow \prod_{t\,;\,R_t=a_i} C_t \cdot \prod_{t\,;\,R_t\neq a_i} \frac{1-C_t}{K-1}$
3: $Z \leftarrow \sum_{a_i} s_{a_i}$
4: **return** $\{s_{a_i}/Z\}_{a_i\in\mathcal{U}}$
5: **end function**
6: Initialize $\mathrm{scores}_n \leftarrow$ SCORE($\{(R_1^n, C_1^n)\}$) for all $n \in [N]$; set $D_{\text{rem}} \leftarrow [N]$, calls $\leftarrow N$
7: **for** $t = 2$ to B **do**
8: $D_{\text{rem}} \leftarrow \{n \in D_{\text{rem}}\,;\, \max_i \mathrm{scores}_n[i] < \gamma\}$
9: **if** $D_{\text{rem}} = \emptyset$ **then**
10: **break**
11: **end if**
12: Query LLM for each $n \in D_{\text{rem}}$; calls $+= |D_{\text{rem}}|$
13: **for** $n \in D_{\text{rem}}$ **do**
14: $\mathrm{scores}_n \leftarrow$ SCORE($\{(R_1^n, C_1^n), \ldots, (R_t^n, C_t^n)\}$)
15: **end for**
16: **end for**
17: **return** $\hat{y}_n = a_{\arg\max_i \mathrm{scores}_n[i]}$ for all n, and usage $=$ calls$/N$

3.4 Theoretical Analysis: Ideal Scenario

We now analyze CGES under an ideal calibrated-confidence model. The following theorem shows that the CGES posterior concentrates on the true answer as m grows.

Theorem 3.2. *Under Assumptions 1, 2, 3, and 4, and provided $C_t \in (0, 1)$ almost surely with $\mathbb{P}(C_t = 1/K) = 0$ and $\mathbb{E}\left[\left|\log\frac{C_t(K-1)}{1-C_t}\right|\right] < \infty$, the Bayesian confidence-based aggregator identifies the correct answer with probability tending to one as the number of samples m grows:*

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

where $X_i = \prod_{t=1}^m \mathbb{P}(R_t \mid C_t, I = i) \,/\, \sum_{k=1}^K \prod_{t=1}^m \mathbb{P}(R_t \mid C_t, I = k)$. In fact, $X_I \to 1$ and $X_k \to 0$ for all $k \neq I$ almost surely.

Proof Sketch. For any wrong index $k \neq I$, the log-likelihood ratio of the samples under hypotheses I vs. k has strictly positive expected increment whenever $C_t \neq 1/K$. The Strong Law of Large Numbers then drives the cumulative ratio to $+\infty$, forcing $X_I \to 1$ and $X_k \to 0$ a.s. Full proof in Appendix B.

Theorems 3.2 and 3.5 characterize the limiting behavior of X_i as the number of samples grows. The next proposition gives the CGES threshold γ a calibrated meaning at the actual (finite) stopping time used by Algorithm 1.

Proposition 3.3 (Posterior-risk control under early stopping)**.** *Let Assumptions 1–4 hold and fix $\gamma \in (0, 1)$. Define the threshold-stopping time $\tau_\gamma :=$*

$\inf\{m \geq 1 : \max_i X_i^{(m)} \geq \gamma\}$ *and the returned label* $\hat{I} := \arg\max_i X_i^{(\tau_\gamma)}$ *(with ties broken arbitrarily). On the event* $\{\tau_\gamma \leq B\}$, $\mathbb{P}(I \neq \hat{I} \mid \mathrm{Obs}_{\tau_\gamma}) \leq 1 - \gamma$.

Proof Sketch. Lemma 3.1 combined with Eq. (3) identifies the CGES score with the true Bayesian posterior at every sample size m. The stopping rule fires precisely when $\max_i X_i^{(\tau_\gamma)} \geq \gamma$, so the posterior on the returned candidate is at least γ, and its complement gives the stated risk bound. A complete proof is provided in Appendix E.

In fact, no decision rule that uses only the same response–confidence observations attains a smaller conditional probability of error than CGES at τ_γ, and τ_γ is the earliest sample count at which any such rule could have certified the same posterior-risk level (rigorous statement and proof in Appendix E, Corollary E.1).

Remark 3.4 (Tightness–frequency tradeoff at $\gamma \to 1$). The bound trades tightness against frequency of triggering: as $\gamma \to 1$, the threshold-stopping event $\{\tau_\gamma \leq B\}$ becomes rare, and has probability zero at $\gamma = 1$ since $C_t \in (0, 1)$ a.s. implies $\max_i X_i^{(m)} < 1$ for every finite m. The bound is informative for moderate γ, where $1 - \gamma$ is small and the algorithm halts with positive probability before exhausting the budget.

In the realistic regime of Theorem 3.5, X_i is a pseudo-posterior rather than the true posterior, so Proposition 3.3 becomes a pseudo-risk statement; empirically (Section 4), γ continues to provide a useful efficiency–accuracy knob.

3.5 Theoretical Analysis: Noisy Confidence Signals

We now drop the two idealistic Assumptions 3 and 4 and ask whether CGES remains consistent when confidence is only a noisy, possibly response-dependent signal.

Theorem 3.5 (Consistency under noisy confidence signals)**.** *Let* $\ell(c) := \log \frac{c(K-1)}{1-c}$. *Assume* $\{(R_t, C_t)\}_{t=1}^m$ *are i.i.d. samples from an arbitrary joint distribution over* $\mathcal{U} \times (0, 1)$, *with* $\mathbb{E}[|\ell(C_t)|] < \infty$. *Let* X_i *denote the normalized score computed by CGES (Algorithm 1) using the auxiliary likelihood of Eq. (3): under all four Assumptions 1–4,* X_i *coincides with the true Bayesian posterior on candidate* a_i *via Lemma 3.1; under only Assumptions 1–2,* X_i *is a normalized scoring rule that need not equal the true posterior. Without loss of generality, let* a_1 *denote the correct answer.*

For each incorrect candidate a_k *(*$k \neq 1$*), define the expected log-likelihood drift*

$$\mu_k := \mathbb{E}\big[\big(\mathbf{1}\{R_t = a_1\} - \mathbf{1}\{R_t = a_k\}\big)\, \ell(C_t)\big],$$

where the expectation is over the true joint distribution of (R_t, C_t).

If $\mu_k > 0$ *for every* $k \neq 1$, *then* $X_1 \xrightarrow{\text{a.s.}} 1$ *and* $X_k \xrightarrow{\text{a.s.}} 0$ *for all* $k \neq 1$, *as* $m \to \infty$. *Conversely, if* $\mu_{k^\star} < 0$ *for some* $k^\star \neq 1$, *then* $X_1 \xrightarrow{\text{a.s.}} 0$ *as* $m \to \infty$.

Interpretation. The drift μ_k is the gap in expected scoring reward $\ell(C_t)$ between emissions of the true answer a_1 and emissions of a_k, under the true joint of (R_t, C_t). The condition $\mu_k > 0$ requires neither calibrated confidences nor any independence between R_t and C_t; it allows even $\mathbb{P}(R_t = a_1) < \mathbb{P}(R_t = a_k)$ when correct emissions carry sufficiently higher confidence. This is the minority-but-confident regime in which majority voting fails (Figure 1b). The boundary $\mu_k = 0$ is not covered. **The auxiliary scoring rule derived from the ideal model retains its consistency under the realistic assumptions alone whenever** $\mu_k > 0$**,** a condition automatic under the ideal model. Only under the ideal model does γ carry a calibrated posterior-risk interpretation (Proposition 3.3); in the noisy regime it acts as an empirical accuracy–efficiency knob (Section 4).

Proof Sketch. For any $k \neq 1$, the log-likelihood ratio of the samples under hypotheses $I = 1$ vs. $I = k$ has one-step expected increment μ_k under the true joint. If $\mu_k > 0$, the Strong Law of Large Numbers forces the cumulative ratio to $+\infty$ and $X_1 \to 1$ a.s.; if $\mu_{k^\star} < 0$ for some $k^\star$, the cumulative ratio diverges to $-\infty$ and $X_1 \to 0$ a.s. Full proof in Appendix C.

Dynamic candidate set. Theorems 3.2 and 3.5 fix the candidate set $\mathcal{U}$ and its size K, whereas the online implementation grows them with the observed answers (Section 3.3). The consistency guarantee extends to this dynamic-support version: once the observed support stabilizes, the positive-drift condition of Theorem 3.5 drives the true answer's normalized score to one (rigorous statement and proof in Appendix D, Proposition D.1). The result is asymptotic and does not preclude premature stopping before the true answer first appears.

Remark 3.6 (Missing ground-truth among candidates). Theorems 3.2 and 3.5 analyze CGES under the event $A \in \mathcal{U}$, so I is well-defined. If

$A \notin \mathcal{U}$, any procedure restricted to $\mathcal{U}$ cannot return A: the error probability has an irreducible term $\mathbb{P}(A \notin \mathcal{U})$ that no self-consistency-style scheme can remove. In this misspecified regime, CGES (like other sampling-based methods) selects the candidate in $\mathcal{U}$ with the highest posterior mass, and the above consistency guarantees do not apply. In practice, a high threshold γ can flag such cases (e.g., when no candidate attains substantial posterior probability), but correctness guarantees are impossible without access to A.

4 Experiments

We evaluate CGES on five reasoning benchmarks with two open-source 7B models. The experiments answer three questions. **Does confidence-aware aggregation match self-consistency accuracy with fewer model calls? Do the gains survive wall-clock timing and matched-budget comparisons? Does the realistic-regime theory in Theorem 3.5 predict where CGES wins and where it does not?**

4.1 Setup

We use **AIME24** (30 problems), **MATH500** (Hendrycks et al., 2021), **GSM8K** (Cobbe et al., 2021), **MMLU_Pro** (Wang et al., 2024), and **GPQA Diamond** (Rein et al., 2024). Harder tasks (AIME24, GPQA) use *DeepSeek-R1-Distill-Qwen-7B* (DeepSeek-AI et al., 2025); the rest use *Qwen2.5-7B* (Yang et al., 2025). Baselines are SC (Wang et al., 2023), ESC (Li et al., 2024), ASC (Aggarwal et al., 2023), and DSC (Wang et al., 2025a). CGES uses five confidence signals (LNS (Malinin and Gales, 2021), DeepConf (Fu et al., 2026), MARS (Bakman et al., 2024)) plus one near-ideal process-reward reference (CGES-RM, `Qwen2.5-Math-PRM-7B` (Zhang et al., 2025)). All numbers are means over 10 seeds at $B{=}16$. For CGES rows in Table 1, we report the *efficient* threshold: the smallest γ within 0.2pp of SC accuracy. Pareto curves sweep $\gamma \in [0.7, 0.9999]$. Full baseline descriptions, confidence definitions, hyperparameters, smaller-budget runs ($B{=}4, 8$), and the γ grid appear in Appendix F–L.

4.2 CGES matches SC with about half the calls

Table 1 reports accuracy and call counts at $B{=}16$. The five probability-based CGES variants cut average calls from 16.0 to between 6.7 and 7.5 while staying within 0.2pp of SC accuracy on every benchmark except GPQA. The most efficient deployable variant, CGES-DeepConf (B10), uses **6.71** calls (a **58%** reduction) at -0.4pp accuracy. ESC and ASC give smaller savings and lose 2pp on GPQA. DSC posts the lowest call count overall but at -1.9pp on average, a gap that widens at smaller budgets (Appendix Tables 3 and 4). CGES-RM acts as a ceiling. When its training distribution matches the task it improves accuracy by 0.4pp on MATH500 (2.88 calls) and 0.7pp on GSM8K (1.25 calls), but accuracy drops outside that range, an effect we explain in Section 4.4.

Call counts translate to real time. SC is the only method with a fixed sample budget, so its B samples can be drawn in one batched call. All adaptive methods, including ESC, ASC, DSC, and CGES, are sequential because each sampling decision depends on prior outputs. Thus, the relevant cost is sequential rounds rather than raw calls. Table 2 reports wall-clock time per question at batch size 1, where CGES still reduces latency by 40–50% over SC, confirming that fewer calls translate into real time savings.

4.3 Pareto frontier

Figure 3 addresses matched-budget comparisons directly. The CGES curves Pareto-dominate SC and all stopping baselines across the full range on AIME24, MATH500, and MMLU_Pro. CGES-RM on MATH500 sits well above every other curve, illustrating the ceiling.

4.4 When the Confidence Signal Is Uninformative

Theorem 3.5 shows CGES works only when the confidence signal is informative, that is, when it carries a positive drift toward correct answers. GPQA is the case where this fails. The confidence scores there barely separate correct from incorrect responses, so the drift is small, the posterior never concentrates, and CGES neither saves calls nor fully matches SC accuracy (Table 1). The other four benchmarks carry informative signals and CGES behaves as the theory predicts. This is not a weakness specific to one estimator. On GPQA every confidence signal is weakly informative, including the reward model. We measure signal informativeness per task and method, and confirm that it tracks CGES efficiency, in Appendix J.

Table 1: Main results at budget $B = 16$. CGES rows use the efficient γ, the smallest γ matching SC$\pm$0.2 pp, or the largest γ if no γ reaches it. Mean$\pm$std across 10 seeds. Δ rows show difference vs. SC (green=improvement, red=regression). Last column: avg over tasks. **Bold = best**, underline = second-best per column.

Method	AIME24		GPQA		MATH500		GSM8K		MMLU_Pro		**Avg**	
	Acc (%)	#Calls	Acc (%)	#Calls	Acc (%)	#Calls	Acc (%)	#Calls	Acc (%)	#Calls	Acc (%)	#Calls
SC	$77.3_{\pm1.3}$	$16.00_{\pm0.00}$	$\mathbf{50.0}_{\pm0.6}$	$16.00_{\pm0.00}$	$81.2_{\pm0.8}$	$16.00_{\pm0.00}$	$94.2_{\pm0.2}$	$16.00_{\pm0.00}$	$\mathbf{62.1}_{\pm0.3}$	$16.00_{\pm0.00}$	**73.0**	16.00
ESC (w=4)	$77.3_{\pm1.3}$	$10.68_{\pm0.41}$	$48.0_{\pm1.0}$	$12.88_{\pm0.19}$	$81.1_{\pm0.7}$	$7.74_{\pm0.07}$	$94.2_{\pm0.2}$	$5.16_{\pm0.03}$	$62.0_{\pm0.4}$	$8.99_{\pm0.07}$	72.5	9.09
Δ	0.0	−5.3	−2.0	−3.1	−0.1	−8.3	0.0	−10.8	−0.1	−7.0	−0.4	−6.9
ESC (w=8)	$77.3_{\pm1.3}$	$13.73_{\pm0.27}$	$48.0_{\pm1.0}$	$15.20_{\pm0.09}$	$81.2_{\pm0.8}$	$11.42_{\pm0.09}$	$94.2_{\pm0.2}$	$9.41_{\pm0.04}$	$62.1_{\pm0.3}$	$12.87_{\pm0.04}$	72.6	12.53
Δ	0.0	−2.3	−2.0	−0.8	0.0	−4.6	0.0	−6.6	0.0	−3.1	−0.4	−3.5
ASC-β	$77.3_{\pm1.3}$	$9.84_{\pm0.34}$	$48.1_{\pm1.0}$	$12.37_{\pm0.14}$	$81.2_{\pm0.8}$	$7.40_{\pm0.07}$	$94.2_{\pm0.2}$	$5.00_{\pm0.03}$	$62.0_{\pm0.3}$	$8.55_{\pm0.06}$	72.6	8.63
Δ	0.0	−6.2	−1.9	−3.6	0.0	−8.6	0.0	−11.0	−0.1	−7.4	−0.4	−7.4
ASC-Dir	$77.3_{\pm1.3}$	$10.88_{\pm0.36}$	$48.1_{\pm1.0}$	$13.15_{\pm0.25}$	$81.3_{\pm0.8}$	$7.79_{\pm0.13}$	$94.2_{\pm0.2}$	$5.17_{\pm0.04}$	$62.0_{\pm0.3}$	$9.07_{\pm0.06}$	72.6	9.21
Δ	0.0	−5.1	−1.9	−2.8	+0.1	−8.2	0.0	−10.8	−0.1	−6.9	−0.4	−6.8
DSC	$74.3_{\pm2.1}$	$6.43_{\pm0.63}$	$46.4_{\pm2.5}$	$\mathbf{6.69}_{\pm0.29}$	$79.7_{\pm0.4}$	$4.85_{\pm0.06}$	$93.9_{\pm0.3}$	$2.85_{\pm0.06}$	$60.9_{\pm0.4}$	$\mathbf{5.18}_{\pm0.10}$	71.0	**5.20**†
Δ	−3.0	−9.6	−3.6	−9.3	−1.5	−11.2	−0.3	−13.2	−1.2	−10.8	−1.9	−10.8
CGES-LNS (arith)	$\mathbf{77.7}_{\pm1.5}$	$7.77_{\pm0.32}$	$47.8_{\pm1.3}$	$11.90_{\pm0.16}$	$81.3_{\pm0.5}$	$5.99_{\pm0.11}$	$94.1_{\pm0.3}$	$2.41_{\pm0.02}$	$61.9_{\pm0.3}$	$8.24_{\pm0.05}$	72.6	7.26
Δ	+0.4	−8.2	−2.2	−4.1	+0.1	−10.0	−0.1	−13.6	−0.2	−7.8	−0.4	−8.7
CGES-LNS (geom)	$77.3_{\pm1.3}$	$6.05_{\pm0.37}$	$47.4_{\pm1.1}$	$13.62_{\pm0.13}$	$81.1_{\pm0.3}$	$5.65_{\pm0.14}$	$94.1_{\pm0.2}$	$3.04_{\pm0.02}$	$61.9_{\pm0.4}$	$8.95_{\pm0.05}$	72.4	7.46
Δ	0.0	−9.9	−2.6	−2.4	−0.1	−10.3	−0.1	−13.0	−0.2	−7.1	−0.6	−8.5
CGES-DeepConf (B10)	$77.7_{\pm1.5}$	$5.62_{\pm0.37}$	$47.9_{\pm1.1}$	$12.36_{\pm0.15}$	$81.0_{\pm0.5}$	$4.89_{\pm0.10}$	$94.1_{\pm0.3}$	$2.41_{\pm0.02}$	$61.9_{\pm0.3}$	$8.26_{\pm0.05}$	72.5	6.71
Δ	+0.4	−10.4	−2.1	−3.6	−0.2	−11.1	−0.1	−13.6	−0.2	−7.7	−0.4	−9.3
CGES-DeepConf (tail)	$77.0_{\pm1.8}$	$8.64_{\pm0.25}$	$47.7_{\pm1.2}$	$11.48_{\pm0.15}$	$81.2_{\pm0.5}$	$5.98_{\pm0.14}$	$94.1_{\pm0.3}$	$2.41_{\pm0.02}$	$61.9_{\pm0.3}$	$8.23_{\pm0.05}$	72.4	7.35
Δ	−0.3	−7.4	−2.3	−4.5	0.0	−10.0	−0.1	−13.6	−0.2	−7.8	−0.6	−8.7
CGES-MARS	$77.3_{\pm1.3}$	$\mathbf{5.20}_{\pm0.34}$	$47.9_{\pm1.1}$	$14.21_{\pm0.31}$	$81.3_{\pm0.5}$	$5.89_{\pm0.13}$	$94.2_{\pm0.1}$	$3.11_{\pm0.03}$	$62.0_{\pm0.3}$	$8.80_{\pm0.05}$	72.6	7.44
Δ	0.0	−10.8	−2.1	−1.8	+0.1	−10.1	0.0	−12.9	−0.1	−7.2	−0.4	−8.6
CGES-RM	$66.7_{\pm0.0}$	$13.60_{\pm0.00}$	$45.2_{\pm2.3}$	$14.51_{\pm0.10}$	$\mathbf{81.6}_{\pm0.8}$	$\mathbf{2.88}_{\pm0.07}$	$\mathbf{94.9}_{\pm0.4}$	$\mathbf{1.25}_{\pm0.03}$	$58.3_{\pm0.6}$	$12.31_{\pm0.03}$	69.3	8.91
Δ	−10.6	−2.4	−4.8	−1.5	+0.4	−13.1	+0.7	−14.8	−3.8	−3.7	−3.6	−7.1

†DSC requires an additional LLM-based difficulty ranking step using the same inference model ($\approx$0.6 amortized calls/question; each call ranks B=5 questions jointly), excluded from #Calls but included in wall-clock time.

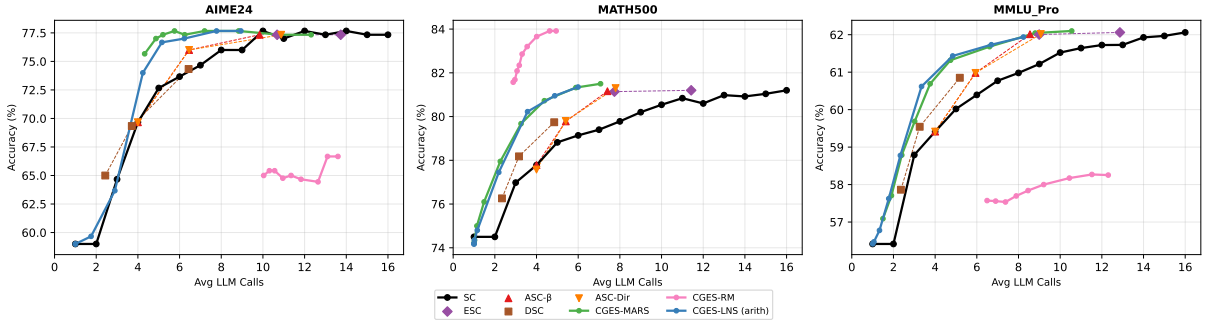


Figure 3: Accuracy vs. average LLM calls at B=16. SC (black) staircases through B=4, 8, 16. CGES-LNS / CGES-MARS (solid) sweep $\gamma \in [0.7, 0.9999]$. DSC / ASC / ESC (dashed) plot one point per budget. Points above-left of SC are Pareto-dominant.

Table 2: Wall-clock time per question (s) at $B = 16$, assuming LLM batch size 1. Wall = avg_calls $\times$ (T_{LLM} + T_{conf}).

Method	AIME24	GPQA	MATH500	GSM8K	MMLU-Pro	Avg
SC	136.01	88.25	4.24	1.18	2.45	46.43
ESC (w=4)	90.80	71.03	2.05	0.38	1.38	33.13
ESC (w=8)	116.74	83.85	3.03	0.70	1.97	41.26
ASC-β	83.65	68.21	1.96	0.37	1.31	31.10
ASC-Dir	92.46	72.54	2.07	0.38	1.39	33.77
DSC	54.86	**36.98**	1.32	0.24	**0.84**	**18.85**
CGES-LNS (arith)	66.31	65.83	1.60	0.18	1.27	27.04
CGES-LNS (geom)	51.57	75.37	1.50	0.23	1.38	26.01
CGES-DeepConf (B10)	47.90	68.38	1.30	0.18	1.27	23.81
CGES-DeepConf (tail)	73.68	63.51	1.59	0.18	1.27	28.05
CGES-MARS	**47.43**	89.70	1.76	0.77	1.67	28.27
CGES-RM	124.64	86.00	**0.98**	**0.16**	2.80	42.92

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.

Limitations

CGES relies on confidence signals that may be imperfectly calibrated. While Theorem 3.5 shows robustness to noisy estimates, strongly miscalibrated signals can degrade performance. Additionally, the framework assumes a finite candidate set $\mathcal{U}$; if the correct answer is absent from $\mathcal{U}$, no method operating solely on $\mathcal{U}$ can recover it. Future work includes developing more accurate confidence estimators and extending the framework with an explicit abstention hypothesis.

References

Yasin Abbasi-Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvari. 2024. To believe or not to believe your LLM: Iterativeprompting for estimating epistemic uncertainty. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. 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, Singapore. Association for Computational Linguistics.

Yavuz Faruk Bakman, Duygu Nur Yaldiz, Baturalp Buyukates, Chenyang Tao, Dimitrios Dimitriadis, and Salman Avestimehr. 2024. MARS: Meaning-aware response scoring for uncertainty estimation in generative LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7752–7767, Bangkok, Thailand. Association for Computational Linguistics.

Debarun Bhattacharjya, Balaji Ganesan, Junkyu Lee, Radu Marinescu, Katya Mirylenka, Michael Glass, and Xiao Shou. 2025. SIMBA UQ: Similarity-based aggregation for uncertainty quantification in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 15880–15894, Suzhou, China. Association for Computational Linguistics.

Zhenni Bi, Kai Han, Chuanjian Liu, Yehui Tang, and Yunhe Wang. 2025. Forest-of-thought: Scaling test-time compute for enhancing LLM reasoning. In *Forty-second International Conference on Machine Learning*.

Yanxi Chen, Xuchen Pan, Yaliang Li, Bolin Ding, and Jingren Zhou. 2026. Provable scaling laws for the test-time compute of large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

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

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Yichao Fu, Xuewei Wang, Hao Zhang, Yuandong Tian, and Jiawei Zhao. 2026. Deep think with confidence. In *The Fourteenth International Conference on Learning Representations*.

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Chengsong Huang, Langlin Huang, and Jiaxin Huang. 2024a. Divide, reweight, and conquer: A logit arithmetic approach for in-context learning. *Preprint*, arXiv:2410.10074.

Chengsong Huang, Langlin Huang, Jixuan Leng, Jiacheng Liu, and Jiaxin Huang. 2025. Efficient test-time scaling via self-calibration. *Preprint*, arXiv:2503.00031.

Siyuan Huang, Zhiyuan Ma, Jintao Du, Changhua Meng, Weiqiang Wang, and Zhouhan Lin. 2024b. Mirror-consistency: Harnessing inconsistency in majority voting. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2408–2420, Miami, Florida, USA. Association for Computational Linguistics.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli

Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and 17 others. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

Jing Yu Koh, Stephen McAleer, Daniel Fried, and Ruslan Salakhutdinov. 2024. Tree search for language model agents. *Preprint*, arXiv:2407.01476.

Guillaume Lample, Marie-Anne Lachaux, Thibaut Lavril, Xavier Martinet, Amaury Hayat, Gabriel Ebner, Aurélien Rodriguez, and Timothée Lacroix. 2022. Hypertree proof search for neural theorem proving. *Preprint*, arXiv:2205.11491.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333, Toronto, Canada. Association for Computational Linguistics.

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. In *The Twelfth International Conference on Learning Representations*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*.

Andrey Malinin and Mark Gales. 2021. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *Preprint*, arXiv:2501.19393.

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*.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, and 467 others. 2026. Openai gpt-5 system card. *Preprint*, arXiv:2601.03267.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. 2025. Confidence improves self-consistency in LLMs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20090–20111, Vienna, Austria. Association for Computational Linguistics.

Harshil Vejendla, Haizhou Shi, Yibin Wang, Tunyu Zhang, Huan Zhang, and Hao Wang. 2025. Efficient uncertainty estimation via distillation of bayesian large language models. *Preprint*, arXiv:2505.11731.

Xinglin Wang, Shaoxiong Feng, Yiwei Li, Peiwen Yuan, Yueqi Zhang, Chuyi Tan, Boyuan Pan, Yao Hu, and Kan Li. 2025a. 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, Albuquerque, New Mexico. Association for Computational Linguistics.

Xinglin Wang, Yiwei Li, Shaoxiong Feng, Peiwen Yuan, Yueqi Zhang, Jiayi Shi, Chuyi Tan, Boyuan Pan, Yao Hu, and Kan Li. 2025b. Every rollout counts: Optimal resource allocation for efficient test-time scaling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. MMLU-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2025. Inference scaling laws: An empirical analysis of compute-optimal inference for LLM problem-solving. In *The Thirteenth International Conference on Learning Representations*.

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

Haobo Xu, Sirui Chen, Ruizhong Qiu, Yuchen Yan, Chen Luo, Monica Cheng, Jingrui He, and Hanghang Tong. 2026. Prune as you generate: Online rollout pruning for faster and better rlvr. *Preprint*, arXiv:2603.24840.

Duygu Nur Yaldiz, Yavuz Faruk Bakman, Baturalp Buyukates, Chenyang Tao, Anil Ramakrishna, Dimitrios Dimitriadis, Jieyu Zhao, and Salman Avestimehr. 2025. Do not design, learn: A trainable scoring function for uncertainty estimation in generative LLMs. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 691–713, Albuquerque, New Mexico. Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

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

A Full Proof of Lemma 3.1

Proof. By Bayes' rule,

$$\mathbb{P}(I{=}i \mid \mathrm{Obs}) = \frac{\mathbb{P}(\mathrm{Obs} \mid I{=}i)\,\mathbb{P}(I{=}i)}{\sum_{k=1}^{K} \mathbb{P}(\mathrm{Obs} \mid I{=}k)\,\mathbb{P}(I{=}k)}. \tag{4}$$

Under Assumption 1, the likelihood factorizes as $\mathbb{P}(\mathrm{Obs} \mid I = i) = \prod_{t=1}^{m} \mathbb{P}(R_t, C_t \mid I = i)$. Under the uniform prior (Assumption 2), $\mathbb{P}(I = i) = 1/K$ is constant in i and cancels between numerator and denominator of (4), yielding

$$\mathbb{P}(I{=}i \mid \mathrm{Obs}) = \frac{\prod_t \mathbb{P}(R_t, C_t \mid I{=}i)}{\sum_k \prod_t \mathbb{P}(R_t, C_t \mid I{=}k)}. \tag{5}$$

Next, by the chain rule, $\mathbb{P}(R_t, C_t \mid I = i) = \mathbb{P}(R_t \mid C_t, I = i)\,\mathbb{P}(C_t \mid I = i)$. Assumption 4 implies $\mathbb{P}(C_t \mid I = i)$ is the same for all i, hence we may write $\mathbb{P}(C_t \mid I = i) = \mathbb{P}(C_t)$. Substituting into (5) gives

$$\mathbb{P}(I{=}i \mid \mathrm{Obs}) = \frac{\prod_t \mathbb{P}(R_t \mid C_t, I{=}i)\,\mathbb{P}(C_t)}{\sum_k \prod_t \mathbb{P}(R_t \mid C_t, I{=}k)\,\mathbb{P}(C_t)}.$$

The factor $\prod_{t=1}^{m} \mathbb{P}(C_t)$ is common to every term in numerator and denominator and cancels, yielding (2). □

B Full Proof of Theorem 3.2

Proof. Without loss of generality, relabel so that the true index is $I = 1$. Define

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

It suffices to show $A_1/A_k \xrightarrow{\text{a.s.}} \infty$ for every $k \neq 1$, which implies $X_1 \xrightarrow{\text{a.s.}} 1$ and $X_k \xrightarrow{\text{a.s.}} 0$ for all $k \neq 1$.

Step 1. For a fixed $k \neq 1$, define

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

Under Assumption 3, conditioning on C_t,

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

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

Let $\theta_t := \frac{1-C_t}{K-1}$. Under the true model $I = 1$:

$$\mathbb{P}(R_t = a_1 \mid C_t, I = 1) = C_t,$$
$$\mathbb{P}(R_t = a_k \mid C_t, I = 1) = \theta_t.$$

Step 2. Taking the conditional expectation,

$$\mathbb{E}\Big[Y_k^{(t)} \mid C_t\Big] = (C_t - \theta_t) \log\Big(\tfrac{C_t}{\theta_t}\Big).$$

Since $(a - b)\log(a/b) > 0$ for $a \neq b$, this is strictly positive whenever $C_t \neq 1/K$. Noting $|Y_k^{(t)}| \leq |\log(C_t/\theta_t)|$, the integrability assumption of Theorem 3.2 gives $\mathbb{E}|Y_k^{(t)}| < \infty$; combined with $\mathbb{P}(C_t = 1/K) = 0$,

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

Step 3. By Assumption 1, $\{Y_k^{(t)}\}_{t=1}^{m}$ are i.i.d. with finite mean $\mu_k > 0$. The Strong Law of Large Numbers gives $\frac{1}{m}\sum_{t=1}^{m} Y_k^{(t)} \xrightarrow{\text{a.s.}} \mu_k > 0$, so $\sum_{t=1}^{m} Y_k^{(t)} \xrightarrow{\text{a.s.}} +\infty$. Since $\log(A_1/A_k) = \sum_{t=1}^{m} Y_k^{(t)}$,

$$\frac{A_1}{A_k} = \exp\Big(\textstyle\sum_{t=1}^{m} Y_k^{(t)}\Big) \xrightarrow{\text{a.s.}} \infty.$$

Therefore $A_k/A_1 \to 0$ a.s. for all $k \neq 1$, and so

$$X_1 = \frac{1}{1 + \sum_{k\neq 1} A_k/A_1} \xrightarrow{\text{a.s.}} 1,$$

$$X_k = \frac{A_k/A_1}{1 + \sum_{j\neq 1} A_j/A_1} \xrightarrow{\text{a.s.}} 0.$$

□

C Full Proof of Theorem 3.5

Proof. Without loss of generality, let $I = 1$. Define

$$A_j := \prod_{t=1}^{m} \mathbb{P}(R_t \mid C_t, I = j),$$

$$X_j := A_j \,/\, \textstyle\sum_{k=1}^{K} A_k.$$

It suffices to show $A_1/A_k \xrightarrow{\text{a.s.}} \infty$ for every $k \neq 1$, implying $X_1 \to 1$ and $X_k \to 0$ almost surely.

Step 1 (per-sample increment). Fix $k \neq 1$ and define

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

Writing $\theta_t = \frac{1-C_t}{K-1}$ and $\ell(C_t) := \log C_t - \log \theta_t = \log \frac{C_t(K-1)}{1-C_t}$, the ideal scoring rule of Eq. (3) gives

$$Y_k^{(t)} = \big(\mathbf{1}\{R_t = a_1\} - \mathbf{1}\{R_t = a_k\}\big)\, \ell(C_t),$$

since the contributions from indices $R_t \notin \{a_1, a_k\}$ cancel.

Step 2 (expectation under the true joint). Taking expectations under the true joint distribution of (R_t, C_t),

$$\mathbb{E}\Big[Y_k^{(t)}\Big] = \mathbb{E}\big[\big(\mathbf{1}\{R_t{=}a_1\} - \mathbf{1}\{R_t{=}a_k\}\big)\ell(C_t)\big] = \mu_k.$$

The integrability assumption $\mathbb{E}[|\ell(C_t)|] < \infty$ together with $|\mathbf{1}\{R_t = a_1\} - \mathbf{1}\{R_t = a_k\}| \leq 1$ ensures $\mathbb{E}[|Y_k^{(t)}|] < \infty$. No further structural assumption on the joint distribution of (R_t, C_t) is required.

Step 3 (SLLN). By Assumption 1, $\{Y_k^{(t)}\}_{t=1}^m$ are i.i.d. with mean μ_k. The Strong Law of Large Numbers gives $\frac{1}{m}\sum_t Y_k^{(t)} \xrightarrow{\text{a.s.}} \mu_k$. If $\mu_k > 0$, then $\sum_t Y_k^{(t)} \xrightarrow{\text{a.s.}} +\infty$ and hence $A_1/A_k = \exp\big(\sum_t Y_k^{(t)}\big) \xrightarrow{\text{a.s.}} \infty$. Combining over all $k \neq 1$, $X_1 \xrightarrow{\text{a.s.}} 1$ and $X_k \xrightarrow{\text{a.s.}} 0$.

Converse. If $\mu_{k^\star} < 0$ for some $k^\star \neq 1$, then $\sum_t Y_{k^\star}^{(t)} \xrightarrow{\text{a.s.}} -\infty$, so $A_1/A_{k^\star} \xrightarrow{\text{a.s.}} 0$ and $X_1 \xrightarrow{\text{a.s.}} 0$. □

D Consistency of Dynamic-Support CGES

Theorems 3.2 and 3.5 are stated for a fixed candidate set $\mathcal{U}$ and a fixed support size K. The online implementation of Algorithm 1 instead uses a growing candidate set: after m samples it scores over the distinct answers observed so far together with the explicit OTHER/unseen bucket of Section 3.1. This section shows that the realistic-regime guarantee of Theorem 3.5 extends to this dynamic-support version once the observed support stabilizes.

Setup. Let R_t denote the concrete post-processed answer string produced at sample t, let $C_t \in (0, 1)$ be its confidence score, and let $a^\star$ denote the true answer A. After m samples, let $\mathcal{V}_m := \{R_1, \ldots, R_m\}$ be the observed concrete support, $\mathcal{U}_m := \mathcal{V}_m \cup \{\perp\}$ the dynamic candidate set, and $K_m := |\mathcal{U}_m|$ its size, where $\perp$ denotes the OTHER/unseen category. Each observed sample is a concrete answer, so $R_t \neq \perp$ for all t. Consistent with the dynamic-support implementation described in Section 3.3, at each time m the score is recomputed over all m samples using the current candidate set $\mathcal{U}_m$ and current support size K_m. Writing $\theta_{m,t} := (1 - C_t)/(K_m - 1)$ for the per-sample residual mass, the unnormalized and normalized scores of a candidate $a \in \mathcal{U}_m$ are

$$A_m(a) := \prod_{t=1}^{m} C_t^{\mathbf{1}\{R_t = a\}}\, \theta_{m,t}^{\mathbf{1}\{R_t \neq a\}},$$

$$X_m(a) := \frac{A_m(a)}{\sum_{b\in\mathcal{U}_m} A_m(b)}.$$

For a support size K, write $\ell_K(c) := \log \frac{c(K-1)}{1-c}$; this is the per-sample log-likelihood ratio ℓ of Theorem 3.5 with the support size made explicit.

Proposition D.1 (Dynamic-support consistency after support stabilization)**.** *Fix a realization of the observation sequence $\{(R_t, C_t)\}_{t\geq 1}$, and suppose its observed candidate set stabilizes: there exist a finite time $T < \infty$ and a finite candidate set $\mathcal{U}_\infty$ such that $\mathcal{U}_m = \mathcal{U}_\infty$ for all $m \geq T$. Let $K_\infty := |\mathcal{U}_\infty|$, suppose the true answer satisfies $a^\star \in \mathcal{U}_\infty$, and for each competitor $b \in \mathcal{U}_\infty \setminus \{a^\star\}$ define the per-sample log-ratio increment*

$$Y_t^{(b)} := \big(\mathbf{1}\{R_t = a^\star\} - \mathbf{1}\{R_t = b\}\big) \cdot \ell_{K_\infty}(C_t).$$

If, for every such b,

$$\liminf_{m\to\infty} \frac{1}{m}\sum_{t=1}^{m} Y_t^{(b)} \;>\; 0,$$

then $X_m(a^\star) \to 1$ *and* $X_m(b) \to 0$ *for every* $b \in \mathcal{U}_\infty \setminus \{a^\star\}$.

Proof. For all $m \geq T$ the candidate set has stabilized: $\mathcal{U}_m = \mathcal{U}_\infty$ and $K_m = K_\infty$. Fix any competitor $b \in \mathcal{U}_\infty \setminus \{a^\star\}$. For $m \geq T$ every factor in $A_m(\cdot)$ uses K_∞, so the log-score ratio between $a^\star$ and b is

$$\log \frac{A_m(a^\star)}{A_m(b)} = \sum_{t=1}^{m} Y_t^{(b)}.$$

To see this, consider one sample t and write $\theta_{\infty,t} := (1 - C_t)/(K_\infty - 1)$. If $R_t = a^\star$, the contribution to $A_m(a^\star)$ is C_t and to $A_m(b)$ is $\theta_{\infty,t}$, so the log-ratio contribution is $\ell_{K_\infty}(C_t)$. If $R_t = b$, the same computation gives $-\ell_{K_\infty}(C_t)$. If $R_t \notin \{a^\star, b\}$, both candidates receive the identical factor $\theta_{\infty,t}$, so the contribution is zero. This also covers $b = \perp$, since $R_t \neq \perp$ for every observed sample. By the assumed positive empirical drift, $\sum_{t=1}^{m} Y_t^{(b)} \to +\infty$, hence $A_m(b)/A_m(a^\star) \to 0$ for every $b \in \mathcal{U}_\infty \setminus \{a^\star\}$. Since $\mathcal{U}_\infty$ is finite and $X_m(a^\star) = \big(1 + \sum_{b\neq a^\star} A_m(b)/A_m(a^\star)\big)^{-1}$, it follows that $X_m(a^\star) \to 1$, and hence $X_m(b) \to 0$ for every $b \neq a^\star$. □

Proposition D.1 is a deterministic statement about a single observation sequence. The next corollary gives a stochastic condition under which its hypotheses, and hence consistency, hold almost surely.

Corollary D.2 (Finite-support dynamic CGES)**.** *Assume* $\{(R_t, C_t)\}_{t\geq 1}$ *are i.i.d. (Assumption 1) and the concrete answer distribution has finite support* $\mathcal{V} := \{a : \mathbb{P}(R_t = a) > 0\}$*, with* $a^\star \in \mathcal{V}$*. Then* $\mathcal{U}_m$ *stabilizes almost surely to* $\mathcal{U}_\infty = \mathcal{V} \cup \{\perp\}$*. Let* $K_\infty := |\mathcal{V}| + 1$ *and assume* $\mathbb{E}\big[|\ell_{K_\infty}(C_t)|\big] < \infty$*. Let* $Y_t^{(b)}$ *be the increment of Proposition D.1, and for the OTHER bucket set* $Y_t^{(\perp)} := \mathbf{1}\{R_t = a^\star\}\,\ell_{K_\infty}(C_t)$*, which is the* $b = \perp$ *instance since* $R_t \neq \perp$*. If the per-sample drifts are positive, that is,* $\mu_b := \mathbb{E}[Y_t^{(b)}] > 0$ *for every concrete competitor* $b \in \mathcal{V} \setminus \{a^\star\}$ *and* $\mu_\perp := \mathbb{E}[Y_t^{(\perp)}] > 0$*, then* $X_m(a^\star) \xrightarrow{\text{a.s.}} 1$ *and* $X_m(b) \xrightarrow{\text{a.s.}} 0$ *for every* $b \in \mathcal{U}_\infty \setminus \{a^\star\}$.

Proof. For any fixed $a \in \mathcal{V}$, let $p_a := \mathbb{P}(R_t = a) > 0$. Then $\mathbb{P}(a \notin \mathcal{V}_m) = (1 - p_a)^m \to 0$, so the first hitting time $T_a := \inf\{m \geq 1 : R_m = a\}$ is finite almost surely. Since $\mathcal{V}$ is finite, $T := \max_{a\in\mathcal{V}} T_a$ is finite almost surely, and $\mathcal{U}_m = \mathcal{V} \cup \{\perp\} = \mathcal{U}_\infty$ for all $m \geq T$. The stabilization hypothesis of Proposition D.1 therefore holds almost surely.

It remains to verify the positive empirical drift. Fix a concrete competitor $b \in \mathcal{V}\setminus\{a^\star\}$. The integrability assumption gives $\mathbb{E}|Y_t^{(b)}| \leq \mathbb{E}[|\ell_{K_\infty}(C_t)|] < \infty$, and by hypothesis $\mathbb{E}[Y_t^{(b)}] = \mu_b > 0$. Since the samples are i.i.d., the Strong Law of Large Numbers gives $\frac{1}{m}\sum_{t=1}^{m} Y_t^{(b)} \xrightarrow{\text{a.s.}} \mu_b > 0$. The same argument applied to $Y_t^{(\perp)}$ gives $\frac{1}{m}\sum_{t=1}^{m} Y_t^{(\perp)} \xrightarrow{\text{a.s.}} \mu_\perp > 0$. Thus the positive-drift hypothesis of Proposition D.1 holds almost surely for every competitor in $\mathcal{U}_\infty \setminus \{a^\star\}$, and the claim follows. □

Corollary D.3 (Eventual threshold crossing)**.** *Under the hypotheses of Proposition D.1, fix any threshold* $\gamma \in (0, 1)$*. There exists a finite time* M_γ *such that* $X_m(a^\star) \geq \gamma$ *and* $a^\star \in \arg\max_{a\in\mathcal{U}_m} X_m(a)$ *for all* $m \geq M_\gamma$*. Consequently, a dynamic-support CGES run with no finite budget, or with finite budget* $B \geq M_\gamma$*, that has not stopped before* M_γ*, stops no later than* M_γ *and returns* $a^\star$*. If* $B < M_\gamma$*, the run may instead stop at budget exhaustion before the threshold is crossed.*

Proof. By Proposition D.1, $X_m(a^\star) \to 1$ and $X_m(b) \to 0$ for every $b \in \mathcal{U}_\infty \setminus \{a^\star\}$. Since $\gamma < 1$, there is a finite M_1 with $X_m(a^\star) \geq \gamma$ for all $m \geq M_1$. Since $\mathcal{U}_\infty$ is finite and every competing score converges to zero, there is a finite M_2 with $X_m(a^\star) > X_m(b)$ for every $b \in \mathcal{U}_\infty \setminus \{a^\star\}$ and all $m \geq M_2$. Set $M_\gamma := \max\{M_1, M_2, T\}$. For all $m \geq M_\gamma$ the support has stabilized, $a^\star$ meets the threshold, and $a^\star$ is the strict argmax. The stopping claim then follows for a run with no finite budget or with $B \geq M_\gamma$; if $B < M_\gamma$, the run may stop at budget exhaustion before m reaches M_γ. □

Remark D.4 (Role of the OTHER bucket). The condition $\mu_\perp > 0$ is needed when $\perp$ is included in the normalized score and in the stopping rule $\max_{a\in\mathcal{U}_m} X_m(a) \geq \gamma$. If $\perp$ is used only as a bookkeeping device and the prediction is restricted to concrete observed answers, consistency of the concrete argmax requires only $\mu_b > 0$ for every $b \in \mathcal{V} \setminus \{a^\star\}$.

Remark D.5 (Scope of the result). In the CGES setting the stabilization hypothesis of Proposition D.1 holds automatically. An LLM with a finite vocabulary and a bounded generation length emits answers from a finite set, so the observed candidate set $\mathcal{U}_m$ is a non-decreasing family of subsets of a finite universe and necessarily stabilizes after a finite, run-dependent time on every realization. Consequently the dynamic-support extension introduces no assumption beyond those used for the fixed-support Theorem 3.5; in particular, the finite-support hypothesis of Corollary D.2 is always met.

The genuine finite-budget caveat is different. Proposition D.1 is asymptotic and does not rule out premature stopping before $a^\star$ has entered the observed support: if $a^\star \notin \mathcal{U}_m$, no rule restricted to $\mathcal{U}_m$ can return it. This is intrinsic to any method whose candidate set is built from observed samples; in finite-budget runs the risk is controlled through the stopping threshold and can be reduced further by requiring a small minimum number of samples before early stopping is allowed.

E Full Proof of Proposition 3.3

Proof. Let $\mathcal{F}_m := \sigma\big(\{(R_t, C_t)\}_{t=1}^m\big)$ denote the natural filtration generated by the first m samples, and write $\mathrm{Obs}_m = \{(R_t, C_t)\}_{t=1}^m$. We write $X_i^{(m)}$ for the score on candidate a_i computed by Algorithm 1 from Obs_m.

Step 1 (Posterior identification). By Lemma 3.1, Assumptions 1, 2, and 4 imply that the true Bayesian posterior over the correct index I given Obs_m admits the closed form of Eq. (2). Under Assumption 3, the conditional likelihood $\mathbb{P}(R_t \mid C_t, I = i)$ takes the explicit one-vs-rest form of Eq. (3), which is precisely the auxiliary likelihood used by Algorithm 1. Consequently, the normalized score satisfies

$$X_i^{(m)} \;=\; \mathbb{P}(I = i \mid \mathrm{Obs}_m), \tag{6}$$

for every $i \in [K]$ and every $m \geq 1$.

Step 2 (Stopping time). Define $\tau_\gamma := \inf\{m \geq 1 : \max_{i\in[K]} X_i^{(m)} \geq \gamma\}$, with $\inf \emptyset := +\infty$. Since $\max_i X_i^{(m)}$ is $\mathcal{F}_m$-measurable for each m, the event $\{\tau_\gamma \leq m\} = \bigcup_{j=1}^m \{\max_i X_i^{(j)} \geq \gamma\}$ lies in $\mathcal{F}_m$, so τ_γ is a stopping time with respect to $\{\mathcal{F}_m\}_{m\geq 1}$. Algorithm 1 terminates at $T := \min(\tau_\gamma, B)$ and returns

$$\hat{I} \;:=\; \arg\max_{i\in[K]} X_i^{(T)},$$

breaking ties arbitrarily among maximizers.

Step 3 (Posterior identity at the stopping time). Since T is bounded by B, the events $\{T = m\}$ for $m = 1, \ldots, B$ partition the sample space; each is $\mathcal{F}_m$-measurable because T is a stopping time. Combining this partition with the deterministic-time identity (6),

$$\begin{aligned}\mathbb{P}(I{=}i \mid \mathcal{F}_T) &= \sum_{m=1}^{B} \mathbf{1}\{T{=}m\}\, \mathbb{P}(I{=}i \mid \mathcal{F}_m) \\ &= \sum_{m=1}^{B} \mathbf{1}\{T{=}m\}\, X_i^{(m)} \\ &= X_i^{(T)},\end{aligned} \tag{7}$$

so the posterior identity extends from deterministic times to the random stopping time T.

Step 4 (Risk bound on the threshold-stopping event). On the event $\{\tau_\gamma \leq B\}$, we have $T = \tau_\gamma$ and, by the definition of τ_γ, $\max_i X_i^{(\tau_\gamma)} \geq \gamma$. Combining (7) with the definition of $\hat{I}$,

$$\begin{aligned}\mathbb{P}\big(I{=}\hat{I} \;\big|\; \mathcal{F}_{\tau_\gamma}\big) &= X_{\hat{I}}^{(\tau_\gamma)} \\ &= \max_{i\in[K]} X_i^{(\tau_\gamma)} \;\geq\; \gamma.\end{aligned}$$

Taking the complement yields

$$\begin{aligned}\mathbb{P}\big(I \neq \hat{I} \;\big|\; \mathcal{F}_{\tau_\gamma}\big) &= 1 - X_{\hat{I}}^{(\tau_\gamma)} \\ &\leq 1 - \gamma,\end{aligned}$$

which is the bound in the proposition. □

Remark (budget-exhaustion event). The bound in Proposition 3.3 applies on $\{\tau_\gamma \leq B\}$, where the algorithm halts because the posterior on the top candidate exceeds γ. On the complementary event $\{\tau_\gamma > B\}$, Algorithm 1 halts at the budget with $\max_i X_i^{(B)} < \gamma$; the exact identity $\mathbb{P}(I \neq \hat{I} \mid \mathrm{Obs}_B) = 1 - \max_i X_i^{(B)}$ continues to hold by Step 1 (which is independent of the stopping rule), but the conditional risk is no longer controlled by $1 - \gamma$. Empirically, the budget-exhaustion event is rare for moderate γ and B (Section 4).

Corollary E.1 (Bayes-optimal decision and earliest certified stopping)**.** *Under Assumptions 1–4, for every finite $m \geq 1$ and any (possibly randomized) decision rule δ_m measurable with respect to $\mathcal{F}_m$,*

$$\mathbb{P}\big(\delta_m \neq I \;\big|\; \mathcal{F}_m\big) \;\geq\; 1 - \max_{i\in[K]} X_i^{(m)},$$

with equality attained by $\hat{I}_m \in \arg\max_i X_i^{(m)}$. *The same inequality holds at any bounded* $\{\mathcal{F}_m\}$*-stopping time* $T \le B$*, with* $\mathcal{F}_m$ *replaced by* $\mathcal{F}_T$*. Consequently, for* $\gamma \in (0,1)$ *and* $\tau_\gamma := \inf\{m \ge 1 : \max_i X_i^{(m)} \ge \gamma\}$*, any* $\{\mathcal{F}_m\}$*-measurable rule whose conditional success probability at its bounded stopping time* T *is at least* γ *must satisfy* $T \ge \tau_\gamma$ *almost surely.*

Proof. By Step 1 of the proof of Proposition 3.3, $X_i^{(m)} = \mathbb{P}(I = i \mid \mathcal{F}_m)$. Conditional on $\mathcal{F}_m$, the randomization in δ_m is independent of I, so

$$\mathbb{P}\big(\delta_m = I \;\big|\; \mathcal{F}_m\big) = \sum_{i=1}^{K} \mathbb{P}(\delta_m = i \mid \mathcal{F}_m)\, X_i^{(m)} \le \max_{i\in[K]} X_i^{(m)},$$

with equality whenever δ_m is supported on $\arg\max_i X_i^{(m)}$, in particular for $\hat{I}_m$. Taking the complement gives the displayed inequality at deterministic times. Step 3 of the same proof established the stopped-time identity $\mathbb{P}(I = i \mid \mathcal{F}_T) = X_i^{(T)}$ for every bounded $\{\mathcal{F}_m\}$-stopping time $T \le B$; the same conditional argument applied at $\mathcal{F}_T$ yields the inequality at T. Finally, $\mathbb{P}(\delta_T = I \mid \mathcal{F}_T) \ge \gamma$ forces $\max_i X_i^{(T)} \ge \gamma$, and by the definition of τ_γ this can occur only when $T \ge \tau_\gamma$. □

F Hyperparameters and Decoding

For all benchmarks we use temperature 0.7 and top-p=0.95 for sampling. Self-consistency runs with budget $B \in \{4, 8, 16\}$. ESC uses window sizes $w \in \{4, 8\}$. ASC uses the original authors' threshold settings. For DSC we use the difficulty bucket schedule from Wang et al. (2025a). Each table cell reports the mean and standard deviation over 10 random seeds.

Prompt Templates. Dataset-specific answer-format constraints simplify parsing. Prompt templates are shown in Figure 4.

G Baselines

Self-Consistency (SC) (Wang et al., 2023) samples a fixed budget of B responses and returns the majority vote. **Early-Stopping Self-Consistency (ESC)** (Li et al., 2024) halts once the last w predictions agree. **Adaptive-Consistency (ASC)** (Aggarwal et al., 2023) stops once a Bayesian estimate of the majority is stable. We use the Beta-binomial variant (ASC-β) and the Dirichlet-multinomial variant (ASC-Dir). **Difficulty-Adaptive Self-Consistency (DSC)** (Wang et al., 2025a) adds an LLM-based difficulty ranking step that allocates more calls to harder questions. The ranking step is excluded from #Calls but included in the wall-clock numbers in Table 2.

```
Question: {sample['question']}
Provide your step-by-step reasoning
first, and then print "The answer is
\boxed{X}", where X is the final answer,
at the end of your response.
```

(a) Prompt used for **MATH500, AIME24, GSM8K**

```
Question: {sample['question']}
Provide your step-by-step reasoning first, and
then print "The answer is (X)", where X is the
answer choice (one capital letter), at the end
of your response.
```

(b) Prompt used for **GPQA, MMLU_Pro**

Figure 4: Two prompt templates for evaluation.

H Confidence Estimation Strategies

We compare several strategies for estimating the *scalar* confidence $C_t \in (0,1)$ of each sampled answer R_t. Each strategy maps model outputs (token probabilities or reward scores) to a scalar C_t that we interpret as an estimate of the probability that the corresponding response R_t is correct. These estimates need not be perfectly calibrated. They serve as noisy confidence signals in the sense formalized in Section 3.1 and Theorem 3.5. Our framework operates at the response level, but confidence estimation relies on finer token-level granularities. Specifically, let a response R_t consist of a token sequence $\{T_1, \ldots, T_L\}$ with associated autoregressive probabilities $\{p_1, \ldots, p_L\}$. We use lowercase p_ℓ to denote these token-level probabilities from the underlying language model, and uppercase $\mathbb{P}(\cdot)$ for probabilities over candidate answers $a_j \in \mathcal{U}$ as in Section 3.1. The role of the present section is to map token-level scores (or reward-model outputs) into a single scalar confidence C_t. Building on this, we describe three token-based approaches and one verifier-based alternative.

Length-Normalized Scoring (LNS) (Malinin and Gales, 2021). A natural way to quantify the likelihood of a response is by averaging over token probabilities. The *geometric mean* yields the standard length-normalized score $\text{LNS}_{\text{geom}} = \exp\big(\frac{1}{L}\sum_{\ell=1}^{L}\log p_\ell\big)$, while the *arithmetic mean* provides a 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 denote them in results as *LNS (geom)* and *LNS (arith)*.

DeepConf (Fu et al., 2026). DeepConf forms a confidence signal from the least certain parts of the generation rather than the whole sequence, which makes it sensitive to local reasoning errors. We use two variants. *DeepConf-B10* averages the lowest-probability tokens, taking the mean over the bottom 10% of $\{p_1, \ldots, p_L\}$. *DeepConf-Tail* averages the token probabilities in the final segment of the response (the last 20% of tokens), where the answer is committed. Both are squashed to $(0, 1)$ and set as C_t, denoted *DeepConf-B10* and *DeepConf-Tail*.

MARS (Step-Weighted Scoring) (Bakman et al., 2024). The MARS method generalizes LNS by assigning different weights to different positions in the sequence. Each token T_ℓ receives an exponent $w(R, Q, L, \ell) \triangleq \frac{1}{2L} + \frac{u(R,Q,\ell)}{2}$, where R denotes the full textual response string, so the overall score becomes $\bar{P}(R \mid Q, \theta) = \prod_{\ell=1}^{L} p_\ell^{\,w(R,Q,L,\ell)}$, where θ denotes the parameters of the language model. Here $u(R, Q, \ell)$ is an importance score for token T_ℓ, such as the semantic change in the output when masking that token. To address the expense of per-token weighting for long reasoning responses, we adopt a *step-wise* variant of MARS. Instead of per-token weights, we assign weights at the granularity of reasoning steps or sentence segments. Although the step-importance score is recomputed each iteration, this overhead (6 layers, ∼50M parameters) is negligible compared to the 7B-parameter inference model we query for R_t. We denote the resulting confidence as $C_t = \text{MARS}$.

Reward Model Confidence. In addition to the above token-level methods, we also consider a model-based approach that evaluates entire responses directly. A trained reward model assigns a quality score to each R_t, which we use as C_t. In our study we use the `Qwen2.5-Math-PRM-7B` process reward model (Zhang et al., 2025), which outputs a scalar in $(0, 1)$ that correlates with alignment to the ground truth on math-style reasoning. We include it as a near-optimal reference rather than a deployable setting, since the PRM is trained specifically for step-level scoring and its training distribution overlaps with several benchmarks, so its signal approximates an upper bound on confidence quality on in-domain tasks. We denote this variant as *RM Confidence*.

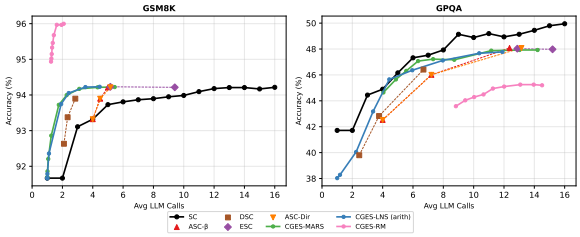


Figure 5: Pareto frontier for GSM8K and GPQA (same setup as Figure 3). CGES dominates the baselines on GSM8K where token-probability confidence is well-calibrated. On GPQA the available signals do not discriminate well, so CGES tracks SC without improving on it.

I Additional Results at Smaller Budgets

We re-run the main experiments at $B{=}4$ and $B{=}8$ (Tables 3 and 4). ESC is omitted at $B{=}4$ because its minimum window size $w{\geq}4$ exceeds the budget. The picture is the same as at $B{=}16$. CGES variants match SC accuracy within a small margin while using fewer calls. CGES-RM on math tasks improves accuracy further. DSC stays the most aggressive on call count but the accuracy gap to SC widens to -2.7pp at $B{=}4$ and -3.1pp at $B{=}8$.

Figure 5 adds GSM8K and GPQA. On GSM8K, CGES-RM reaches roughly 95% accuracy at 1.2 calls, illustrating what a strong confidence signal enables. On GPQA, no variant strictly dominates SC, which matches the calibration analysis in Appendix J.

J Confidence Signal Quality and CGES Efficiency

The main body claims that CGES helps only when the confidence signal is informative, and that GPQA is the case where it is not. This appendix makes that claim quantitative. We measure signal quality in three ways and show each one tracks CGES efficiency, defined as $1 - \overline{\#\text{Calls}}/B$, where $\overline{\#\text{Calls}}$ is the average number of LLM calls per question and B is the budget.

Calibration and discrimination. A confidence signal can be good in two distinct ways. *Cal-*

Table 3: Results at budget $B = 4$. CGES rows use the efficient γ (smallest γ matching SC$\pm$0.2 pp). Mean$\pm$std across 10 seeds. Δ rows show difference vs. SC (green=improvement, red=regression). Last column: avg over tasks. Bold = best, underline = second-best per column. ESC omitted: its minimum window size ($w \geq 4$) exceeds this budget, making comparison meaningless.

Method	AIME24		GPQA		MATH500		GSM8K		MMLU_Pro		**Avg**	
	Acc (%)	#Calls	Acc (%)	#Calls	Acc (%)	#Calls	Acc (%)	#Calls	Acc (%)	#Calls	Acc (%)	#Calls
SC	$69.7_{\pm 5.5}$	$4.00_{\pm 0.00}$	$\mathbf{44.9}_{\pm 1.8}$	$4.00_{\pm 0.00}$	$77.8_{\pm 0.9}$	$4.00_{\pm 0.00}$	$93.3_{\pm 0.4}$	$4.00_{\pm 0.00}$	$59.4_{\pm 0.6}$	$4.00_{\pm 0.00}$	69.0	4.00
ASC-β	$69.7_{\pm 5.5}$	$4.00_{\pm 0.00}$	$42.5_{\pm 1.9}$	$4.00_{\pm 0.00}$	$77.8_{\pm 0.9}$	$4.00_{\pm 0.00}$	$93.3_{\pm 0.4}$	$4.00_{\pm 0.00}$	$59.4_{\pm 0.6}$	$4.00_{\pm 0.00}$	68.5	4.00
Δ	0.0	0.0	-2.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.5	0.0
ASC-Dir	$69.7_{\pm 5.5}$	$4.00_{\pm 0.00}$	$42.5_{\pm 1.9}$	$4.00_{\pm 0.00}$	$77.6_{\pm 0.8}$	$4.00_{\pm 0.00}$	$93.3_{\pm 0.4}$	$4.00_{\pm 0.00}$	$59.4_{\pm 0.6}$	$4.00_{\pm 0.00}$	68.5	4.00
Δ	0.0	0.0	-2.4	0.0	-0.2	0.0	0.0	0.0	0.0	0.0	-0.5	0.0
DSC	$65.0_{\pm 4.8}$	$\mathbf{2.43}_{\pm 0.09}$	$39.8_{\pm 1.8}$	$\mathbf{2.46}_{\pm 0.04}$	$76.3_{\pm 1.1}$	$\underline{2.34}_{\pm 0.01}$	$92.6_{\pm 0.3}$	$2.10_{\pm 0.01}$	$57.9_{\pm 0.9}$	$\mathbf{2.36}_{\pm 0.01}$	66.3	**2.34**[†]
Δ	-4.7	-1.6	-5.1	-1.5	-1.5	-1.7	-0.7	-1.9	-1.5	-1.6	-2.7	-1.7
CGES-LNS (arith)	$69.7_{\pm 3.8}$	$3.53_{\pm 0.06}$	$\underline{44.2}_{\pm 1.5}$	$4.00_{\pm 0.00}$	$78.2_{\pm 0.9}$	$2.58_{\pm 0.02}$	$\underline{93.4}_{\pm 0.4}$	$1.83_{\pm 0.01}$	$59.5_{\pm 0.7}$	$\underline{2.75}_{\pm 0.02}$	69.0	2.94
Δ	0.0	-0.5	-0.7	0.0	$+0.4$	-1.4	$+0.1$	-2.2	$+0.1$	-1.2	0.0	-1.1
CGES-LNS (geom)	$\mathbf{70.7}_{\pm 4.2}$	$2.89_{\pm 0.14}$	$44.0_{\pm 1.8}$	$4.00_{\pm 0.00}$	$77.9_{\pm 0.9}$	$2.38_{\pm 0.01}$	$93.3_{\pm 0.4}$	$\underline{1.63}_{\pm 0.02}$	$59.3_{\pm 0.7}$	$3.01_{\pm 0.01}$	$\underline{69.1}$	2.78
Δ	$+1.0$	-1.1	-0.9	0.0	$+0.1$	-1.6	0.0	-2.4	-0.1	-1.0	0.0	-1.2
CGES-DeepConf (B10)	$70.0_{\pm 4.5}$	$2.97_{\pm 0.11}$	$44.1_{\pm 2.0}$	$4.00_{\pm 0.00}$	$\underline{78.5}_{\pm 0.9}$	$2.59_{\pm 0.02}$	$93.4_{\pm 0.4}$	$1.83_{\pm 0.01}$	$\underline{59.5}_{\pm 0.6}$	$2.77_{\pm 0.02}$	**69.1**	2.83
Δ	$+0.3$	-1.0	-0.8	0.0	$+0.7$	-1.4	$+0.1$	-2.2	$+0.1$	-1.2	$+0.1$	-1.2
CGES-DeepConf (tail)	$69.3_{\pm 4.9}$	$3.98_{\pm 0.03}$	$43.5_{\pm 2.2}$	$4.00_{\pm 0.01}$	$78.1_{\pm 0.8}$	$2.57_{\pm 0.02}$	$93.4_{\pm 0.4}$	$1.83_{\pm 0.01}$	$59.5_{\pm 0.7}$	$2.75_{\pm 0.02}$	68.8	3.03
Δ	-0.4	0.0	-1.4	0.0	$+0.3$	-1.4	$+0.1$	-2.2	$+0.1$	-1.2	-0.3	-1.0
CGES-MARS	$\underline{70.3}_{\pm 3.8}$	$\underline{2.67}_{\pm 0.09}$	$43.4_{\pm 1.7}$	$4.00_{\pm 0.00}$	$77.8_{\pm 1.0}$	$2.48_{\pm 0.01}$	$93.4_{\pm 0.4}$	$1.69_{\pm 0.01}$	$\mathbf{59.5}_{\pm 0.6}$	$3.03_{\pm 0.01}$	68.9	$\underline{2.78}$
Δ	$+0.6$	-1.3	-1.5	0.0	0.0	-1.5	$+0.1$	-2.3	$+0.1$	-1.0	-0.1	-1.2
CGES-RM	$52.3_{\pm 3.4}$	$3.98_{\pm 0.01}$	$42.5_{\pm 2.0}$	$\underline{3.99}_{\pm 0.01}$	$\mathbf{80.2}_{\pm 0.6}$	$\mathbf{1.63}_{\pm 0.02}$	$\mathbf{94.4}_{\pm 0.5}$	$\mathbf{1.13}_{\pm 0.01}$	$59.2_{\pm 0.6}$	$3.91_{\pm 0.00}$	65.7	2.93
Δ	-17.4	-0.0	-2.4	-0.0	$+2.4$	-2.4	$+1.1$	-2.9	-0.2	-0.1	-3.3	-1.1

[†]DSC requires an additional LLM-based difficulty ranking step using the same inference model ($\approx$0.6 amortized calls/question; each call ranks B=5 questions jointly), excluded from #Calls but included in wall-clock time.

ibration, measured by Expected Calibration Error (ECE), asks whether the average confidence matches the true accuracy. *Discrimination*, measured by AUC-ROC, asks whether the signal ranks correct responses above incorrect ones. The drift μ_k in Theorem 3.5 depends on the second, not the first. A signal can be well calibrated yet carry no drift if it cannot separate correct from incorrect responses. Table 5 reports both quantities and CGES efficiency for every (method, task) pair.

GPQA is the uninformative case. Table 5 shows that efficiency follows discrimination, not calibration. The sharpest evidence is CGES-RM on GPQA versus MATH500. The two have the same ECE (0.050), so they look equally well calibrated, yet their AUC differs sharply (0.685 vs. 0.954) and so does efficiency (0.128 vs. 0.711). On GPQA the reward model assigns scores narrowly clustered around the overall accuracy regardless of correctness, which makes ECE small but leaves AUC near random. Every other signal on GPQA shows the same pattern of near-random AUC, so the drift stays small for all of them and CGES neither saves calls nor fully matches SC accuracy there, exactly as Theorem 3.5 predicts for a weak drift. AIME24 is a second hard case for the reward model, where both ECE and AUC are poor.

Empirical drift. The quantity the theorem actually names is the drift. Its empirical estimate is

$$\hat{\mu} = \mathbb{E}[\ell(C_t) \mid \text{correct}] - \mathbb{E}[\ell(C_t) \mid \text{incorrect}],$$
$$\ell(c) = \log \frac{c(K-1)}{1-c},$$

the sample version of the condition $\mu_k > 0$, where $\ell(c) = \log \frac{c(K-1)}{1-c}$ is the log-odds transform from Algorithm 1. Its sign predicts whether CGES converges and its magnitude predicts how fast. Figure 6 plots $\hat{\mu}$ against CGES efficiency. Within each confidence family the two are strongly correlated, with r=0.97 (p=0.005) for CGES-RM across the five tasks and r=0.53 (p=0.006) for the logprob-based methods across 25 (method, task) pairs. Pooling all 30 pairs gives a weaker correlation (r=0.30) because PRM rewards and token-probability scores occupy different ranges. PRM scores spread across $[0, 1]$ while token probabilities cluster near 0.9, and the log-odds transform ℓ stretches the former far more than the latter. This is visible in the two panels of Figure 6, where the CGES-RM drift axis runs to about 12 while the logprob axis stays below 0.4. A standardized drift $\hat{\mu}/\sigma_{\ell(C)}$ partly removes this scale gap, and a fully scale-free cross-family predictor is left to future work. In practice $\hat{\mu}$ is best read as a task-level predictor. It answers whether CGES will help on a given dataset, which a practitioner can check on a small labelled slice before deploying.

Table 4: Results at budget $B = 8$. CGES rows use the efficient γ (smallest γ matching SC$\pm$0.2 pp). Mean$\pm$std across 10 seeds. Δ rows show difference vs. SC (green=improvement, red=regression). Last column: avg over tasks. Bold = best, underline = second-best per column. ESC omitted: at $B = 8$ its smallest window (w=4) leaves only two windows, too few for its early-stopping rule to be meaningful.

Method	AIME24		GPQA		MATH500		GSM8K		MMLU_Pro		**Avg**	
	Acc (%)	#Calls	Acc (%)	#Calls	Acc (%)	#Calls	Acc (%)	#Calls	Acc (%)	#Calls	Acc (%)	#Calls
SC	$76.0_{\pm 2.9}$	$8.00_{\pm 0.00}$	$\mathbf{47.9}_{\pm 2.1}$	$8.00_{\pm 0.00}$	$79.8_{\pm 0.6}$	$8.00_{\pm 0.00}$	$93.9_{\pm 0.2}$	$8.00_{\pm 0.00}$	$61.0_{\pm 0.4}$	$8.00_{\pm 0.00}$	**71.7**	8.00
ASC-β	$76.0_{\pm 2.9}$	$6.46_{\pm 0.10}$	$46.0_{\pm 2.1}$	$\underline{7.22}_{\pm 0.07}$	$79.8_{\pm 0.6}$	$5.41_{\pm 0.03}$	$93.9_{\pm 0.2}$	$4.49_{\pm 0.01}$	$61.0_{\pm 0.4}$	$5.94_{\pm 0.02}$	71.3	5.90
Δ	0.0	−1.5	−1.9	−0.8	0.0	−2.6	0.0	−3.5	0.0	−2.1	−0.4	−2.1
ASC-Dir	$76.0_{\pm 2.9}$	$6.46_{\pm 0.10}$	$46.0_{\pm 2.1}$	$7.22_{\pm 0.07}$	$79.8_{\pm 0.6}$	$5.41_{\pm 0.03}$	$93.9_{\pm 0.2}$	$4.49_{\pm 0.01}$	$61.0_{\pm 0.4}$	$5.94_{\pm 0.02}$	71.3	5.90
Δ	0.0	−1.5	−1.9	−0.8	0.0	−2.6	0.0	−3.5	0.0	−2.1	−0.4	−2.1
DSC	$69.3_{\pm 4.7}$	$\mathbf{3.71}_{\pm 0.27}$	$42.8_{\pm 2.6}$	$\mathbf{3.76}_{\pm 0.11}$	$78.2_{\pm 0.7}$	$3.16_{\pm 0.03}$	$93.4_{\pm 0.3}$	$2.34_{\pm 0.02}$	$59.5_{\pm 0.5}$	$\mathbf{3.26}_{\pm 0.04}$	68.7	**3.25**†
Δ	−6.7	−4.3	−5.1	−4.2	−1.6	−4.8	−0.5	−5.7	−1.5	−4.7	−3.1	−4.8
CGES-LNS (arith)	$76.3_{\pm 2.3}$	$4.29_{\pm 0.24}$	$46.4_{\pm 1.8}$	$7.67_{\pm 0.02}$	$79.8_{\pm 0.4}$	$3.17_{\pm 0.05}$	$\underline{94.0}_{\pm 0.3}$	$2.35_{\pm 0.02}$	$\underline{61.1}_{\pm 0.4}$	$4.37_{\pm 0.02}$	71.5	<u>4.37</u>
Δ	+0.3	−3.7	−1.5	−0.3	0.0	−4.8	+0.1	−5.7	+0.1	−3.6	−0.2	−3.6
CGES-LNS (geom)	$\underline{76.7}_{\pm 2.6}$	$4.43_{\pm 0.26}$	$46.3_{\pm 1.6}$	$7.98_{\pm 0.01}$	$\underline{80.1}_{\pm 0.3}$	$3.63_{\pm 0.05}$	$93.9_{\pm 0.3}$	$\underline{2.22}_{\pm 0.02}$	$61.0_{\pm 0.5}$	$4.46_{\pm 0.02}$	71.6	4.54
Δ	+0.7	−3.6	−1.6	0.0	+0.3	−4.4	0.0	−5.8	0.0	−3.5	−0.1	−3.5
CGES-DeepConf (B10)	$\mathbf{76.7}_{\pm 1.5}$	$4.61_{\pm 0.20}$	$\underline{46.6}_{\pm 1.7}$	$7.82_{\pm 0.01}$	$79.8_{\pm 0.3}$	$3.19_{\pm 0.04}$	$94.0_{\pm 0.3}$	$2.35_{\pm 0.02}$	$\mathbf{61.1}_{\pm 0.4}$	$4.39_{\pm 0.02}$	<u>71.6</u>	4.47
Δ	+0.7	−3.4	−1.3	−0.2	0.0	−4.8	+0.1	−5.7	+0.1	−3.6	−0.1	−3.5
CGES-DeepConf (tail)	$76.3_{\pm 3.1}$	$4.88_{\pm 0.20}$	$46.2_{\pm 1.9}$	$7.56_{\pm 0.04}$	$79.6_{\pm 0.2}$	$\underline{3.16}_{\pm 0.04}$	$94.0_{\pm 0.3}$	$2.35_{\pm 0.02}$	$61.0_{\pm 0.4}$	$\underline{4.37}_{\pm 0.02}$	71.4	4.46
Δ	+0.3	−3.1	−1.7	−0.4	−0.2	−4.8	+0.1	−5.7	0.0	−3.6	−0.3	−3.5
CGES-MARS	$76.3_{\pm 2.8}$	$\underline{4.12}_{\pm 0.21}$	$46.3_{\pm 1.4}$	$8.00_{\pm 0.01}$	$80.1_{\pm 0.4}$	$3.81_{\pm 0.06}$	$93.9_{\pm 0.3}$	$2.24_{\pm 0.02}$	$60.9_{\pm 0.4}$	$4.38_{\pm 0.02}$	71.5	4.51
Δ	+0.3	−3.9	−1.6	0.0	+0.3	−4.2	0.0	−5.8	−0.1	−3.6	−0.2	−3.5
CGES-RM	$52.7_{\pm 4.7}$	$7.46_{\pm 0.06}$	$43.7_{\pm 1.2}$	$7.88_{\pm 0.03}$	$\mathbf{80.8}_{\pm 0.8}$	$\mathbf{2.13}_{\pm 0.04}$	$\mathbf{94.8}_{\pm 0.5}$	$\mathbf{1.19}_{\pm 0.02}$	$58.5_{\pm 0.4}$	$7.16_{\pm 0.02}$	66.1	5.16
Δ	−23.3	−0.5	−4.2	−0.1	+1.0	−5.9	+0.9	−6.8	−2.5	−0.8	−5.6	−2.8

†DSC requires an additional LLM-based difficulty ranking step using the same inference model ($\approx$0.6 amortized calls/question; each call ranks B=5 questions jointly), excluded from #Calls but included in wall-clock time.

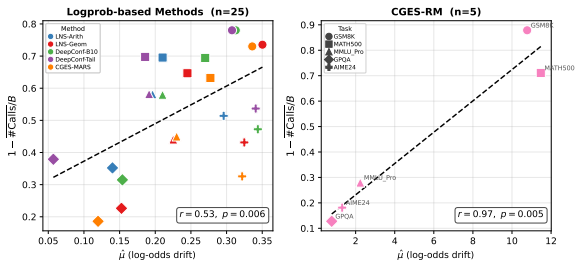


Figure 6: Empirical drift $\hat{\mu}$ vs. CGES efficiency $(1 - \overline{\#\text{Calls}}/B)$ at γ=0.999, B=16. Each point is one (method, task) pair. **Left**, the five logprob-based methods over 25 pairs (r=0.53, p=0.006). **Right**, CGES-RM over the five tasks (r=0.97, p=0.005). Within each family a larger drift yields a larger call reduction, as Theorem 3.5 predicts. The two panels are kept separate because PRM and token-probability scores live on different scales, so the x-axes are not comparable.

Table 5: Confidence signal quality per method and task. ECE (lower is better calibrated). AUC-ROC (higher discriminates correct from incorrect responses better). Efficiency $= 1 - \overline{\#\text{Calls}}/B$ at B=16, γ=0.999. GPQA shows the gap. CGES-RM has low ECE there, comparable to MATH500, yet near-random AUC, which explains its poor efficiency.

Method	AIME24			GPQA			MATH500			GSM8K			MMLU-Pro		
	ECE	AUC	Eff.	ECE	AUC	Eff.	ECE	AUC	Eff.	ECE	AUC	Eff.	ECE	AUC	Eff.
CGES-LNS (arith)	0.268	0.752	0.514	0.392	0.648	0.352	0.199	0.641	0.695	0.029	0.725	0.780	0.292	0.593	0.581
CGES-LNS (geom)	0.169	0.766	0.431	0.267	0.653	0.226	0.159	0.647	0.647	0.030	0.719	0.735	0.188	0.597	0.441
CGES-DeepConf (B10)	0.219	0.824	0.472	0.360	0.677	0.315	0.199	0.673	0.694	0.029	0.725	0.780	0.291	0.599	0.580
CGES-DeepConf (tail)	0.284	0.760	0.537	0.423	0.548	0.379	0.199	0.641	0.697	0.029	0.725	0.780	0.292	0.593	0.582
CGES-MARS	0.117	0.826	0.326	0.205	0.650	0.186	0.139	0.673	0.632	0.035	0.720	0.730	0.188	0.617	0.450
CGES-RM	0.216	0.682	0.181	0.050	0.685	0.128	0.050	0.954	0.711	0.021	0.940	0.879	0.141	0.758	0.280

K Robustness to Confidence Noise

To stress-test the directional-drift assumption we add Gaussian noise of standard deviation σ to each confidence score (after clipping to $(0, 1)$). Figure 7 plots accuracy as a function of σ on all five benchmarks. CGES is essentially flat for $\sigma \leq 0.05$ on every task and degrades gracefully for larger σ. On GSM8K and MATH500 the drop is roughly 2 to 5pp at σ=0.5. On AIME24 the drop is larger because the calibration was already weaker. The shape of these curves matches the theory. Noise shrinks the drift μ_k but does not flip its sign until σ is large enough to dominate the signal.

L Selecting the Stopping Threshold γ in Practice

A practical question is how to pick γ without a labelled validation set. Two options work in our experiments. First, γ has a direct calibrated interpretation in the ideal regime (Proposition 3.3). A user who wants posterior risk at most $1 - \gamma$ can pick γ directly. Second, in the noisy regime, the empirical drift $\hat{\mu}$ on a small labelled slice (50 to 100 questions) predicts efficiency within a confidence family (Appendix J). The user can pick γ by sweeping on this slice. In our tables, the efficient γ lies between 0.90 and 0.999 for every (method, task) pair, and the curves in Figure 3 are flat near the top of this range, so the choice is not sharp.

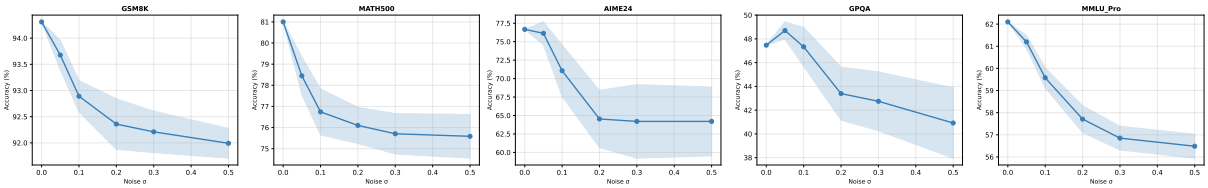


Figure 7: Accuracy of CGES under additive Gaussian noise injected into the confidence scores (σ on the x-axis). Bands show ± 1 standard deviation over 10 seeds at B=16. Small σ leaves accuracy nearly unchanged. Larger σ degrades accuracy gracefully, consistent with the directional-drift consistency result.

M The Use of Large Language Models (LLMs)

We used large language models (LLMs) solely to aid with polishing the writing and improving clarity of exposition. No part of the research ideation, methodology, analysis, or experimental results was generated by LLMs. The authors take full responsibility for the content of this paper.