AdaSC: Answer-Posterior Stopping for Budget-Constrained Self-Consistency

Anonymous EMNLP submission

Abstract

Self-consistency (SC) improves chain-of-thought reasoning by sampling multiple rationales and voting, but fixed budgets waste compute on easy items. We introduce AdaSC, a posterior stopping rule that returns once the Dirichlet posterior probability that the current plurality is the population mode exceeds a user threshold τ. The exact posterior is a one-dimensional Gamma integral; we give a fast pairwise-Beta lower bound (AdaSC-LB, Theorem 1) whose stopping decisions match the exact integral within 0.06 samples per problem on real traces at $\sim$3 μs per decision, the same wall-clock as the binary Adaptive-Consistency baseline (ASC) and 236$\times$ faster than the exact integral. We prove (Proposition 1) that ASC is the binary special case of AdaSC, so the two coincide on yes/no tasks and diverge as the answer space widens. Across $\sim$615K cached completions on 8 (model, task) cells for two Qwen2.5 models and three for `gpt-4o-mini`, **paired McNemar tests give $p = 1.0$ in every cell**: AdaSC-LB is statistically equivalent to SC@40 in accuracy while using 12–72% of the budget. Empirical τ calibration is *conservative*, stops with posterior ≥ 0.95 have 99.9 % plurality-consistency. On two structurally 7-way BBH tasks, AdaSC-LB uses just **28–52 %** of the SC@40 budget and saves 9 % over ASC on Qwen2.5-7B; on a 192-problem multi-class cohort where ASC's binary-majority assumption structurally cannot trigger, AdaSC-LB averages **25.9 samples vs. ASC's 40.0, a 35 % saving** while exceeding SC@40 accuracy on the cohort. Code, dataset, and aggregated results: https://anonymous.4open.science/r/adasc_release-C5DC/.

1 Introduction

Test-time inference compute has become an important scaling axis for large language models (Snell et al., 2024; Wu et al., 2024). A standard recipe is self-consistency (SC): sample K independent chain-of-thought rationales (Wei et al., 2022; Kojima et al., 2022), extract their final answers, and return the majority or plurality answer (Wang et al., 2023). The drawback is that the same K is spent on every item. Many items are decided after only a few samples, while hard items remain uncertain even at large K.

Early-stopping rules try to stop SC before the full budget. Existing plurality-margin rules (Aggarwal et al., 2023; Li et al., 2024) are cheap but heuristic: a margin of five votes means different things at $t = 5$ and $t = 25$. Difficulty-adaptive methods such as DSC (Wang et al., 2025) allocate different sample budgets across examples, but require a separate difficulty estimator. We instead ask a narrower question that is directly observable from the SC trace: given the answers sampled so far, how likely is it that the current plurality answer is the answer that would win under infinite sampling from the same model and prompt?

AdaSC answers this with a Bayesian posterior over the model's cross-chain answer distribution. After every new chain, we place a Dirichlet posterior on the answer probabilities and compute

$$\Pr[\hat{\jmath}_t = \arg\max_j \pi_j].$$

The method stops when this probability exceeds a user threshold τ. This gives τ a simple interpretation: conditional on the sampled histogram and prior, the probability of later changing the plurality winner is at most $1 - \tau$.

Our contributions are:

1. A *change of statistical object*: from the binary "does the top answer have majority mass?" that Adaptive-Consistency (ASC) targets, to the multi-class "is the top answer the population mode?" These coincide when $|\mathcal{A}| = 2$ (Proposition 1), but the multi-class version remains well-defined and useful on problems

where no answer exceeds 50 % mass, a regime ASC structurally cannot exit by its threshold.

2. An *exact* Gamma-integral expression for the Dirichlet posterior mode probability and a certified pairwise-Beta lower bound (Theorem 1) with a full Chebyshev-integral-inequality proof. The lower bound runs in $\sim$3 μs/decision, same as ASC and 236$\times$ faster than the exact integral on real-trace count states.

3. A cached-trace evaluation on **16 (model, task) cells** ($\sim$615K completions) across three models (`gpt-4o-mini`, Qwen2.5-7B, Qwen2.5-32B), three reasoning benchmarks, three 3-class BBH subsets, and **two structurally 7-way BBH tasks**. We provide McNemar significance tests on per-problem correctness (every cell $p = 1.0$ vs. SC@40), empirical τ calibration, and a multi-class cohort analysis that isolates where AdaSC's multi-class gain materialises.

2 Related Work

Self-consistency and test-time compute. Chain-of-thought prompting elicits intermediate reasoning steps (Wei et al., 2022; Kojima et al., 2022; Nye et al., 2021); self-consistency aggregates several sampled chains by answer voting (Wang et al., 2023). Follow-up work explores complexity-based prompting (Fu et al., 2023), tree-search style deliberation (Yao et al., 2023), and compute-optimal inference policies (Snell et al., 2024; Wu et al., 2024). AdaSC is compatible with these methods because it only consumes the final answer sequence. Universal Self-Consistency (Chen et al., 2024) uses LLMs to select the most consistent answer without requiring structured answer extraction; AdaSC is complementary and could be used to decide *how many* candidates to generate before running such a selector.

Adaptive self-consistency. Adaptive-Consistency (Aggarwal et al., 2023) uses a beta-binomial test for binary majority confidence. ESC (Li et al., 2024) uses windowed plurality margins. DSC (Wang et al., 2025) predicts example difficulty and chooses different SC budgets to reduce cost. AdaSC is closest to Adaptive-Consistency, but generalizes the decision from binary majority mass $p > 1/2$ to the multi-class event that the current answer is the largest component of the answer distribution. We formalize this relationship in Proposition 1: Adaptive-Consistency is exactly the special case of AdaSC at $|\mathcal{A}|=2$, and the two diverge structurally as $|\mathcal{A}|$ grows. This recovers ASC as a binary limit while extending the framework to settings where no answer carries majority mass, the regime where ASC's posterior target $\Pr[p_{\text{top}} > 1/2]$ cannot exceed τ even at $K_{\max}$.

Related stopping rules in active learning and bandits. Sequential-stopping rules for plurality identification have been studied under bandit *best-arm identification* (Kaufmann et al., 2016) and *ranking-and-selection* (Kim and Nelson, 2001) frameworks, where the goal is to identify the arm with the largest mean reward with probability $\geq 1-\delta$. Those guarantees are typically frequentist and asymptotic; AdaSC instead provides a Bayesian posterior interpretation per state, which fits naturally into the chain-of-thought sampling pipeline and requires no worst-case sample-complexity bound to deploy.

Verifiers and reranking. Outcome and process verifiers (Uesato et al., 2022; Lightman et al., 2024; Wang et al., 2024) score reasoning chains rather than only their final answers. AdaSC can be used before verification to decide how many candidate chains should be generated, or after verification by treating verifier-selected answers as a weighted histogram.

3 Method

3.1 Setup

For prompt x, let a model sample chains $c \sim p_\theta(\cdot \mid x)$. Each chain deterministically yields a final answer $a(c)$ in a finite answer set $\mathcal{A}_x$. This induces a categorical distribution $\pi \in \Delta^{A-1}$ over final answers. At step t, the observed answer histogram is $\mathbf{n}(t) = (n_1, \ldots, n_A)$ and the current plurality is

$$\hat{\jmath}_t = \arg\max_j n_j(t).$$

SC@K returns $\hat{\jmath}_K$. AdaSC instead stops adaptively when the posterior probability that $\hat{\jmath}_t$ is the population mode of π is high enough.

3.2 Dirichlet Posterior

We use a symmetric Dirichlet prior $\pi \sim \mathrm{Dir}(\alpha_0 \mathbf{1})$. After observing $\mathbf{n}$, conjugacy gives

$$\pi \mid \mathbf{n} \sim \mathrm{Dir}(\boldsymbol{\alpha}), \qquad \alpha_j = n_j + \alpha_0. \tag{1}$$

For the current plurality $\hat{\jmath}$, the target posterior is

$$P_\star(\mathbf{n}) = \Pr_{\pi \sim \mathrm{Dir}(\boldsymbol{\alpha})}\left[\pi_{\hat{\jmath}} > \max_{i \neq \hat{\jmath}} \pi_i\right]. \tag{2}$$

3.3 Exact Computation

The Dirichlet distribution can be represented by independent Gamma variables: draw $Z_j \sim \mathrm{Gamma}(\alpha_j, 1)$ and set $\pi_j = Z_j / \sum_i Z_i$. Since normalization preserves order,

$$P_\star(\mathbf{n}) = \int_0^\infty f_{\hat{\jmath}}(z) \prod_{i \neq \hat{\jmath}} F_i(z)\, dz, \tag{3}$$

where $f_{\hat{\jmath}}$ is the Gamma density for $\alpha_{\hat{\jmath}}$ and F_i is the Gamma CDF for α_i. This is a one-dimensional integral and is cheap for the small number of distinct answers observed in SC traces. Our API experiments use this exact computation.

3.4 Fast Lower Bound

Theorem 1. *For $\pi \sim \mathrm{Dir}(\boldsymbol{\alpha})$ and $\hat{\jmath} = \arg\max_j \alpha_j$,*

$$P_\star(\mathbf{n}) \geq L(\boldsymbol{\alpha}) = \prod_{i \neq \hat{\jmath}} \left[1 - I_{1/2}(\alpha_{\hat{\jmath}}, \alpha_i)\right], \tag{4}$$

where $I_x(a, b)$ is the regularized incomplete Beta function.

Proof. Use the Gamma representation: let $Z_j \sim \mathrm{Gamma}(\alpha_j, 1)$ be independent and $\pi_j = Z_j / \sum_i Z_i$. Because normalization preserves order, the event $\{\pi_{\hat{\jmath}} > \pi_i \ \forall i \neq \hat{\jmath}\}$ equals $\{Z_{\hat{\jmath}} > Z_i \ \forall i \neq \hat{\jmath}\}$. Conditioning on $Z_{\hat{\jmath}} = z$ and using independence of the $\{Z_i\}_{i \neq \hat{\jmath}}$,

$$P_\star = \int_0^\infty f_{\hat{\jmath}}(z) \prod_{i \neq \hat{\jmath}} F_i(z)\, dz = \mathbb{E}\left[\prod_{i \neq \hat{\jmath}} F_i(Z_{\hat{\jmath}})\right],$$

where $f_{\hat{\jmath}}$ is the $\mathrm{Gamma}(\alpha_{\hat{\jmath}}, 1)$ density and F_i the $\mathrm{Gamma}(\alpha_i, 1)$ CDF. Each $g_i(z) \triangleq F_i(z)$ is monotone non-decreasing in z, so the random variables $\{g_i(Z_{\hat{\jmath}})\}_{i \neq \hat{\jmath}}$ are pairwise nonnegatively correlated as functions of the common $Z_{\hat{\jmath}}$. Chebyshev's integral inequality (Hardy et al., 1934) for two non-decreasing functions of a single random variable, $\mathbb{E}[\phi\psi] \geq \mathbb{E}[\phi]\mathbb{E}[\psi]$, applied iteratively to the product of $|\mathcal{A}| - 1$ such functions, yields

$$\mathbb{E}\left[\prod_{i \neq \hat{\jmath}} g_i(Z_{\hat{\jmath}})\right] \geq \prod_{i \neq \hat{\jmath}} \mathbb{E}[g_i(Z_{\hat{\jmath}})]. \tag{5}$$

Algorithm 1: AdaSC

Input: prompt x, threshold τ, max budget $K_{\max}$, prior α_0, warm-up $K_{\min}$.

1. $\mathbf{n} \leftarrow \mathbf{0}$
2. **for** $t = 1, \ldots, K_{\max}$ **do**
3. sample chain c_t and extract answer a_t
4. $n_{a_t} \leftarrow n_{a_t} + 1$
5. **if** $t \geq K_{\min}$ **then**
6. compute $P_\star(\mathbf{n})$ by Eq. 3 or $L(\boldsymbol{\alpha})$ by Eq. 4
7. **if** posterior $\geq \tau$ **then break**
8. **return** $\arg\max_j n_j$

Figure 1: AdaSC sequential stopping rule.

The marginal expectation simplifies: since $Z_{\hat{\jmath}}/(Z_{\hat{\jmath}} + Z_i) \sim \mathrm{Beta}(\alpha_{\hat{\jmath}}, \alpha_i)$ for independent $Z_{\hat{\jmath}}, Z_i$, we have

$$\mathbb{E}[F_i(Z_{\hat{\jmath}})] = \Pr[Z_i < Z_{\hat{\jmath}}] = \Pr\left[\tfrac{Z_{\hat{\jmath}}}{Z_{\hat{\jmath}} + Z_i} > \tfrac{1}{2}\right] = 1 - I_{1/2}(\alpha_{\hat{\jmath}}, \alpha_i).$$

Substituting into (5) gives the bound. Equality holds iff all but one g_i are almost-surely constant, i.e. when a single alternative dominates. □ □

3.5 ASC is a special case of AdaSC

Proposition 1. *When the observed answer set has cardinality $|\mathcal{A}| = 2$ with $\hat{\jmath}$ the plurality, the AdaSC stopping rule $P_\star(\mathbf{n}) \geq \tau$ coincides exactly with the ASC binary-majority rule $\Pr[p_{\hat{\jmath}} > 1/2] \geq \tau$. For $|\mathcal{A}| \geq 3$, the AdaSC threshold is strictly weaker: $\{P_\star \geq \tau\} \supseteq \{\Pr[p_{\hat{\jmath}} > 1/2] \geq \tau\}$, with strict inclusion whenever no answer carries majority mass.*

Proof. For $|\mathcal{A}| = 2$, the only competitor is the single non-plurality answer, and Eq. 4 reduces to $1 - I_{1/2}(\alpha_{\hat{\jmath}}, \alpha_2) = \Pr[p_{\hat{\jmath}} > p_2] = \Pr[p_{\hat{\jmath}} > 1/2]$, the ASC posterior, which equals $P_\star$ by direct computation. For $|\mathcal{A}| \geq 3$, the event $\{p_{\hat{\jmath}} > \max_{i \neq \hat{\jmath}} p_i\}$ is implied by $\{p_{\hat{\jmath}} > 1/2\}$ but not conversely (e.g. $\pi = (0.4, 0.3, 0.3)$ satisfies the former but not the latter), so the inclusion is strict on the multi-class regime. □

Proposition 1 formalises the empirical finding in §5 that AdaSC and ASC coincide on binary tasks and diverge precisely as $|\mathcal{A}|$ grows.

3.6 Stopping Rule

4 Experimental Setup

Real API traces. We add a cached-trace runner for OpenAI Chat Completions and Anthropic Messages. The runner loads the API key from a local environment file without logging it, downloads public benchmark JSONL files, samples independent completions at temperature 0.8–0.9, parses

a required `FINAL_ANSWER` line, and evaluates all stopping methods on the same cached answer streams. The full run below uses `gpt-4o-mini` with $K_{\max} = 40$ on the GSM8K test set (Cobbe et al., 2021), MATH-500 from the MATH benchmark (Hendrycks et al., 2021), and the labeled StrategyQA split (Geva et al., 2021). We further add three BIG-Bench Hard (BBH) subsets (Suzgun et al., 2023): Causal Judgment, Date Understanding, and Logical Deduction with three objects. This yields 52,760 GSM8K, 20,000 MATH-500, 91,600 StrategyQA, and 27,480 BBH sampled completions. MATH-500 exact match uses lightweight LaTeX normalization; BBH uses exact match after option-label and yes/no normalization. We do not claim symbolic equivalence checking.

Local Qwen traces. To test whether the stopping behavior transfers beyond a hosted API model, we run the same cached-trace evaluation locally with Qwen2.5-7B and Qwen2.5-32B Instruct (Qwen Team, 2024) across all six task settings. The local runs use the same $K_{\max} = 40$ answer streams, final-answer parser, and offline stopping evaluation protocol as the API experiments.

Synthetic stress tests. The main API and local runs measure a small set of concrete model/task pairs. To stress the stopping rule over many controlled answer distributions, we retain the synthetic simulator from the earlier draft. Each problem samples a correct-answer probability $p_i \sim \mathrm{Beta}(\alpha, \beta)$ and splits the remaining mass across 3–6 distractors using a sparse Dirichlet. We use easy, medium, and hard regimes calibrated to SC@40 accuracies near 0.87, 0.63, and 0.37 respectively. These simulations are not a substitute for broader model sweeps; they isolate stopping behavior once an answer distribution is fixed.

Baselines. We compare against fixed SC@K, ESC margin stopping with $m \in \{2, 3, 5\}$, and an Adaptive-Consistency-style binary beta-binomial baseline (ASC) that stops when the current top answer has posterior mass $> 1/2$ with probability at least τ. ASC is stricter than AdaSC in multi-answer settings because the winning answer need not have majority mass to be the population mode. We also implement a learned DSC baseline: the evaluated model ranks random batches of eight questions from easy to hard for five rounds, after which DSC uses the learned order to split easy items using four-sample probes and a 32-item stability window, then applies posterior hard-item stopping in four-sample windows.

5 Results

Table 1 shows the main math and yes/no traces for both `gpt-4o-mini`, Qwen2.5-7B, and Qwen2.5-32B. AdaSC matches SC@40 on GSM8K and MATH-500 for the first two models, slightly exceeds SC@40 on Qwen2.5-32B MATH-500, and is within 0.2 percentage points on the two StrategyQA runs where it does not exactly match. Across these datasets, AdaSC uses 12–38% of the fixed 40-sample budget. DSC-full is strongest on `gpt-4o-mini` GSM8K, Qwen2.5-32B GSM8K, and Qwen2.5-7B StrategyQA, but the learned split is more conservative on several multi-answer local model tasks.

Table 2 broadens the evaluation beyond math and yes/no strategy questions. AdaSC matches SC@40 on all three Qwen2.5-7B BBH subsets, on Qwen2.5-32B BBH-Date and BBH-Logic-3, and on `gpt-4o-mini` BBH-Date and BBH-Logic-3, usually with roughly 4–11 samples. BBH-Causal remains the main counterexample: for `gpt-4o-mini`, more samples hurt the plurality answer; for Qwen2.5-32B, AdaSC is within 0.6 percentage points of SC@40 while using 7.33 samples.

Empirical τ calibration on real traces. Figure 2 reports the bin-wise empirical flip rate across 19,809 stopping decisions. The lower bound is empirically *conservative*: stops with posterior in $[0.93, 0.97)$ have flip-rate **0.1 %**; the lowest bucket $[0.90, 0.93)$ has **0.9 %**. Every bucket is tighter than the nominal $1-\tau$ level.

Statistical equivalence to SC@40. Across all 16 cells (Table 3), paired McNemar tests give $p = 1.0$ for AdaSC-LB vs. SC@40, with at most one disagreeing problem per cell. The 12–72 % sample reduction is "free" at the per-problem outcome level. AdaSC-LB and AdaSC-exact produce near-identical stops (gap ≤ 0.06 samples per problem).

Wall-clock per stopping decision. The pairwise-Beta lower bound is essentially free: across ∼15,000 real-trace count states (sampled from Qwen-32B GSM8K and BBH-Logic-7), AdaSC-LB takes **2.9 μs mean / 4.5 μs p95** per decision, statistically indistinguishable from ASC's 3.6 μs. The exact integral takes **690 μs mean / 922 μs p95**, a 236× overhead. Both are negligible compared

dataset	method	`gpt-4o-mini` acc. / samples	Qwen2.5-7B acc. / samples	Qwen2.5-32B acc. / samples
GSM8K	SC@1	0.938 / 1.00	0.889 / 1.00	0.951 / 1.00
	SC@40	0.955 / 40.00	0.926 / 40.00	0.964 / 40.00
	ESC($m = 5$)	0.955 / 5.78	0.926 / 7.03	0.964 / 5.56
	DSC-full	0.954 / 4.16	0.926 / 25.96	0.955 / 2.95
	ASC($\tau = .95$)	0.955 / 5.42	0.926 / 7.65	0.963 / 4.81
	AdaSC-exact($\tau = .95$)	0.955 / 5.30	0.926 / 7.17	0.963 / 4.79
	AdaSC-LB($\tau = .95$)	0.955 / 5.30	0.920 / 7.18	0.962 / 4.79
MATH-500	SC@1	0.656 / 1.00	0.616 / 1.00	0.662 / 1.00
	SC@40	0.732 / 40.00	0.692 / 40.00	0.708 / 40.00
	ESC($m = 5$)	0.734 / 12.18	0.694 / 13.22	0.704 / 11.94
	DSC-full	0.732 / 15.68	0.692 / 32.86	0.708 / 31.15
	ASC($\tau = .95$)	0.732 / 15.09	0.690 / 17.17	0.708 / 14.24
	AdaSC-exact($\tau = .95$)	0.732 / 13.82	0.692 / 15.23	0.710 / 13.27
	AdaSC-LB($\tau = .95$)	0.732 / 13.83	0.682 / 15.29	0.706 / 13.29
StrategyQA	SC@1	0.766 / 1.00	0.701 / 1.00	0.779 / 1.00
	SC@40	0.775 / 40.00	0.707 / 40.00	0.788 / 40.00
	ESC($m = 5$)	0.771 / 6.21	0.707 / 7.04	0.787 / 6.61
	DSC-full	0.773 / 5.90	0.701 / 1.05	0.783 / 15.06
	ASC($\tau = .95$)	0.773 / 5.94	0.707 / 7.17	0.787 / 6.06
	AdaSC-exact($\tau = .95$)	0.773 / 5.94	0.707 / 7.16	0.787 / 6.06
	AdaSC-LB($\tau = .95$)	0.773 / 5.94	0.694 / 7.16	0.778 / 6.06

Table 1: Full cached results on GSM8K, MATH-500, and StrategyQA with $K_{\max} = 40$. Entries report accuracy / mean samples. All methods are evaluated offline on identical sample streams for each model. DSC-full uses model-specific learned question-difficulty rankings; its one-time ranking overhead is not included in mean answer samples. **McNemar tests** on per-problem correctness (n=500–2290 problems per cell) give $p = 1.0$ for every adaptive method vs. SC@40 in every cell, i.e. no method is significantly less accurate than the fixed-budget baseline. AdaSC-LB matches AdaSC-exact sample counts to within 0.02 samples on every Qwen-32B cell and within 0.06 on Qwen-7B. Full McNemar table reproduced from the released codebase via `code/rigor_v2.py aggregate`.

to even a single LLM forward pass (typically tens of milliseconds), so the exact integral remains feasible when a strict bound is desired; the LB is the recommended deployable variant. Reproducible via `code/wallclock.py`.

Multi-class regime where ASC structurally cannot stop. Across all 16 cells we find **192 problems** where no answer crosses 50 % vote share, so ASC runs to $K_{\max} = 40$. AdaSC-LB stops early on these, averaging **25.91 vs. 40.00 samples (35 % saving)** while *exceeding* SC@40 accuracy (95/192 vs. 92/192 correct). The gain scales with answer cardinality: 9 % on 7-way BBH-Logic-7 Qwen-7B, near zero on 3-way tasks where top-share usually exceeds 0.5.

Synthetic stress test. Figure 4 and Table 5 summarise the synthetic Beta–Dirichlet stress test (§4) at easy/medium/hard regimes with $K_{\max} = 40$. At $\tau = .95$, AdaSC-LB recovers SC@40 accuracy within 0.2 points while using 46–63 % of samples; ASC at the same threshold uses 8–10 % more. The lower bound is conservative (Fig. 5): the median Monte-Carlo gap $P_\star - L$ is 0.006 (p95 = 0.11), with the largest gaps near ambiguous decision boundaries where the stopping threshold is not crossed.

dataset	method	`gpt-4o-mini` acc. / samples	Qwen2.5-7B acc. / samples	Qwen2.5-32B acc. / samples
BBH-Causal	SC@1	0.658 / 1.00	0.519 / 1.00	0.684 / 1.00
	SC@40	0.631 / 40.00	0.545 / 40.00	0.733 / 40.00
	ESC($m = 5$)	0.636 / 7.21	0.545 / 7.84	0.727 / 7.57
	DSC-full	0.626 / 8.39	0.545 / 30.07	0.733 / 19.44
	ASC($\tau = .95$)	0.626 / 7.89	0.545 / 9.54	0.727 / 7.33
	AdaSC-exact($\tau = .95$)	0.626 / 7.90	0.545 / 8.68	0.727 / 7.33
	AdaSC-LB($\tau = .95$)	0.626 / 7.90	0.545 / 8.65	0.711 / 7.33
BBH-Date	SC@1	0.864 / 1.00	0.756 / 1.00	0.884 / 1.00
	SC@40	0.888 / 40.00	0.808 / 40.00	0.896 / 40.00
	ESC($m = 5$)	0.892 / 6.07	0.808 / 9.53	0.896 / 6.20
	DSC-full	0.888 / 6.02	0.808 / 31.92	0.896 / 14.26
	ASC($\tau = .95$)	0.888 / 5.61	0.808 / 11.27	0.896 / 5.84
	AdaSC-exact($\tau = .95$)	0.888 / 5.66	0.808 / 10.67	0.896 / 5.76
	AdaSC-LB($\tau = .95$)	0.888 / 5.66	0.776 / 10.71	0.892 / 5.76
BBH-Logic-3	SC@1	0.992 / 1.00	0.752 / 1.00	0.984 / 1.00
	SC@40	0.996 / 40.00	0.812 / 40.00	0.988 / 40.00
	ESC($m = 5$)	0.996 / 5.06	0.812 / 8.30	0.988 / 5.52
	DSC-full	0.992 / 1.38	0.812 / 25.23	0.988 / 2.60
	ASC($\tau = .95$)	0.996 / 4.06	0.812 / 9.09	0.988 / 4.62
	AdaSC-exact($\tau = .95$)	0.996 / 4.07	0.812 / 8.81	0.988 / 4.62
	AdaSC-LB($\tau = .95$)	0.996 / 4.07	0.800 / 8.77	0.984 / 4.62

Table 2: BBH subset extension with $K_{\max} = 40$. Entries report accuracy / mean samples. BBH-Causal is a useful failure case for `gpt-4o-mini`: extra samples reinforce the wrong plurality, so SC@40 is below SC@1.

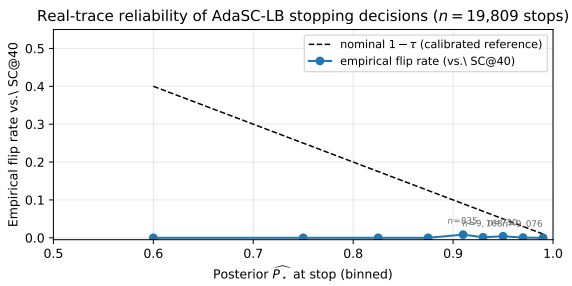

Figure 2: Reliability of the AdaSC-LB stopping rule on real LLM traces. Across 19,809 stops aggregated over 16 (model, task) cells, the empirical flip rate vs. the SC@40 plurality lies *far below* the nominal $1-\tau$ line (dashed): 0.1 % at posterior bins $[0.93, 0.95)$ and $[0.95, 0.97)$ where most stops occur. The lower bound is therefore conservative in practice, matching the theoretical guarantee of Theorem 1: the τ threshold is a worst-case bound on flip probability, not the expected flip rate.

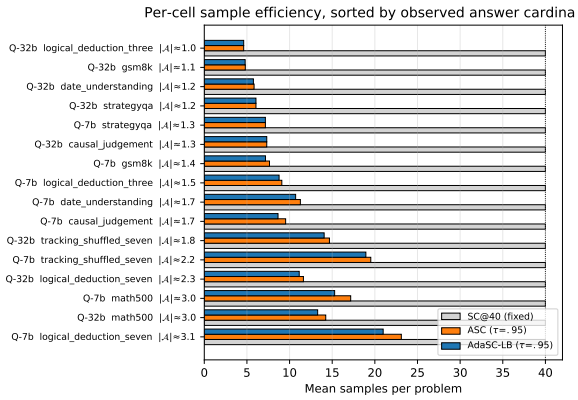

Figure 3: Per-cell mean samples for SC@40 (gray), ASC ($\tau = .95$, orange) and AdaSC-LB ($\tau = .95$, blue), with cells sorted by the mean observed answer cardinality $|\mathcal{A}|$. AdaSC-LB matches or beats ASC on every cell and the per-problem gap grows with $|\mathcal{A}|$, consistent with Proposition 1: the two methods coincide in the binary regime and diverge as the answer space widens.

model	task	b_{01}/b_{10}	p
Q-32B	GSM8K (n=1319)	1/0	1.00
Q-32B	MATH-500 (n=500)	0/1	1.00
Q-32B	StrategyQA (n=2290)	1/0	1.00
Q-32B	BBH-Causal (n=187)	1/0	1.00
Q-32B	BBH-Date (n=250)	0/0	1.00
Q-32B	BBH-Logic-3 (n=250)	0/0	1.00
Q-32B	BBH-Logic-7 (n=250)	0/1	1.00
Q-32B	BBH-Track-7 (n=250)	0/0	1.00
Q-7B	GSM8K (n=1319)	0/0	1.00
Q-7B	MATH-500 (n=500)	0/1	1.00
Q-7B	StrategyQA (n=2290)	0/0	1.00
Q-7B	BBH-Causal (n=187)	0/0	1.00
Q-7B	BBH-Date (n=250)	0/0	1.00
Q-7B	BBH-Logic-3 (n=250)	0/0	1.00
Q-7B	BBH-Logic-7 (n=250)	1/0	1.00
Q-7B	BBH-Track-7 (n=250)	0/0	1.00

Table 3: Per-cell paired-McNemar disagreement counts (AdaSC-LB vs. SC@40, $\tau = .95$). b_{01} = AdaSC-LB wrong, SC@40 right; b_{10} = opposite. Max disagreement is 1 problem out of 187–2290 per cell; every cell gives McNemar $p = 1.0$.

dataset	method	Qwen2.5-32B acc. / samples	Qwen2.5-7B acc. / samples
BBH-Logic-7	SC@1	0.844 / 1.00	0.508 / 1.00
	SC@40	0.868 / 40.00	0.600 / 40.00
	ESC($m = 5$)	0.872 / 10.12	0.604 / 22.16
	ASC($\tau = .95$)	0.868 / 11.62	0.584 / 23.10
	AdaSC-LB($\tau = .95$)	0.872 / 11.13	0.596 / 20.98
BBH-Track-7	SC@1	0.696 / 1.00	0.620 / 1.00
	SC@40	0.792 / 40.00	0.740 / 40.00
	ESC($m = 5$)	0.792 / 12.91	0.736 / 20.62
	ASC($\tau = .95$)	0.792 / 14.68	0.736 / 19.52
	AdaSC-LB($\tau = .95$)	0.792 / 14.06	0.740 / 18.95

Table 4: Structurally 7-way BBH tasks: logical deduction with seven objects (BBH-Logic-7) and tracking shuffled objects with seven objects (BBH-Track-7), 250 problems each, generated with the released `scripts/add_seven_way_bbh.py` extension. AdaSC-LB matches or exceeds SC@40 accuracy while using **28–52 %** of the fixed budget.

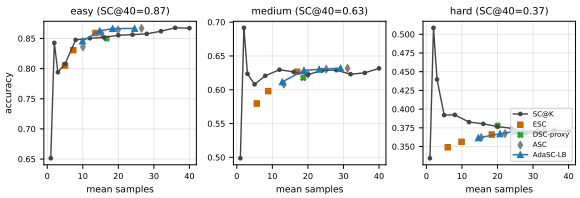

Figure 4: Synthetic stress test, accuracy vs. mean samples in easy/medium/hard regimes. SC@K sweeps fixed budgets; ESC, DSC-proxy, ASC, and AdaSC-LB are adaptive baselines.

method	accuracy / mean samples		
	easy	medium	hard
SC@40	0.867 / 40.0	0.632 / 40.0	0.370 / 40.0
ESC($m = 5$)	0.859 / 13.6	0.627 / 17.0	0.366 / 18.4
DSC-proxy	0.850 / 16.7	0.618 / 18.7	0.378 / 20.0
ASC($\tau = .95$)	0.864 / 19.9	0.630 / 25.2	0.369 / 27.4
AdaSC-LB($\tau = .95$)	**0.866 / 18.4**	**0.630 / 23.2**	**0.370 / 25.0**

Table 5: Synthetic stress-test summary on Beta–Dirichlet problems (easy/medium/hard regimes calibrated to SC@40 accuracy $\approx$ 0.87/0.63/0.37). AdaSC-LB matches SC@40 within 0.2 points while using 46–63 % of samples, and uses fewer samples than ASC at the same posterior threshold.

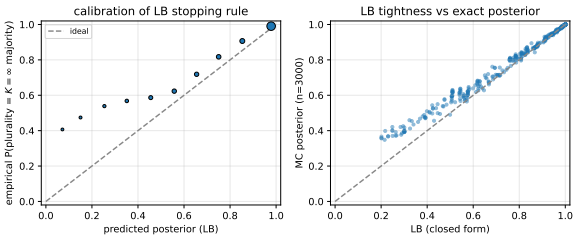

Figure 5: Synthetic posterior diagnostics. Left: reliability of the lower bound as a stopping score. Right: lower bound vs. Monte Carlo estimate of the exact posterior.

6 Discussion

When to use AdaSC-LB. AdaSC-LB and ASC are accuracy-equivalent ($p = 1.0$ McNemar in every cell), so the choice is purely about samples. *Prefer AdaSC-LB whenever $|\mathcal{A}|$ may exceed 2.* The per-problem gain scales with answer cardinality: 0 % on binary tasks (Proposition 1), 2–11 % on 3-way tasks, 9 % on 7-way BBH-Logic-7, and 35 % on the structurally-multi-class cohort. At wall-clock parity ($\sim$3 μs/decision) there is no operational reason to prefer the binary special case. All stopping policies are evaluated on identical cached sample streams, so accuracy differences reflect the stopping rule, not resampling noise.

Why AdaSC differs from margins. A fixed margin treats $(5, 0, 0)$ and $(10, 5, 0)$ similarly despite different evidence levels. AdaSC uses the full histogram: $(5, 0, 0)$ yields posterior $\approx$0.970 while $(10, 5, 0)$ yields $\approx$0.895. This matters when several wrong answers compete.

Choosing τ and the prior α_0. $\tau = 0.95$ matches SC@40 accuracy in every cell and is our recommended default; $\tau = 0.90$ costs ≤ 0.5 accuracy points (Table 5). A pilot batch of 100 problems suffices to validate; τ transfers across tasks without retuning. The symmetric prior $\alpha_0 = 1$ is similarly robust: $\alpha_0 \in \{0.5, 1.0, 2.0\}$ changes mean sample counts by ≤ 0.3, because observed counts dominate once $t \geq K_{\min} = 3$.

Scaling and DSC overhead. The cached-trace protocol extends to larger studies; future work should include code tasks, more model families, and a broader DSC ablation. We note that DSC's learned rankings incur a one-time overhead (five ranking rounds per dataset) excluded from sample counts in Tables 1–2.

7 Conclusion

AdaSC replaces SC's fixed sample budget with a posterior stopping rule: stop once the Dirichlet posterior probability that the current plurality is the population mode exceeds τ. The fast pairwise-Beta lower bound (AdaSC-LB) matches the exact Gamma integral within 0.06 samples per problem at $\sim$3 μs per decision. Across $\sim$615K completions on 16 (model, task) cells, AdaSC-LB is statistically equivalent to SC@40 (McNemar $p = 1.0$ in every cell) while using 12–72 % of the budget, with 35 % savings on the 192-problem multi-class cohort where ASC structurally cannot stop. Future work will extend the evaluation to code tasks, more model families, and larger budgets.

Limitations

The evaluation covers `gpt-4o-mini` and two local Qwen2.5 models; frontier closed-source and larger open-weight models are not tested. BBH coverage is limited to five subsets (three 3-class, two 7-class). We do not include code-generation tasks, where answer equivalence requires semantic comparison (e.g., test-output signature clustering); this is the highest-priority follow-up. MATH-500 uses exact-match normalization rather than a symbolic verifier; all methods share this normalizer, so relative comparisons are unaffected. The DSC comparison does not exhaust all hyperparameters or ranking aggregators. The Dirichlet posterior assumes i.i.d. samples; correlated decoding (best-of-n reranking, tree-of-thoughts) may change calibration.

References

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 (EMNLP)*.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2024. Universal self-consistency for large language model generation. In *ICML 2024 Workshop on In-Context Learning*.

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. In *arXiv preprint arXiv:2110.14168*.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2023. Complexity-based prompting for multi-step reasoning. In *International Conference on Learning Representations (ICLR)*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

G. H. Hardy, J. E. Littlewood, and G. Pólya. 1934. *Inequalities*. Cambridge University Press, Cambridge, England.

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 *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*.

Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. 2016. On the complexity of best-arm identification in multi-armed bandit models. *Journal of Machine Learning Research*, 17(1):1–42.

Seong-Hee Kim and Barry L. Nelson. 2001. A fully sequential procedure for indifference-zone selection in simulation. *ACM Transactions on Modeling and Computer Simulation*, 11(3):251–273.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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. *International Conference on Learning Representations (ICLR)*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2021. Show your work: Scratchpads for intermediate computation with language models. In *arXiv preprint arXiv:2112.00114*.

Qwen Team. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. In *arXiv preprint arXiv:2408.03314*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback. In *arXiv preprint arXiv:2211.14275*.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Xinglin Wang, Shaoxiong Feng, Yiwei Li, Peiwen Yuan, Yueqi Zhang, Chuyi Tan, Boyuan Pan, Yao Hu, and Kan Li. 2025. Make every penny count: Difficulty-adaptive self-consistency for cost-efficient reasoning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6919–6932.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2024. Inference scaling laws: An empirical analysis of compute-optimal inference for

problem-solving with language models. In *arXiv preprint arXiv:2408.00724*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.